Kende Mária, Nagy István

Internetes adatbázis-alkalmazások fejlesztése

(Oracle Application Express)

– 2014 –

Tartalomjegyzék

Bevezetés....5
Application Express....5
Az Application Express előnyei....7
Mit lehet ezzel az eszközzel készíteni?....8
Application Express és az XE kapcsolata....8
1. fejezet. A munkakörnyezet létrehozása....9
Munkaterület, séma és felhasználó létrehozása....10
A felhasználó főablakának komponensei....15
A komponensek és a Funkció Sáv használata....17
Adminisztrátori (fejlesztői) teendők....17
Fejlesztő felhasználó létrehozás....18
Az APEX eszközeinek bemutatása....19
Az SQL műhely (SQL workshop)....19
Az SQL Commands komponens....19
Tábla listázása....20
Explain menüpont....21
Describe menüpont....22
Saved SQL menüpont....23
History menüpont....23
Az SQL Scripts komponens....24
Szkript fájl készítése és szerkesztése....24
Szkript fájl betöltése.....26
A Query Builder komponens....27
A Lekérdezés Varázsló (Query Builder) használata....28
Az Objektum varázsló komponens....33
Objektum-létrehozás....35
Segédprogramok (`Utilities`) komponens....38
Az Application Builder komponens....41
2. fejezet. Alkalmazás-létrehozás alapjai....42
Mintaalkalmazás megtekintése....43
Tábla megjelenítés Az első saját alkalmazás készítése....50
Alkalmazás készítésének kezdete....50
Lap (Page) felépítése, szerkesztése....57
Tartomány létrehozása....64
Riportok, Tábla-megjelenítés....65
Dolgozó tábla létrehozása táblatartalmak másolásával....69
Gomb (Button)....81
A gomb létrehozása a Komponens nézetben.....81
A gomb létrehozása a Fa nézetben.....85
3. fejezet Saját alkalmazás készítésének folytatása....88
Interaktív riport létrehozása.....89
Lap törlése....98
Navigáló gomb létrehozása....99
Fül utólagos létrehozása és megjelenítése....101

4. fejezet. Elemek (Items) ... 106
Elemek típusai ... 106
Hivatkozás Elemekre ... 114
Új Elem létrehozása ... 114
Elemek külalakjának megváltoztatása ... 117
Paraméterezett (dinamikus) riport készítése ... 118
Tartományok helyzetének változtatása ... 124
Riport fejlécének módosítása ... 126
"no data found" hibaüzenet kiküszöbölése. ... 127
Felületi Minta megváltoztatása ... 128
Tábla megjelenítése szövegdobozban ... 134
5. fejezet. Folyamatok létrehozása ... 136
Elágazás (Branches) ... 136
Aktivizálási Pont és Tevékenység ... 136
Az Elágazás (`Branch`) feltétele: ... 136
Az aktivizálási pont (`Branch Point`) lehet: ... 136
Listából kiválasztott tábla megjelenítése (példa) ... 137
Munkamenet (Session State) ... 143
Kiszámító Folyamatok (Computations) ... 145
Alkalmazászintű Kiszámító Folyamat ... 146
Lapszintű Kiszámító Folyamat létrehozása ... 146
Általános Folyamatok (Processes) ... 149
Példa (Általános Folyamatokra) ... 150
Computation és Process példa ... 154
6. fejezet. SQL utasítások futtatása alkalmazásból ... 162
Dinamikus SQL utasítás ... 162
OPEN-FOR, FETCH, CLOSE utasítások használata ... 168
Kurzorváltozók ... 168
Parancsfuttatás (DML, DDL) ... 172
Dinamikus Táblalétrehozás ... 175
7. fejezet. Adatbevitel-adatmódosítás ... 183
Új alkalmazás készítése ... 183
Az elemek (Items) jellemzőinek megváltoztatása ... 185
Adatbevitel ... 187
Adatrekordok szerkesztése ... 188
Feliratok magyarítása ... 191
Kapcsolat a két lap között ... 192
Adatbevitel már létező alkalmazáshoz ... 197
Trigger létrehozása ... 205
8. fejezet. Adatszintek közötti lefúrás ... 206
Készítsünk egy új alkalmazást ... 206
Készítsük el az első munkahely lapot ... 207
Készítsük el a második lapot ... 208
Rejtett Elem hozzáadása ... 209
Feltétel hozzáadása ... 209
Összekapcsolás ... 210
Fejléc magyarítása ... 212

9. fejezet. Táblázatos Forma készítése 214
Az `empno` (elsődleges kulcs) érték automatikus beállítása....215
Oszlop-lista választása....217
Táblázatos Forma (Tabuláló Form) hozzáadása egy alkalmazáshoz....220
Adatbeviteli trigger létrehozása a rekord azonosítására....226
10. fejezet. Hitelesítés és engedélyezés 229
Felhasználó létrehozása 231
Az alkalmazás hitelesítésének megváltoztatása....234
Engedélyezési osztály létrehozása 235
Példa 236
Fül letiltása 241
Felhasználói Csoport kezelése....245
11. fejezet. Összefoglaló feladat....248
Megoldás 249
A `Bemutatkozo` lap létrehozása 249
A `Karbantarto` lap elkészítése....261
A `Karbantarto` lap külalakjának utólagos megváltoztatása....265
Felhasználói csoport, Felhasználó létrehozása 269
A biztonság beállítása....270

Bevezetés

Az APEX és az XE jellemzői:

- Egyszerű telepítés
- Egyszerű üzemeltetés
- Egyszerű fejlesztés
- Ingyenes online közösségi fórum
- Ingyen használható akár üzleti célból is
- Az ingyenes XE változatban elkészített alkalmazások változatlanul használhatók a professzionális adatbázis-kezelő rendszerekben, éás fordítva!

Ez az ingyenes program komponensként tartalmazza az Oracle Database 11g adatbázis-motort, valamint az Oracle Application Express web-fejlesztő eszközt. Ilymódon a felhasználó tud adatbázist kezelni lokálisan és a web-en is.
Jellemzői

- Egy új fejlesztési irány.
- Böngésző alapú alkalmazásfejlesztés az Oracle adatbázis számára.
- Az adatbázis központú fejlesztésekhez ad egy fejlesztő környezetet.

Az Oracle webes fejlesztő eszközei:

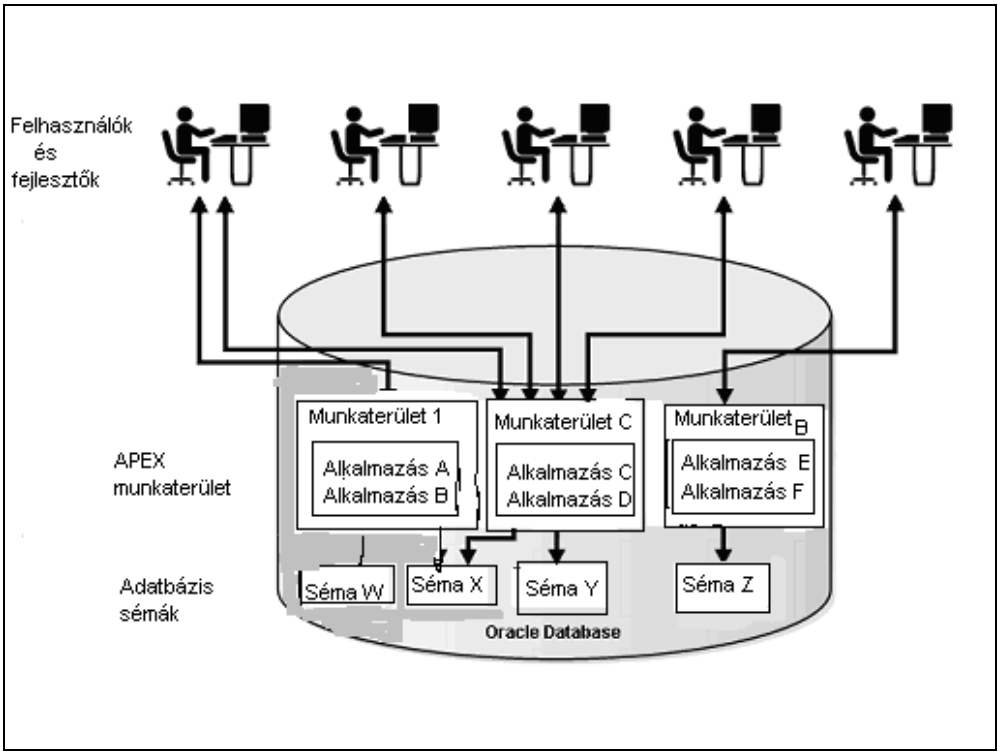
- a PL/SQL alapú Web Toolkit,
- az Application Express, mely az Oracle 11g Enterprise (Oracle fizetős) verziójának része, de az XE-ben is megtalálható (Application Builder néven),
- a Java alapú professzionális fejlesztések céljára a JDeveloper fejlesztő környezet.

Mivel az alábbiakban bemutatásra kerülő fejlesztő környezetek szemléletes grafikus felületeken keresztül használhatóak, ezért azt javasoljuk, hogy e könyvet olyan módon tanulmányozza a Tisztelt Olvasó, hogy a leírtakat mindjárt ki is próbálja az éppen tárgyalt fejlesztő környezetben.

Application Express

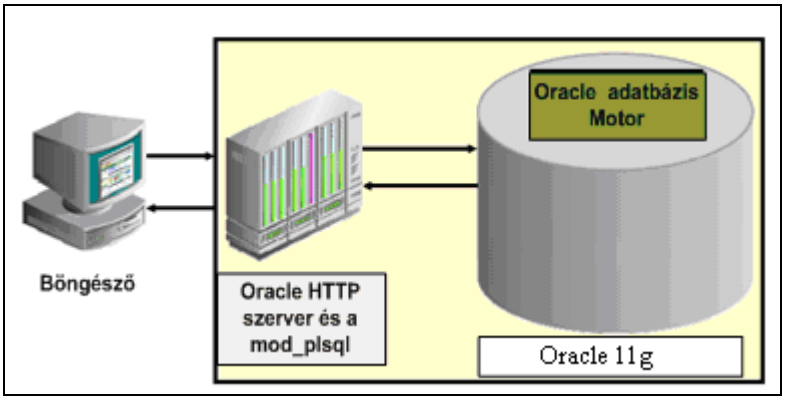
Az Application Express (APEX) az Oracle adatbáziskezelő rendszer PL/SQL nyelvi környezetébe ágyazott webes eszközkészletre (Web-Toolkit) épülő, úgynevezett vékonykliensű (webes) alkalmazások fejlesztésére szolgáló, az Oracle professzionális (pénzért

megvásárolható) adatbázis-motorját kiegészítő, de önmagában ingyenesen letölthető alkalmazásfejlesztő eszköz. E két eszköz (tehát az Oracle adatbázis-motor és az APEX) felhasználásából az Oracle cég létrehozott egy ingyenes, web-es adatbázis-alkalmazások létrehozására is alkalmas fejlesztő eszközt Express Edition (XE) néven. Az XE bizonyos (alapvetően az erőforrások méretére vonatkozó) korlátozásokon túl teljes értékű web-es adatbázis-alkalmazások létrehozására alkalmas. Az XE-ben a web-es alkalmazások létrehozására szolgáló (az APEX-szel ekvivalens) komponens neve Application Builder. Mivel a web-es alkalmazások létrehozásához az alapvető adatbázis-ismereten túlmenően ennek használata a legfontosabb, így az alábbiakban először az APEX-et mutatjuk be.
Az APEX program (ha nem az XE részeként használjuk Application Builder néven, akkor) az Oracle 11gR2 beépített része, melynek használatát azonban a telepítés során engedélyezni kell. Alkalmazása során használunk egy HTTP-szervert (web-szervert), amely lehet az Oracle beépített változata, vagy letölthető egy Apache webszerver címéről (például a http://httpd.apache.org címről), valamint a mod_plsqsl modult, mely szintén része az Oracle-nek.
Az Application Express megismerésére szolgáló ingyenes lehetőség az apex.oracle.com címen keresztül való belépés. Itt a felhasználók kapnak egy munkaterületet (workspace). Az ennek használatához szükséges jelszót a regisztrálást követően egy megadott email címre az Oracle küldi ki.



0.1. ábra ApplicationExpress alapfogalmai

A 0.1. ábra megmutatja a kapcsolatot a felhasználó, fejlesztő munkaterület és az adatbázis séma között. Sok felhasználó és fejlesztő eléri ugyanazt a munkaterületet, valamint egy felhasználó vagy fejlesztő különböző munkaterülettet elérhet. A munkaterület kapcsolódhat egy vagy több sémával.



0.2. ábra Apex architekturája

Az Application Express felépítését a 0.2. ábra mutatja, ahol a három architektúra réteg az alábbi:

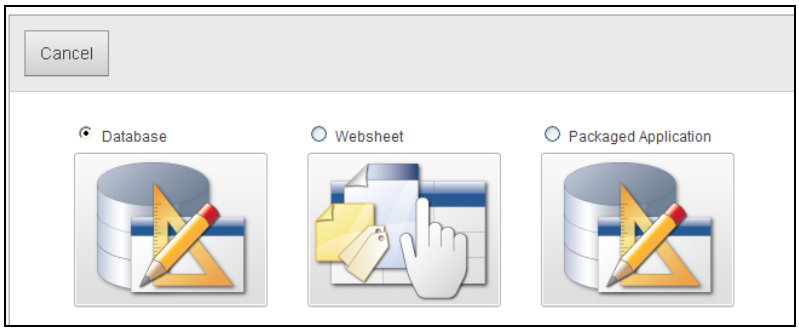
- A böngésző (`Browser`) megfelel a kliens rétegnek. A felhasználó a böngészőn keresztül a http web protokollt használva PL/SQL web fejlesztéssel éri el az adatbázist. Egy URL címet küld az adatbázis felé.
- Oracle HTTP szerver a Gateway (`mod_plsql`) programmal együtt adja a középső réteget. A PL/SQL alapú alkalmazásba beépül a Gateway program. Ez arra szolgál, hogy a böngészőtől érkezett html alapú kérést értelmezze és továbbítsa az adatbázis motor felé. A Gateway más néven `mod_plsql` program kétirányban kommunikál. Egyrészt értelmezi a klienstől érkező kérést, feldolgozza, és a PL/SQL programot továbbítja az adatbázis motor felé. Másrészt fogadja és továbbítja a az adatbázis szervertől érkezett választ.
- A PL/SQL program az adatbázis motoron (`Oracle DB-engine`) fut, majd visszaadja a választ, ami egy html kimenet, a középső rétegen keresztül a böngészőnek, amely azt megjeleníti.

A webes fejlesztés PL/SQL csomagok használatával történik. Ezek a csomagok az adatbázis-motorban vannak tárolva. Az adatbázist e csomagokon (htp, htf, és owa csomagokon) keresztül lehet elérni, és az adatok úgy jelennek meg, mint a generált web lap dinamikus adatai. E csomagok összefoglaló neve PL/SQL Web Toolkit.

Az Application Express előnyei

- Egyszerű az alkalmazásfejlesztő környezet.
- Egyszerű megtanulni annak, aki ismeri az SQL és PL/SQL programnyelveket.
- Egyszerű architektúra.

- Változatos sablonok és minták állnak rendelkezésre, lásd a 0.3. ábrát.
- Platform független környezet.



0.3. ábra. *A fejlesztőrendszer különböző fejlesztő felületei.*

Mit lehet ezzel az eszközzel készíteni?

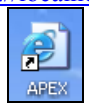
- Az Oracle adatbázissal interaktív kapcsolatban áll, menedzseli az objektumokat és új objektumokat is létre tud hozni.
- Webes alkalmazást készíthetünk.
- A webes alkalmazás biztonsága a jogosultság-kezeléssel (autentikáció) és felhasználói név – jelszó párossal történő biztonságos belépéssel (autorizáció) oldható meg.
- Megvalósítja az alkalmazáshoz rendelt funkciókat.

Application Express és az XE kapcsolata

A fentiek alapján tehát az Application Express egyszerűen elkészíthető, könnyen megtanulható, és változatos kinézetű webes adatbázis-alkalmazások készítésére használható.

Az XE fejlesztő környezetben a fentiekben bemutatott mindhárom architektúra réteg benne van, és a web-fejlesztés teljesen azonosan történik, mint az Application Express-ben, továbbá nagy előnye, hogy nincs szükség más programok telepítésére a működéséhez, és a használata ingyenes. Bár a mögötte lévő adatbázis maximum 4GB lehet, a kis- és középvállalkozások igényét ez bőven kielégíti. Kis erőforrás-igénye, a nagyon egyszerű telepítése, valamint a szintén nagyon egyszerű alkalmazás-fejlesztési lehetősége miatt a használata igen népszerű.

1. fejezet. A munkakörnyezet létrehozása

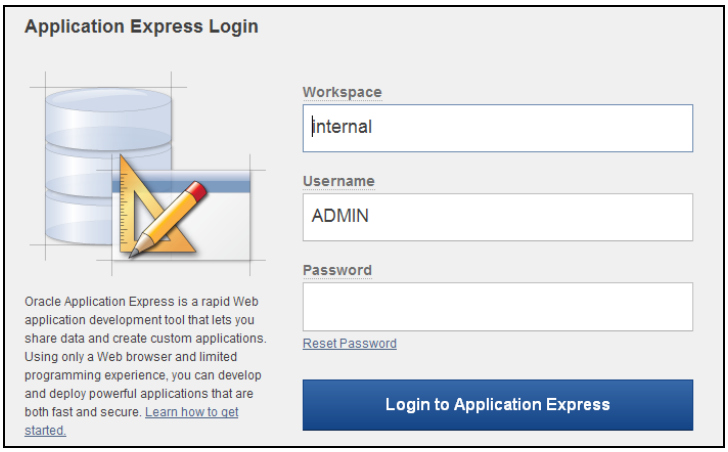
Az Oracle 11gR2, valamint az ehhez kapcsolódó APEX telepítési útmutatója a Mellékletben található.
A telepítést követően az APEX internetes alkalmazás indítása a http://localhost:8080/APEX címről történik. E címhez rendelhetünk egy parancsikont az asztalon: 

Ezután már csak erre kell kattintani, és az APEX elindul.

Az első bejelentkező ablak:

Lépjünk be rendszer-adminisztrátorként, ahol az adminisztráror jelszava `OraclE-0`.

Ezután kattintsunk a kék `Login to Application Express` gombra.



1.1.ábra *Belépés az Apex alapértelmezett adminisztrátorként*

Munkaterület, séma és felhasználó létrehozása

Mielőtt létrehoznánk a munka-környezetünket, ismerkedjünk meg az Application Express (APEX) alapvető fogalmaival.

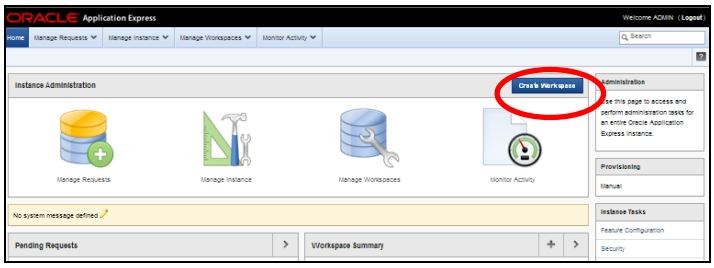
Először is rendszer-adminisztrátorként (administrator) létre kell hoznunk egy adatbázis-munkaterületet (`workspace`), majd ehhez az adatbázis-munkaterülethez hozzá kell rendelnünk egy gazdát (ő az adatbázis-felhasználó, a `database user`; vagyis a `schema`), és e munkaterület APEX-adminisztrátori és fejlesztői jogkörökkel is rendelkező felhasználóját (ez az APEX-adminisztrátor, vagyis az `APEX-admin user`), végül pedig létrehozhatunk olyan APEX-felhasználókat (`APEX-user`), akik nem rendelkeznek adminisztrátori jogkörrel (például nem módosíthatnak, és nem törölhetnek alkalmazásokat).
Nagyon fontos, hogy a séma (`schema`) és az `APEX-admin user` ugyanannak a dolognak két oldala.
A séma rendelkezik az adatbázishoz kapcsolódó jogkörökkel, ő az adatbázis "gazdája". Az adatbázis-használathoz kapcsolódó jogosultságokat (például könyvtárhasználat), illetve a séma törlését nem is lehet elvégezni az APEX környezetből, csak az SQL*Plus-ból (illetve az SQL-Developer-ből).
Az `APEX-admin user` viszont a séma által felügyelt adatbázis-területen történő APEX alkalmazások létrehozására vonatkozó jogosultságok "gazdája". Ő létrehozhat alkalmazásokat, törölheti is azokat, illetve létrehozhat olyan APEX-felhasználókat (APEX-end user), akik csak használhatják az egyes általa létrehozott alkalmazásokat, de azokat nem módosíthatják, illetve nem törölhetik.
Megjegyezzük, hogy egy munkaterületnek több gazdája is lehet, vagyis az adatbázis munkaterületek megosztottak!

Az APEX fejlesztői környezetének hozzáféréséhez a felhasználónak be kell jelentkeznie egy megosztott munkaterületre (`workspace`). A felhasználók számára négyféle szerepkört választhatunk:

- Fejlesztő (`Developer`): létrehozhat és szerkeszthet alkalmazásokat
- Munkaterület adminisztrátorok olyan felhasználók, akik elvégzik az adminisztrátori teendőket, menedzselik a felhasználói azonosításokat (account), figyelik a munkaterületet, és nézik a log fájlokat.
- Végfelhasználó alkalmazás felhasználó anélkül, hogy a fejlesztéshez hozzáférne, vagy adminisztrátori képessége lenne.
- Superuser példa (instance) adminisztrátor, használja az APEX szervíz alkalmazását, mint házigazda.

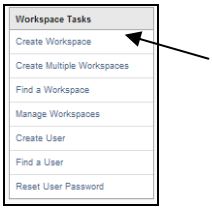
Miután a fejlesztőkkel tisztában vagyunk, hozzuk létre a munkaterületünket (work1), majd egy sémát (sema1), egy munkaterületi adminisztrátort, és ha akarunk, saját magunknak egy fejlesztőt.

A munka menete a következő:



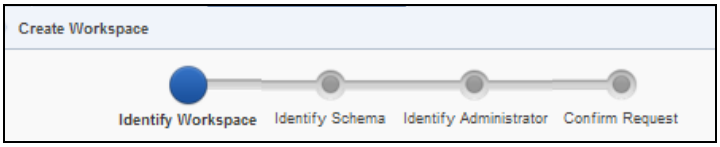
1.2. ábra. *Az APEX adminisztrátor bejelentkező ablaka*

Kattintsunk a jobb felső kék gombra vagy a jobb szélen levő taskok közül a `Create Workspace` menüpontra (1.3. ábra). A Munkaterületünk neve legyen `Work1`.

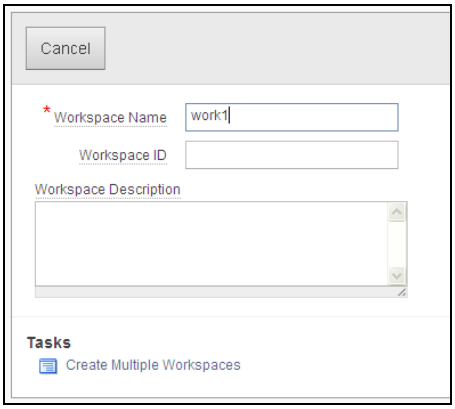


1.3. ábra. *A create workspace indítása taskból*

Menjünk végig a varázslón.

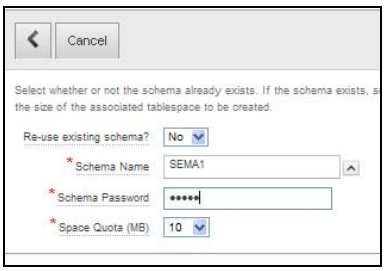


1.4. ábra *A munkaterület létrehozásának folyamata.*



1.5. ábra *A munkaterület nevének megadása*

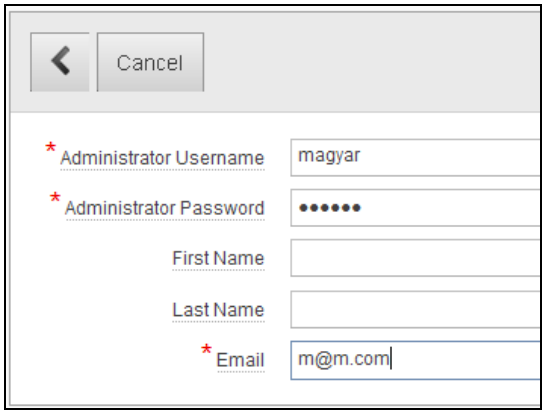
Az első ablakba írjuk be a munkaterületünk nevét (1.5. ábra), legyen `work1`, majd kattintsunk a `Next` gombra.



1.6. ábra Sémanév és quota megadása.

Már létező séma helyett egy újat hozzunk létre, melynek neve legyen `sema1`, és a jelszava is legyen `sema1`. A quota megadását változtathatjuk. (1.6. ábra). `Next`

Következő lépésben meg kell adnunk annak a felhasználónak a nevét, aki minden fejlesztési munkát elvégezhet az internetes fejlesztői környezetben. Legyen e felhasználó neve `magyar`, akinek jelszava `tigis`, és bármilyen email címet megadhatunk, hiszen nem kívánjuk igénybe venni a supportot. `Next`.



1.7. ábra *Adminisztrátori fejlesztő*

Megjegyezzük, hogy esetenként előfordulhat APEX-indítási hiba (például ha nem elég gyors a számítógépünk). Ekkor a teendő: Vezérlőpult – Felügyeleti eszközök – Szolgáltatások, és e listából újraindítás.

You have requested to provision a new Workspace.

Workspace Information:

Name	work1
Security Group ID	System Assigned
Description	...

Administrator Information:

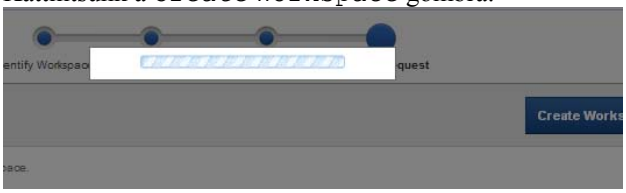
User Name	magyar
E-mail	m@m.com

Schema Information:

Reuse Existing Schema	No
Schema Name	SEMA1
Tablespace will be created	APEX_XXX
Datafile for tablespace	D:/ORACLE/ORADATA/ORCL/APEX_XXX.DBF

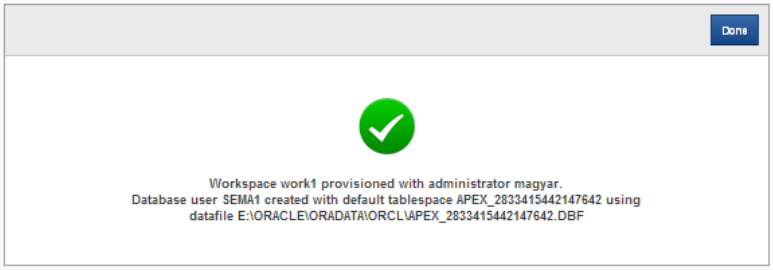
1.8.ábra. *A létrehozandó munkaterület, séma és fejlesztő felhasználó összesítése.*

Kattintsunk a `Create Workspace` gombra.



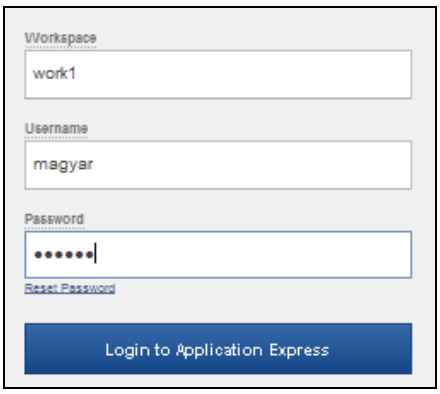
1.9. ábra *Dolgozik a létrehozáson.*

A létrehozás némi időt igénybe vesz.
A létrehozás után megjelenik a következő ablak.



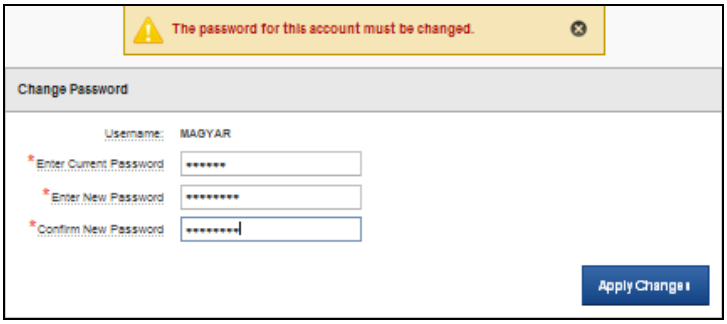
1.10.ábra létrehoztuk a munkakörnyezetünket.

Kattintsunk a `Done` gombra, és a megjelenő ablakban minden fontos információt megtalálunk a használatunkban lévő internetes fejlesztői környezet munkaterületeire, sémáira, felhasználóira vonatkozóan. Innen lehet létrehozni törölni a felhasználókat és a munkaterületeket (Például olyan munkaterületet és felhasználót, amelyik hiba miatt nem jött létre, de a rendszer érzékeli.) Lépjünk ki a rendszergazda hatóköréből (`Logout`) és lépjünk be (`Return to Login Page`) `magyar`-ként. (1.11. ábra)



1.11. ábra Belépés aktuális felhasználóként.

Munkaterület (`workspace`) : work1
Felhasználónév (`username`): magyar
Jelszó (`password`) tigris
Kattintsunk a `Login....` gombra
Sajnos ez az egyszerű jelszó nem megfelelő az APEX számára ezért meg kell változtatnunk.



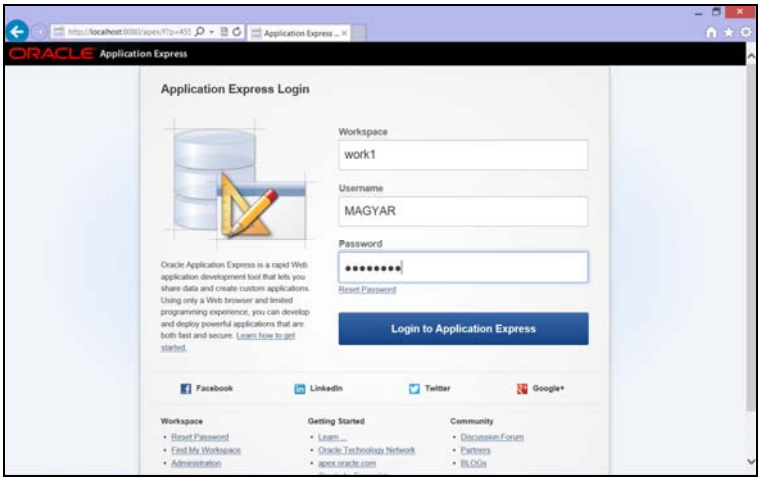
1.12. ábra *Jelszóváltoztatás*

Felhasználónév magyar
Aktuális jelszó : tigris
Új jelszó: Tigris-1
Mégegyszer az új jelszó: Tigris-1
(Az adminisztrátor jelszavában kis és nagybetűk, számok és különleges karakterek is kellenek.)
Érvényesítsük a változásokat (`Apply Changes`), majd kattintsunk a `Return` gombra és lépjünk be a felhasználói felületre.

1.13. ábra *Sikeres jelszó változtatás*

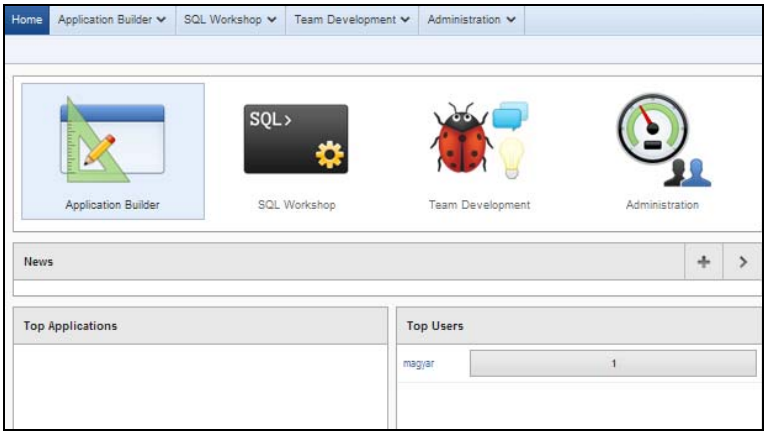
A felhasználó főablakának komponensei

munkaterület :	work1
felhasználó név:	magyar
jelszó:	Tigris-1



1.14. ábra *Belépés* ***magyar/Tigris***-1 *jelszóval*

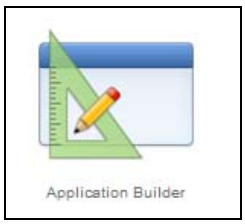
Kattintsunk a `Login...` gombra.
Megjelenik az APEX fejlesztői felülete.



1.15. ábra. *A felhasználó lehetőségei*

Az alábbiakban röviden áttekintjük a főablak komponenseit.

- `Application Builder` továbbiakban AlkalmazásVarázsló, mellyel webes alkalmazásokat lehet létrehozni.



1.16.ábra *Az alkalmazás Varázsló*

- `Az SQL workshop` (SQL, PL/SQL felület) SQL műhely.

1.17. ábra *Az SQL ikonja*

- `Team development` komponens. Segítségével alkalmazásokat lehet csoportmunkában fejleszteni.

1.18. ábra *Csoportmunka ikonja*

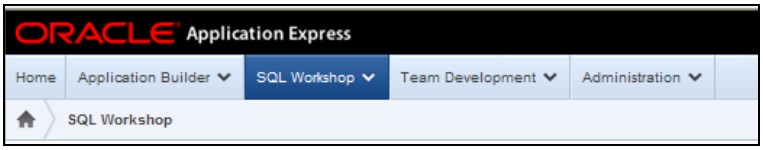
- `Administration`. Az adminisztrátor segítségével csak abban az esetben tudunk dolgozni, ha megfelelő jogosultságunk van. Jelen esetben a `magyar` felhasználónak adminisztrátori jogot adtunk.Tehát az `Administration` komponens segítségével az APEX-adminisztrátori tevékenységek (felhasználó menedzselés, jogosultságkezelés, stb.) elvégezhetők.

1.19.ábra *Az adminisztrációs tevékenység ikonja*

A komponensek és a Funkció Sáv használata

Minden komponenst úgy indítunk, hogy rákattinthatunk a komponens képére (az ikonra), Ha rákkatintunk a komponensre, akkor egy megnyíló ablakban újabb komponensek jelennek meg. A menünél pedig újabb menük.

Az eredeti komponensek sorozata jól látható módon jelenik a legfelső sorban. Alatta megjelenik az úgynevezett Visszalépő Sáv (`Breadcrumb`), amelyiknél a Home mindig a kiinduló állapotot jelzi valamint az utat, amelyen mindig a megfelelő ablakba vissza lehet lépni.



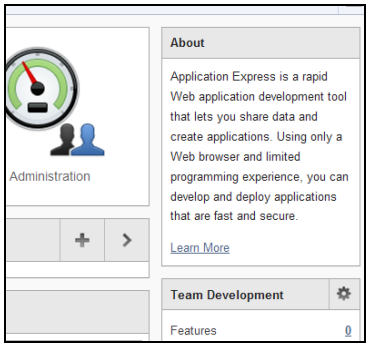
1.20. ábra. *Funkció Sáv*

Ha az aktuális komponens ablakból vissza szeretnénk lépni egy korábbi komponens szintre, akkor egyszerűen csak rá kell kattintani a Funkció Sáv komponens név sorozatában szereplő megfelelő komponens nevére. A legelső „komponens” természetesen a `Home`.

Felhívjuk a figyelmet, hogy a komponens-szintű visszalépésre csak ezt a Visszalépő Sávot használjuk, az alkalmazott böngésző program visszalépés gombját erre a célra soha **NE HASZNÁLJUK**.

Megjegyzés

Figyeljünk fel a képernyők jobb szélén lévő ablakokra, mivel ezek fontos információkat, webes eléréseket és adatokat mutatnak meg. Például az 1.15. ábrán bemutatott részképernyő jobboldali része:

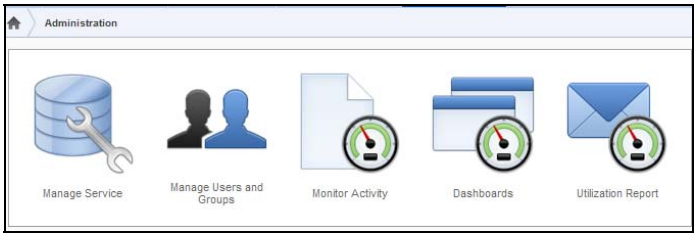


Itt láthatjuk azokat a hiperhivatkozásokat, amelyeket erről a lapról közvetlenül elérhetünk (pl. a Learn more, illetve a Documentation hivatkozások).

Adminisztrátori (fejlesztői) teendők

Kattintsunk az adminisztrátori teendők képére.

FEJLESZTŐ FELHASZNÁLÓ LÉTREHOZÁS



1.21. ábra. *Az adatbázis adminisztrátori teendők*

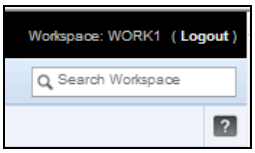
Itt különféle statisztikákat is láthatunk (1.21. ábra). Fontosabbak a taszkok, melyek jobb oldalon találhatók (1.22. ábra). Ezek közül néhány:

- megváltoztathatjuk a jelszót,
- menedzselhetjük a felhasználói csoportokat,
- létrehozhatunk felhasználót,
- törölhetünk felhasználókat.

Jelszó változtatás
Az application expressről
Felhasználói csoport menedzselése
Felhasználó létrehozása
Több felhasználó létrehozása
Több felhasználó törlése
Interaktív riport szervezése
Beállítások
Az exportált adatraktár szervezése
Adatbázis weblapok objektumai

1.22.ábra. *Az adatbázis adminisztrátori teendők a taskban*

Végül lépjünk is ki: `Logout`. (1.23. ábra)

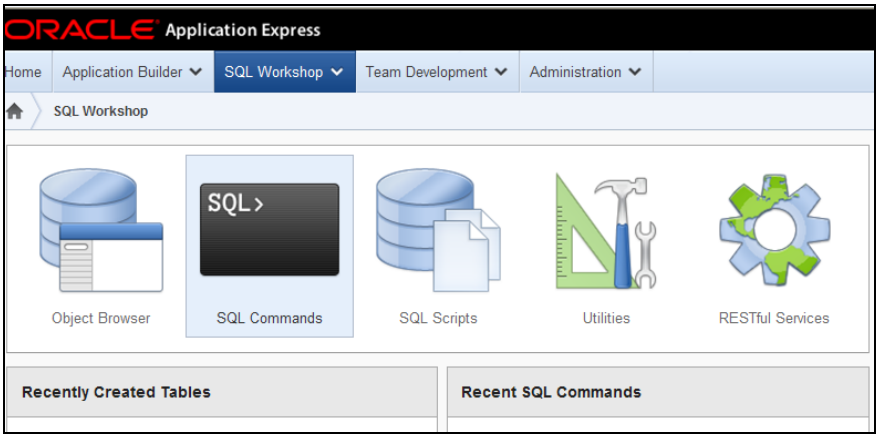


1.23. ábra *Kijelentkezés: Logout. Általában jobbra fent.*

Az APEX eszközeinek bemutatása

Az SQL műhely (SQL workshop)

Az SQL műhely egy olyan komponens, amely egy webböngészőn keresztül interaktív kapcsolatot enged meg az adatbázissal (1.24. ábra).



1.24. ábra *Az SQL műhely tevékenységei*

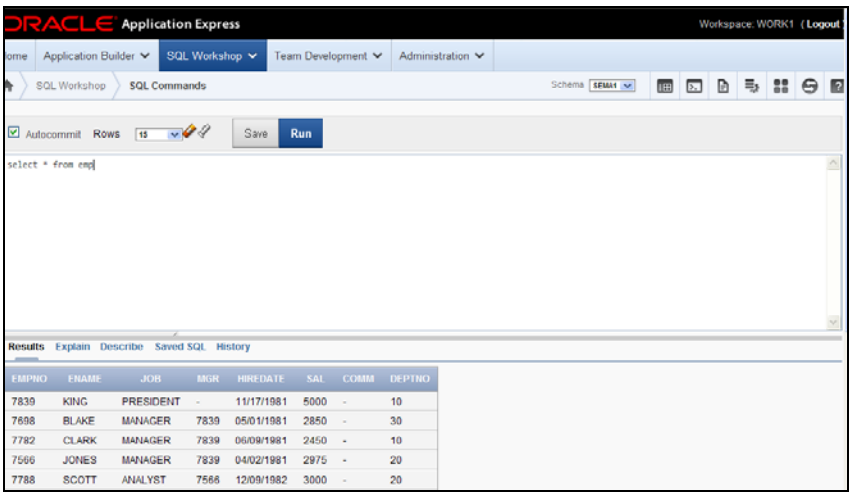
Az itt elvégezhető főbb feladatok:

- adatbázis objektumok megtekintése, kezelése (`Object Browser`),
- SQL utasítások és PL/SQL programok futtatása (`SQL Commands`),
- SQL szkriptek szerkesztése, betöltése, mentése és futtatása (`SQL Scripts`),
- adatbázis objektumokból (tábla, nézet, stb.) szemléletes lekérdezések előállítása (`Query Builder` a `Utilities`-ben),
- adatok betöltése és kimentése (`Data Workshop` a `Utilities`-ben).

Az SQL Commands komponens

Az `SQL Commands` komponenst használjuk az SQL utasítások és a PL/SQL blokkok megírására, szerkesztésére, futtatására és törlésére.(1.25. ábra) E komponens ablaka két részből áll, a felső részen található a szerkeszthető utasításablak, az alsón pedig az eredményablak egy hozzátartozó menüsorral.

A utasításablak felső részén bejelölhető az `Autocommit` jelölőnégyzet (mellyel a DML-utasítások automatikusan véglegesíthetőek), beállítható az eredményablakban megjelenő sorok száma (`Rows`), a parancsok törlése (`CommandClear`), a táblák keresése (`Find Tables`), és itt található a mentés (`Save`), és a futtatás (`Run`) ikonja.



1.25. ábra Az *Sql commands felülete*

A visszalépő sávban a következő ikonok láthatóak. Baloldalon az a komponens-sorozat, amelyen keresztül elértük az `SQL Commands` ikon felületét, jobboldalon pedig az SQL Műhely további lehetőségei, melyek az alábbiak:

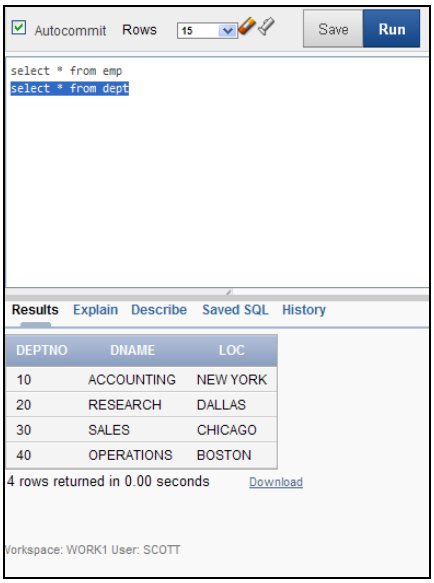
-

kiválasztható, hogy melyik sémában dolgozunk
- az objektum varázsló
- az SQL parancs ablak
- az SQL szkript futtató és szerkesztő ablak
- a Lekérdezés varázsló (`Query Builder`)
- a segédprogramok (`Utilities`)
- `RESTful services`
- Segítség!

TÁBLA LISTÁZÁSA

Rákattintással válasszuk ki az `SQL Commands` komponenst. Írjunk be az utasításablakba egy `SELECT` utasítást, majd a `Run` gombra való kattintással, vagy a `CTRL+Enter` billentyű-kombináció lenyomásával futtassuk le. Az eredményablak jeleníti meg lekérdezés listájából annyi sort, amennyit beállítottunk, vagy a hibaüzenetet (ha a beírt utasítás nem hajtható végre, mert hibás). Egyszerre csak egy utasítás hajtható normál módban végre, ebben az esetben az utasítás végére nem kell pontosvessző. Annyiben egyszerre több utasítás található

az utasítás ablakban, akkor **egy** utasítást jelöljünk ki, és a `Run` gombra való kattintással ez lesz végrehajtva.



1.26. ábra. *Az emp tábla megjelenítése.*

Az alábbiakban bemutatjuk az `SQL Commands` eredményablakának menüsorát.

EXPLAIN MENÜPONT

Itt megtekintjük az SQL utasítások végrehajtási tervét. A végrehajtási tervből egyebek mellett látszik, hogy az adatbázis motor milyen lépések segítségével, milyen objektumokon keresztül (táblák, indexeket, stb.) hajtja végre az SQL utasítást. Ilymódon a teljesítmény problémák mellett, akár a hibás tábla-összekapcsolás (`join`) is tettenérhető.

Nézzük meg például az `SQL Commands` komponens utasításablakába írt, dolgozók és telephelyük listáját megjelenítő `SQL` utasítás (1.27. ábra) végrehajtásának értelmezését (`Explain`, 1.28. ábra). A `Query Plan` táblázatból például kiolvasható, hogy az eredménylista – helyesen – 14 sorból áll, amely a `dept` táblával van összekapcsolva. Láthatók az oszlopok indexei, a séma és a táblák oszlopai. Mindezekből az utasítás hibáit könnyebben kideríthetjük.

Autocommit Rows 15 Save Run

```
select  d.ename as Dnév,
        loc as Telephely
from  emp d, dept r
where  d.deptno = r.deptno
```

Results Explain Describe Saved SQL History

Query Plan

Operation	Options	Object	Rows	Time	Cost	Bytes	Filter Predicates *	Access Predicates
SELECT STATEMENT			14	1	6	280		
MERGE JOIN			14	1	6	280		
TABLE ACCESS	BY INDEX ROWID	DEPT	4	1	2	44		
INDEX	FULL SCAN	SYS_C0017788	4	1	1			
SORT	JOIN		14	1	4	126	"D"."DEPTNO" = "R"."DEPTNO"	"D"."DEPTNO" = "R"."DEPTNO"
TABLE ACCESS	FULL	EMP	14	1	3	126		

* Unindexed columns are shown in red

Index Columns

1.27. ábra *Az explain menüpont*

Owner	Table Name	Index Name	Used in Plan	Columns	Uniqueness	Status	Index Type	Join Index
SEMA1	DEPT	SYS_C0017788	✓	DEPTNO	UNIQUE	VALID	NORMAL	NO
	EMP	SYS_C0017790		EMPNO	UNIQUE	VALID	NORMAL	NO

Table Columns

Table Owner	Table Name	Column Name	Data Type
SEMA1	DEPT	DEPTNO	NUMBER
		DNAME	VARCHAR2
		LOC	VARCHAR2
	EMP	EMPNO	NUMBER
		ENAME	VARCHAR2
		JOB	VARCHAR2
		MGR	NUMBER
		HIREDATE	DATE
		SAL	NUMBER
		COMM	NUMBER
		DEPTNO	NUMBER

1.28. ábra. *A kiadott SQL-utasítás értelmezése*

DESCRIBE MENÜPONT

A `Describe` menüpontra kattintva, alapértelmezésben megkapjuk a táblaszerkezetet megjelenítő `DESCRIBE` parancs használatának leírását. Ha azonban ezt az utasítást ki is adjuk az utasításablakban, akkor az eredményablak automatikusan a `Describe` menüpont szerint jelenik meg, és tartalmazza az utasításban kijelölt tábla szerkezetét (1.29. ábra).

A `Describe` menüpont hatása az `emp` tábla esetén:

Results Explain Describe Saved SQL History

Object Type TABLEObject EMP

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMP	EMPNO	NUMBER	-	4	0	1	-	-	-
	ENAME	VARCHAR2	10	-	-	-	✓	-	-
	JOB	VARCHAR2	9	-	-	-	✓	-	-
	MGR	NUMBER	-	4	0	-	✓	-	-
	HIREDATE	DATE	7	-	-	-	✓	-	-
	SAL	NUMBER	-	7	2	-	✓	-	-
	COMM	NUMBER	-	7	2	-	✓	-	-
	DEPTNO	NUMBER	-	2	0	-	✓	-	-

1 - 8

1.29. ábra. *A Describe menüpont hatása*

SAVED SQL MENÜPONT

Az utasításablakban az `SQL` utasítások felhasználó által megadott néven elmenthetők (a `SAVE` gomb segítségével), majd a `Saved SQL` menüpont, a `Find` kereső-mező és a `Go` gomb segítségével előhívhatók. A megfelelő utasítás hiperlinkjére kattintva, az futtatható, megváltoztatható, és újra elmenthető, illetve akár törölhető is. (1.30. ábra).

Results Explain Describe Saved SQL History

Owner SCOTT Find proba Rows 20 Go Delete Checked

Owner	Name	Description	SQL	Updated By	Updated
SCOTT	proba	-	select loc, ename from emp, dept where emp.deptno = dept.deptno	SCOTT	Now

1.30. ábra. *A Saved SQL hivatkozás* `lapja`

HISTORY MENÜPONT

E menüpont kiválasztása esetén az eredményablakban az aktuális felhasználó (séma) összes, az `SQL Commands` ablakban végrehajtott utasítása látható. Ezeken a `Find` mezőben a korábbi utasítások tetszőleges alsztringje szerint szűrés is végezhető. E listából esetenként egyszerűbb egy már megírt `SQL` utasítást megjeleníteni, mint azt ismételten beírni (1.31. ábra). Amelyik utasítást kijelöljük (kattintással), azt betölti az utasításablakba, ahol azt módosítani, futtatni, illetve elmenteni egyaránt lehet.

Results Explain Describe Saved SQL History

Find

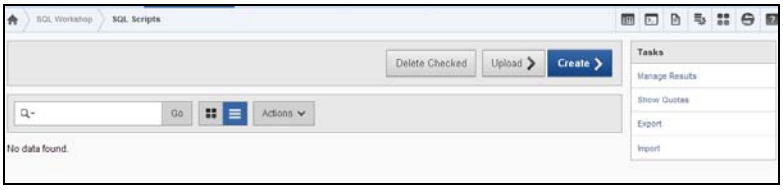
Time	SQL	Schema
2 seconds ago	select * from dept	SEMA1
3 minutes ago	select * from emp	SEMA1
3 minutes ago	select * from emp	SEMA1
13 minutes ago	select loc, ename from emp, dept where emp.deptno = dept.deptno	SEMA1

1.31. ábra. *Az SQL Commands komponensen történt futtatások története*

Az SQL Scripts komponens

Az `SQL műhely` harmadik eleme az `SQL Scripts` komponens. Ennek segítségével lehet egy szkript programot betöltetni, szerkeszteni, majd futtatni, vagy az APEX-ben elkészített szkripteket elmenteni a háttértáron.

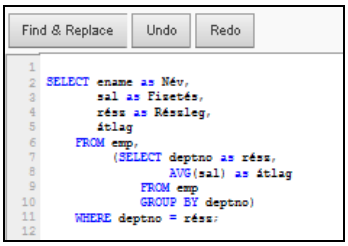
Ezt egy proba.txt nevű fájl segítségével mutatjuk be. Indítsuk el az SQL script komponenst. (1.32. ábra)



1.32.ábra *Az SQL szkript komponens*

SZKRIPT FÁJL KÉSZÍTÉSE ÉS SZERKESZTÉSE

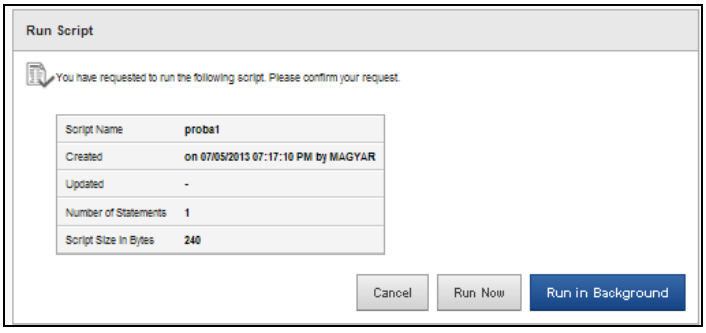
Készítsünk egy szkript fájlt. Kattintsunk a jobb oldali `Task` ablakban a `Create` gombra. Írjuk meg a programot (1.33.ábra).



Listázzuk a dolgozók nevét, fizetését, és a részlegükben dolgozók átlagfizetését. (1.33.ábra)

1.33.ábra Szkript fájl írása, szerkesztése

Futtassuk le (`Run`) (1.34.ábra).

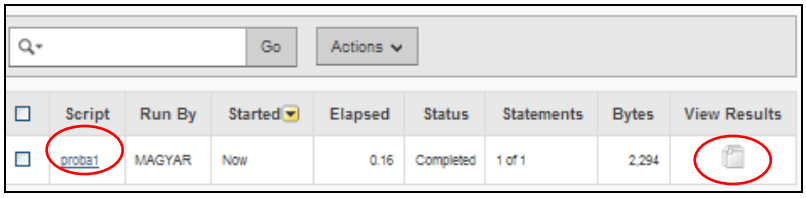


1.34. ábra Szkript fájl futtatása

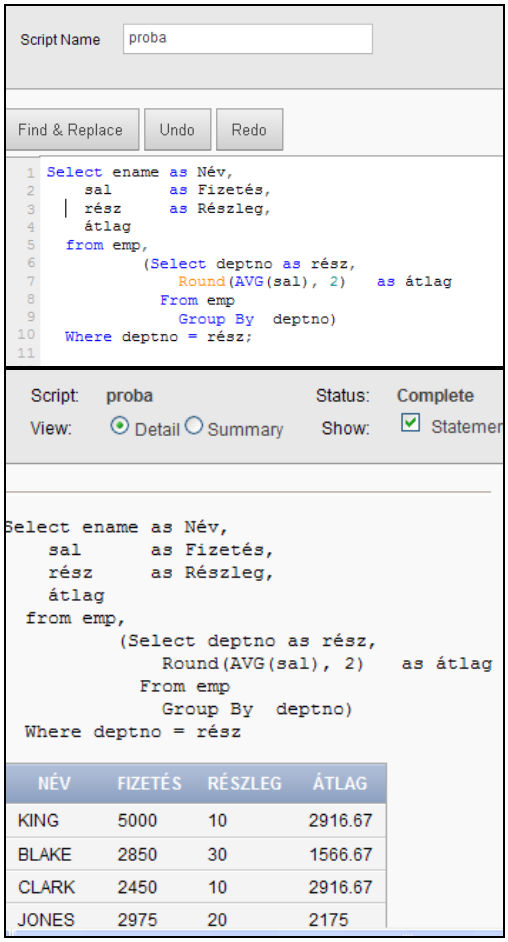
Ha mindent rendben találunk, akkor kattintsunk a `Run Now` gombra.

Nézzük meg az eredményt, kattintsunk a `View Result` ikonra, majd a `Details` rádiógombot jelöljük ki (Ez utóbbi segítségével nem csupán az egyes utasítások eredménye, de még a futásukhoz szükséges idő is megjelenik.). Az eredmény láthatóan jó, csak a kerekítés

hiányzik, kattintsunk az `Edit Script` gombra, és írjuk az átlag elé a `Round` függvényt. vagy menjünk vissza az SQL Script felületre és az `Edit` oszlopban kattintsunk a négyzetben levő ceruzára.



1.35.ábra *A proba1 szkript fájl eredményének megnézése*



Szerkesztés

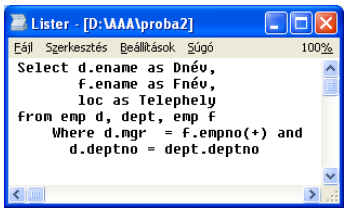
1.36. ábra *Szkript futtatása és eredménye.*

Láthatjuk, hogy a szkript sikeresen lefutott. Bár a képernyő jobb oldalán van egy Export menüpont, mellyel a szkript program text fájlba menthető, az így létrehozott szövegfájl azonban az APEX környezetre vonatkozó beállításokat is tartalmazza, ezért ehelyett célsze-

rűbb a hagyományos Windows-os (kijelölés, Ctrl-C, Ctrl-V) másolással kitenni egy szöveg-fájlba.

SZKRIPT FÁJL BETÖLTÉSE ÉS FUTTATÁSA

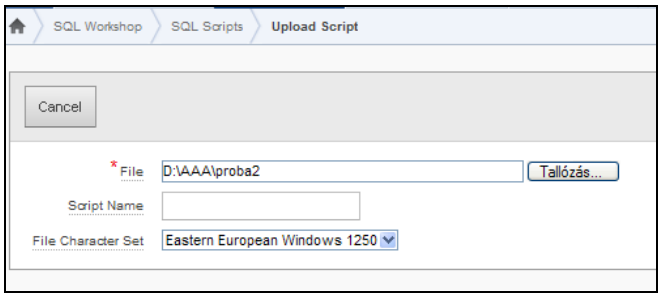
Készítsünk egy szkript fájl, melynek neve legyen proba2.txt, és amely tartalmazzon egy Select utasítást (1.37.ábra).



Litázza a dolgozó nevét a dolgozó főnökének nevét és telephelyüket.

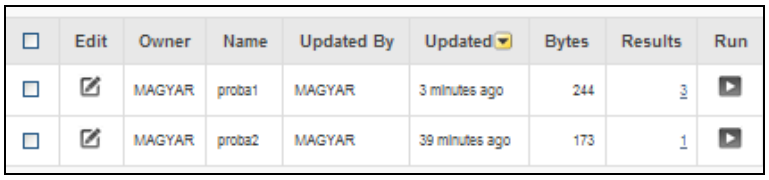
1.37. ábra *Lekérdezés*

Kattintsunk a SQL Script fájl fülre vagy az SQL műhely bármelyik eszközének jobb oldalán található SQL script ikonra. Töltsük be tallózással ezt a fájlt, és válasszuk a Kelet-Európai Windows karakterkészletet, majd kattintsunk az `Upload` gombra (1.38.ábra).



1.38. ábra *Betöltés (Upload)*

Ezek után futtassuk le e szkript programot először a `Run` oszlopbeli ikonra (1.39. ábra), majd a felugró ablak `Run Now` gombjára való kattintással.



Edit	Owner	Name	Updated By	Updated	Bytes	Results	Run
	MAGYAR	proba1	MAGYAR	3 minutes ago	244	3	
	MAGYAR	proba2	MAGYAR	39 minutes ago	173	1	

1.39. ábra Szkript fájlok

Ha nincs hiba, nézzük meg az eredményt. Most más úton menjük oda. Kattintsunk a megfelelő szkript programhoz tartozó `View Results` oszlopban lévő ikonra, és nézzük meg az eredményt. Láthatóan az eredmény jó (1.40. ábra).

DNÉV	FNÉV	TELEPHELY
BLAKE	KING	CHICAGO
JONES	KING	DALLAS
CLARK	KING	NEW YORK
ALLEN	BLAKE	CHICAGO
JAMES	BLAKE	CHICAGO
TURNER	BLAKE	CHICAGO
MARTIN	BLAKE	CHICAGO
WARD	BLAKE	CHICAGO
MILLER	CLARK	NEW YORK
SCOTT	JONES	DALLAS
FORD	JONES	DALLAS
ADAMS	SCOTT	DALLAS
SMITH	FORD	DALLAS
KING	-	NEW YORK

14 rows selected. 0.02 seconds

1.40. ábra

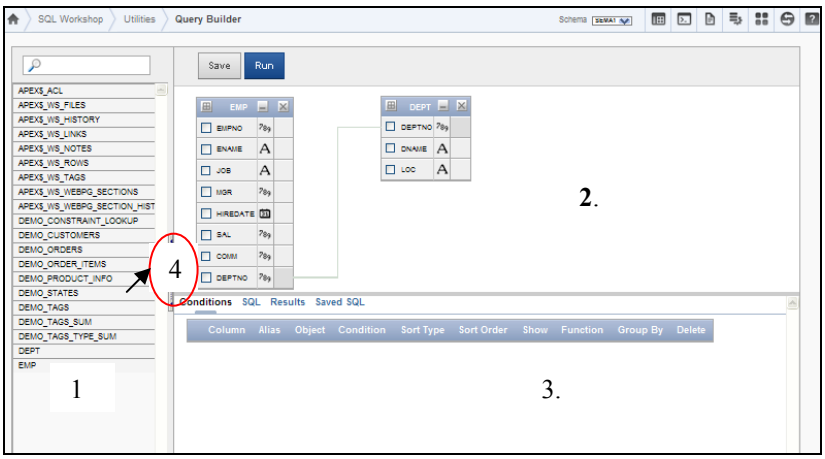
A Query Builder komponens

Az `SQL` műhelyben található a Lekérdezés Varázsló (`Query Builder`) komponens. Segítségével egyszerűbb lekérdezéseket lehet grafikus módon összeállítani.

E funkció az SQL műhelyen belül bármelyik komponens jobb oldali kis ikonjával, vagy pedig a Utilities eszközön belül a

komponens indításával elérhető.
A hozzátartozó ablak négy részből áll (1.41. ábra):

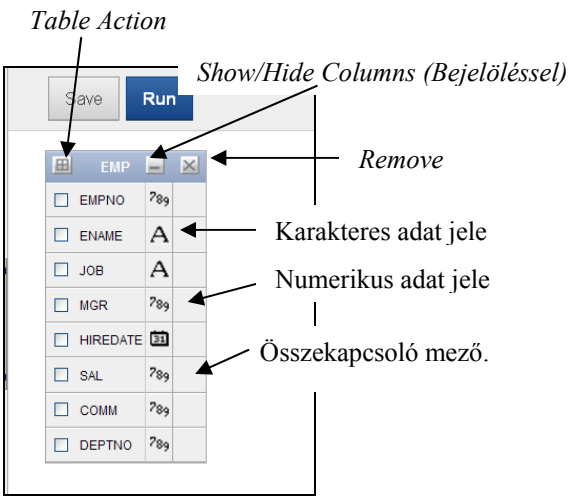
1. Az Objektum Ablak, melyből az elkészítendő lekérdezés objektumai választhatók ki.
2. A Grafikus Tervező Ablak, melyben a kiválasztott objektumok grafikusan összekapcsolhatók. Két funkcionális Eleme az Objektum Táblázat és az objektumok közötti kapcsolatot jelző Kapcsolati Vonal. A Grafikus Tervező Ablak jobb felső részében az összeállított lekérdezés (`SELECT` utasítás) elmentésének (`Save`) és futtatásának (`Run`) gombjai találhatók.
3. Szerkesztő Ablak, melynek menüsora az alábbiakat tartalmazza:
   - `Condition`: itt szerkeszthető a lekérdezés feltételrésze,
   - `SQL`: megjeleníti az összeállított `SELECT` utasítást,
   - `Results`: megjeleníti a lefuttatott lekérdezés eredményét,
   - `Saved SQL`: segítségével az elmentett lekérdezés visszatölthető.
4. Felosztó Léc: erre kattintva eltüntethető az Objektum Ablak, és egyúttal megnő a Grafikus Tervező és a Szerkesztő Ablak. Az erre történő ismételt kattintás hatására visszaáll az eredeti ablakszerkezet.



1.41. ábra. *A QueryBuilder tervező ablaka*

A LEKÉRDEZÉS VARÁZSLÓ (QUERY BUILDER) HASZNÁLATA

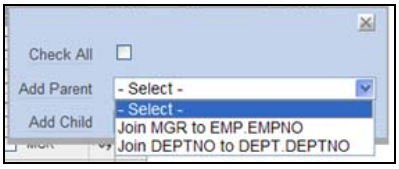
Első lépés az objektum kiválasztása. A kiválasztott objektum nevére kattintva, az átkerül egy úgynevezett Objektum Táblázatként a Grafikus Tervező Ablakba. Az Objektum Táblázatban be lehet jelölni, hogy a tábla mely oszlopai jelenjenek meg a lekérdezésben, illetve, hogy mely oszlopok segítségével valósuljon meg a táblák összekapcsolása. Az 1.41. ábra mutatja a betöltött objektum szerkezetét, vagyis az Objektum Táblázatot grafikus módon. Ez két részből áll; az Objektum Fejlécből, valamint a tábla egyes oszlopainak neveit és tulajdonságait tartalmazó Objektum Sorokból:



1.42. ábra. A grafikus tervező megjelenő táblázat.

Az Objektum Fejléc az alábbiakat tartalmazza

Table Action: Segítségével a tábladefinícióban már szereplő referencia és idegen kulcs megszorítások alapján táblakapcsolatok hozhatók létre, valamint kijelölhető az összes oszlop megjelenítésre, illetve ez visszavonható (`Check all`) (1.43. ábra). Például az `emp` tábla Objektum Táblázatán a *Table Action* gombra kattintva a felugró ablak első `Join` utasítását kijelölve, megjelenik az `emp_1` Objektum Táblázat és a két táblázat (korrelált) összekapcsolása.



1.43. ábra. *Felugró ablak a Table Action gombra kattintáskor*

Táblanév: Az automatikusan generált alias neve, ha még egy emp táblát felteszünk a grafikus felületre (`emp` esetén például `emp_1`).

Show/Hide Columns: Segítségével megjeleníthetőek/elrejthetőek az Objektum Sorok.

Remove: Segítségével törölhető egy tábla a grafikus tervező ablakból.

Az Objektum Sorok az alábbiakat tartalmazzák

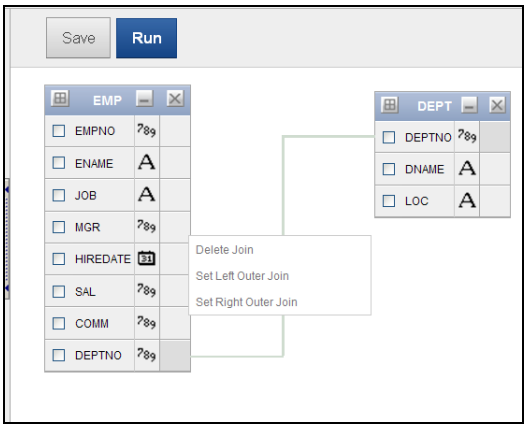
Jelölő négyzet: Erre való kattintással a hozzátartozó oszlop bekerül a lekérdezés szelekciós listájába.

Oszlopnév: Eredeti oszlopnév.

Típus mező: Az oszlop adattipusának grafikus jele.

Összekapcsoló mező: Erre kattintva kezdeményezhetünk egy összekapcsolást egy másik tábla valamely oszlopával. Az összekapcsolás akkor jön létre, ha egymás után két tábla valamely sorainak Összekapcsoló mezőire kattintunk. E kapcsolat grafikai jeleként rajzolódik ki a kijelölt Objektum Sorok között az úgynevezett Kapcsolati Vonal, melyre kattintva,

választhatunk a különböző tipsú külső összekapcsolások közül (1.44. ábra).



1.44.ábra. *Kapcsolati Vonalra kattintva megjelennek az összekapcsolási lehetőségek*

Megjegyzés

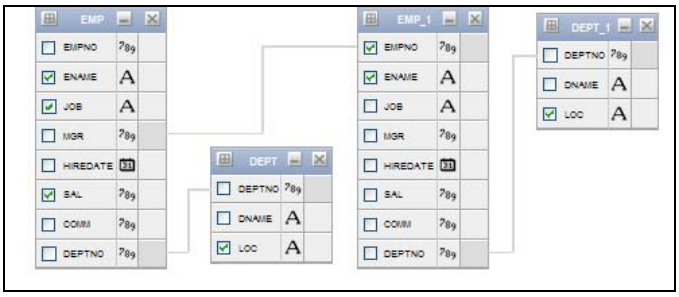
Felhívjuk a figyelmet, hogy az APEX grafikus felületén mindig rá lehet kattintani a halványan megjelenő grafikus elemekre. Az ekkor felugró ablakokból általában segítséget kaphatunk a feladataink megoldásához.

Példa

Listázzuk az 1000 USD-nél magasabb fizetésű dolgozók azonosítóját, nevét, munkakörét, telephelyét, valamint a dolgozó főnökének azonosítóját, nevét és telephelyét, a főnök telephelye szerint rendezve.

Megoldás

A tervezést az 1.45. ábrán látjuk.



1.45. ábra. A tervező sablon

Nézzük meg a lekérdezés feltételeinek beállítására szolgáló Feltétel Lapot a Szerkesztő Ablakban (1.46. ábra).

	Column	Alias	Object	Condition	Sort Type	Sort Order	Show	Function	Group By	Delete
▲ ▼	ENAME	Dolgozó	EMP		Asc		☑		☐	×
▲ ▼	JOB	Munkakör	EMP		Asc		☑		☐	×
▲ ▼	SAL	Fizetés	EMP	>1000	Asc		☑		☐	×
▲ ▼	LOC	DolgozóTelephelye	DEPT		Asc		☑		☐	×
▲ ▼	EMPNO	FőnökAzon	EMP_1		Asc		☑		☐	×
▲ ▼	ENAME	Főnök	EMP_1		Asc		☑		☐	×
▲ ▼	LOC	FőnökTelephelye	DEPT_1		Asc		☑		☐	×

1.46. ábra. *A feltétel lap a Szerkesztő Ablakban*

A Feltétel Lap jeleinek és menüpontjainak jelentését az alábbiakban mutatjuk meg:

Jel, menüpont	*Leírás*
Fel-, lemutató nyíl	Vezérli az oszlopok egymás utáni helyzetét a szelekciós listában
Column	Az oszlop nevét jeleníti meg
Alias	Másodlagos oszlopnév
Object	Az objektum felhasználói neve

Condition	Oszlopfeltétel megadása (a `WHERE` feltétel)
	Például:
	`>=120`
	`='RT'`
	`IN (SELECT deptno FROM dept)`
	`BETWEEN sysdate AND sysdate-20`
	`LIKE '%KM%'`

Sort Type	A rendezés típusa: ASC, DESC
Sort Order	Sorbarendezés a megfelelő oszlopok szerint. Itt számokat kell beírni, tehát `1, 2, 3,`
Show	A szelekciós listában való megjelenés előírása/letiltása

Function	A `NUMBER` típusú függvények:
	`COUNT, COUNT DISTINCT, AVG, MAX, MIN, SUM`
	A `VARCHAR2` és `CHAR` típusú függvények:
	`COUNT, COUNT DISTINCT, INITCAP, LENGTH, LOWER, LTRIM, RTRIM, TRIM, UPPER`
	A `DATE` és `TIMESTAMP` típusú függvények:
	`COUNT, COUNT DISTINCT`

Group By	A csoportosító oszlopok kijelölése
Delete	A kijelölt oszlop törlése a feltételrészből

A Grafikus Tervező Ablak feletti `RUN` gombra kattintva a Szerkesztő Ablakban megjelenik az eredménylista (1.47. ábra).

Dolgozó	Munkaköre	Fizetés	Jutalék	DolgozóTelephely	Főnök	Főnök Telephely
MILLER	CLERK	1300	-	NEW YORK	CLARK	NEW YORK
CLARK	MANAGER	2450	-	NEW YORK	KING	NEW YORK
JONES	MANAGER	2975	-	DALLAS	KING	NEW YORK
BLAKE	MANAGER	2850	-	CHICAGO	KING	NEW YORK
ADAMS	CLERK	1100	-	DALLAS	SCOTT	DALLAS
SCOTT	ANALYST	3000	-	DALLAS	JONES	DALLAS
FORD	ANALYST	3000	-	DALLAS	JONES	DALLAS
TURNER	SALESMAN	1500	0	CHICAGO	BLAKE	CHICAGO
MARTIN	SALESMAN	1250	1400	CHICAGO	BLAKE	CHICAGO
WARD	SALESMAN	1250	500	CHICAGO	BLAKE	CHICAGO
ALLEN	SALESMAN	1600	300	CHICAGO	BLAKE	CHICAGO
					row(s) 1 - 11 of 11	

1.47. ábra. *A futási eredménye*

Az előállított lekérdezést azonnal megnézhetjük, ha az SQL menüpontra kattintunk. Jellegzetes az objektumhivatkozása. (Az 1.48.ábrán látható a tervezés eredménye.)

```
select EMP.ENAME as "Dolgozó",
    EMP.JOB as "Munkakör",
    EMP.SAL as "Fizetés",
    DEPT.LOC as "DolgozóTelephelye",
    EMP_1.EMPNO as "FőnökAzon",
    EMP_1.ENAME as "Főnök",
    DEPT_1.LOC as "FőnökTelephelye"
 from DEPT DEPT_1,
    EMP EMP_1,
    DEPT DEPT,
    EMP EMP
 where EMP.DEPTNO=DEPT.DEPTNO
    and EMP.MGR=EMP_1.EMPNO
    and EMP_1.DEPTNO=DEPT_1.DEPTNO
    and EMP.SAL >1000
 order by DEPT_1.LOC ASC
```

1.48. ábra. *A grafikusan tervezett lekérdezés generált alakja (SELECT utasítás)*

Megjegyzés

1. Az `SQL` menüpontban az utasítás tartalma nem módosítható.
2. A grafikus összeállítással létrehozható lekérdezések köre korlátozott, láthatjuk, hogy a dolgozók között nincs King, de a bal vagy jobb oldali külső összekapcsolást grafikusan már nem tudjuk megvalósítani.

1. Feladat:
Listázza az egyes munkakörök nevét, és a hozzájuk tartozó átlagfizetést a munkakörök szerint rendezve.

2. Feladat:
Listázza a főnökazonosítókat, valamint az egyes főnökök beosztottjainak összfizetését a főnökazonosító szerint csökkenően rendezve.

3. Feladat
Listázza az egyes telephelyeken dolgozók számát.

Az Objektum varázsló komponens

Az Objektum Varázsló (`Object Browser`) komponens az adatbázis objektumok kezelésére szolgál. Segítségével tetszőleges SQL, DDL és DML utasításnak megfelelő tevékenységek, valamint egyszerűbb lekérdezések egyaránt végrehajthatók.

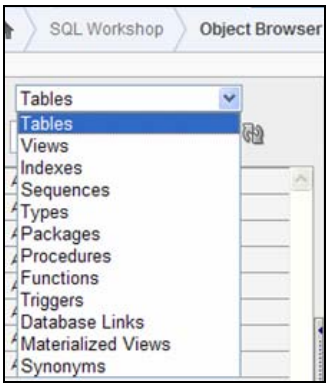
Példa

Vizsgáljuk meg a `dept` táblát az Objektum Varázsló komponens segítségével.

Megoldás

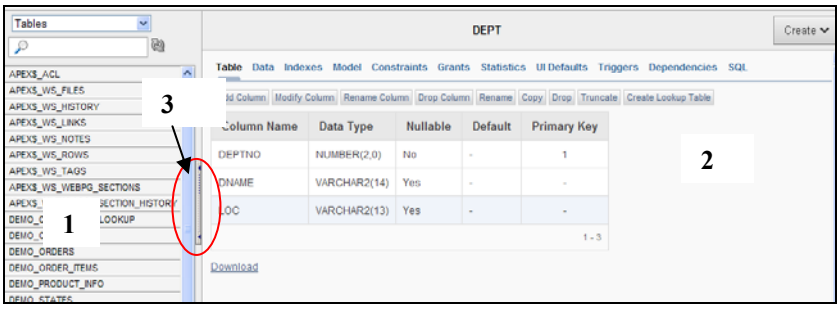
Menjünk a Visszalépő Sávon az SQL műhelybe, és kattintsunk az

ikonra, vagy pedig a jobboldali kis ikon segítségével indítsuk el az Objektum Varázslót. Ez a varázsló az 1.49. ábrán látható mindazon objektumokat kezeli, amelyek a legfölül a `Tables` elemet tartalmazó legördülő mező megnyitásával láthatók. (Megjegyezzük, hogy az ablak jobb oldalán található `Create` gomb segítségével létre is tudjuk hozni a kiválasztott tipusú objektumot.)



1.49. ábra *Lehetséges objektumok.*

Ezekután jelöljük ki a `dept` táblát (1.50. ábra). A megjelenő adatok a `DESCRIBE` utasítás (menüpont) eredményének felelnek meg.



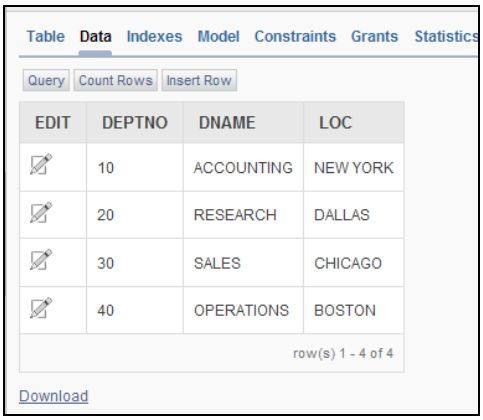
1.50. ábra. *Az* `Object Browser` *komponens ablaka*

Az ablak szerkezete hasonló a korábban már bemutatott lekérdezés varázsló (`Query Builder`) komponens ablakának szerkezetéhez:

1. Objektum Választó Ablak
   A legördülő menüben kiválasztott adatbázis objektum típus objektumait jeleníti meg
2. Objektum Megjelenítő Ablak
   Objektum választó felületen kiválasztott objektumról kaphatunk bővebb információkat, valamint sok objektumkezelési lehetőség áll rendelkezésünkre.
3. Felosztó Léc
   A Felosztó lécre kattintva a bal oldali Objektum Választó Ablak területe eltűnik, és nagyobb lesz az Objektum Megjelenítő Ablak.

A megjelenő ablak baloldalán a lenyíló mezőben kiválaszthatjuk azt az objektumot, amelyet szeretnénk megjeleníteni. A jobboldalon egymás alatt két menüsor látható, ahol a felső a főmenü sora, melynek segítségével az objektumok kezelését végezhetjük el. Az alsó menüsor a kijelölt főmenübeli menüponthoz tartozó almenü sora, itt lehet az objektum rekordjainak kezelését elvégezni.
Könyvünk nem terjed ki e két menüsor teljes ismertetésére. Néhányat azonban röviden ismertetünk.

- `Table` menüpont: megfelel a `DESCRIBE` utasításnak.A menüsor segítségével a DDL utasításoknak megfelelő műveleteket tudjuk elvégezni, mint azt az 1.31. ábrán is láthatjuk.
- `Data` menüpont: A tábla rekordjait jeleníti meg módosítási lehetőséggel együtt. Jelenítsük meg például a `dept` táblát (1.32. ábra).



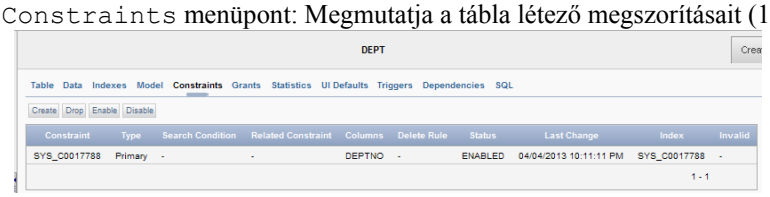
1.51. ábra. *A dept tábla tartalma*

Bármelyik sor első oszlopa az `EDIT` oszlop. Ide kattintva az adott rekord adatait módosíthatjuk (kivéve a piros csillaggal jelölt `Primary Key` oszlopot) az `Apply Changes` gombbal. A táblázat alatt található `Download` linkre kattintva táblázatunk adatait `csv` kiterjesztésű (Excel típusú) fájlba exportálhatjuk.

Az almenü sorában nagyon egyszerű lekérdezést hozhatunk létre (`Query`), megnézhetjük a tábla sorainak számát (a `Count Rows` menüponttal), valamint sorokat szűrhatunk be (`Insert Row`, 1.52. ábra).

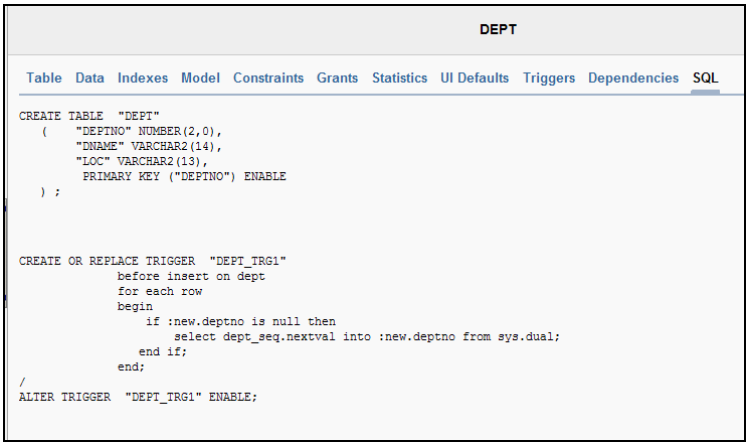
1.52. ábra. *Sorbeszúrás*

- `Constraints` menüpont: Megmutatja a tábla létező megszorításait (1.53. ábra).



1.53. ábra. *A megszorítások megtekintése*

- `SQL` menüpont: Megmutatja a tábla-létrehozási utasításának szerkezetét (1.54.ábra).



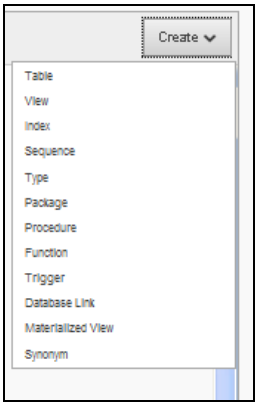
1.54. ábra. *A táblát létrehozó előállított SQL utasítás*

OBJEKTUM-LÉTREHOZÁS

Mivel az Objektum Varázsló (`Objekt Browser`) komponens másik fő funkciója az objektumok kezelése mellett az objektumok létrehozása, ezért az alábbiakban ezt mutatjuk be.

Tábla létrehozás

A `Create` gombra kattintva választhatjuk ki a létrehozandó objektumot (1.55. ábra).



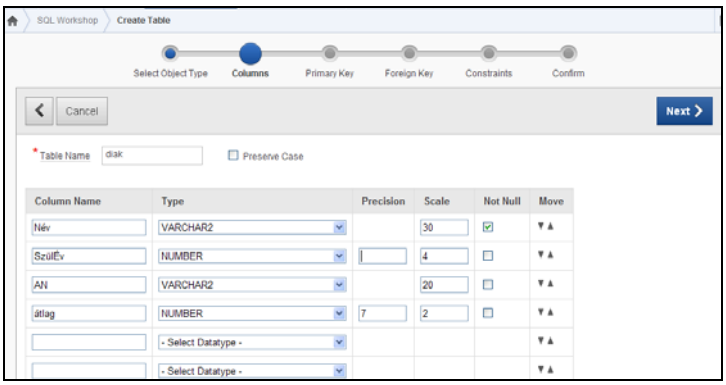
1.55. ábra. *A létrehozandó objektumok*

Példa

Hozzunk létre egy diak táblát, mely tartalmazza a diák nevét, a születési évét, az anyja nevét, és a tanulmányi átlagát.

Megoldás

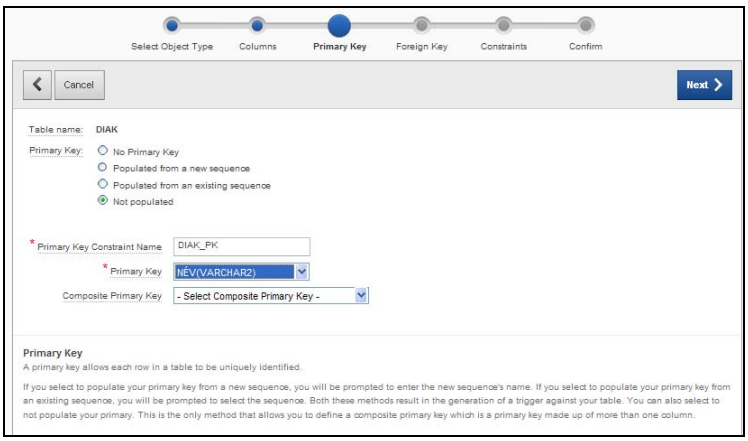
Kattintsunk a létrehozandó objektumra, tehát a `Create` gombra. Ekkor megjelenik az ablak tetején az a lépéssor, amin végig kell menni a tábla létrehozása érdekében. Alatta a megjelenő ablakba írjuk be a tábla nevét (`diak`), a tábla oszlopainak nevét (`Név, SzülÉv`, `AN`, `Átlag`), az adattípusokat pedig válasszuk ki, valamint adjuk meg a méreteket is (1.56. ábra). Ügyeljünk a számadatok megadására (pontosság – `Scale`, karakterek száma – `Precision`)



1.56. ábra. *A tábla létrehozásához szükséges adatok megadása*

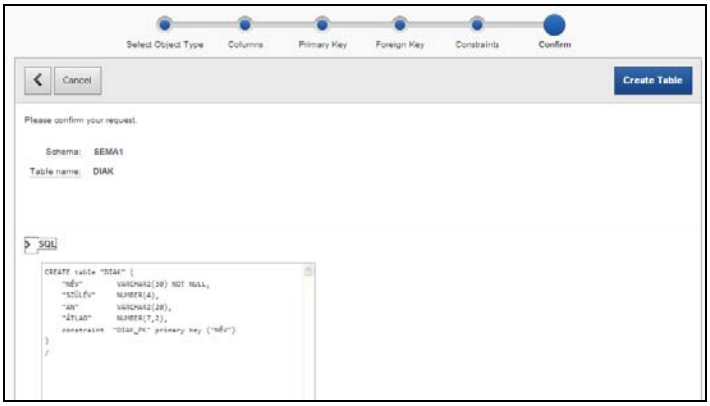
Megjegyzés

Kattintunk ezután a `Next` gombra, majd először adjuk meg a `Primary Key` megszorítást, ezután ismét `Next` (1.57. ábra).



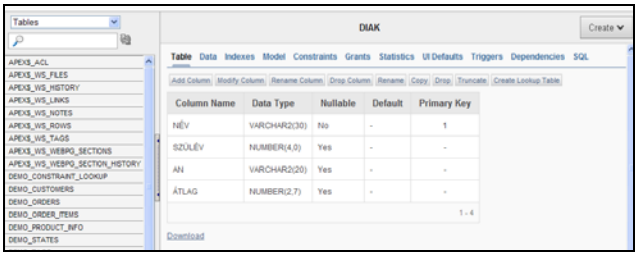
1.57. ábra. *Megszorítás hozzáadása*

Mivel több megszorítást egyenlőre nem adunk hozzá, ezért kattintsunk a `Next` gombra, ismét `Next`, és megint `Next`. Ellenőrizzük, majd kattintsunk az SQL feliratra, majd a `Create Table` gombra (1.58.ábra).



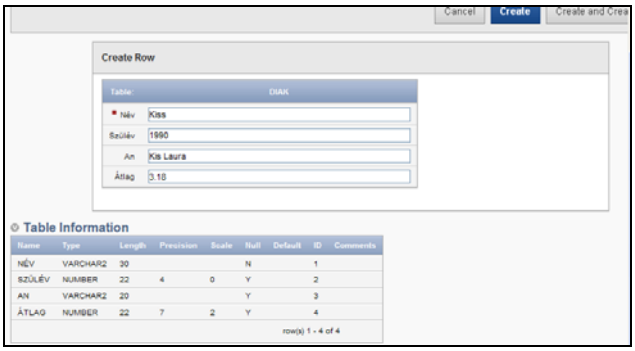
1.58. ábra. *Ellenőrzés*

A tábla létrejött, amint azt a következő (1.59. ábra) ábrán láthatjuk is.



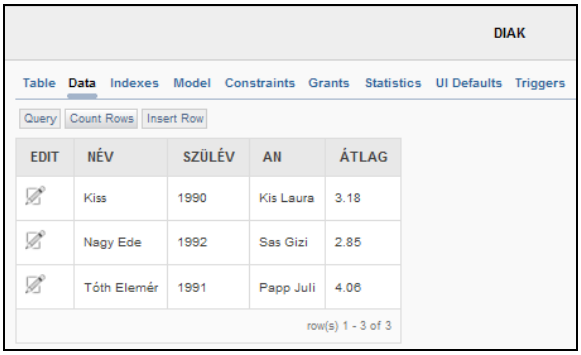
1.59. ábra.A tábla létrejött egy elsődleges kulccsal

Röviden bemutattuk az APEX hagyományos (nem webes alkalmazás-fejlesztő) adatbázis-kezelő felületét, illetve az adatbáziskezelő komponenseket.
Töltsük fel a táblát adatokkal.
Kattintsunk a `Data` Menüpontra, majd az `Insert Row` gombra. Írjuk be az adatokat, és a Tábla Információval ellenőrizhetjük az adatok helyességét (1.60. ábra).



1.60.ábra *Új rekord felvitele és a tábla információ*

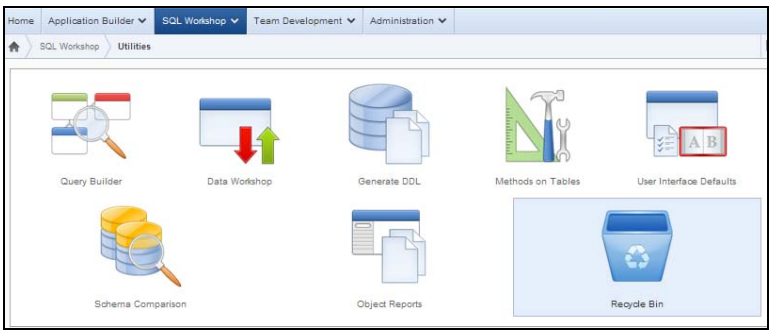
Ezután: `Create`, vagy `Create and Create Another` gomb.



1.61. ábra *A létrehozott és feltöltött adattábla.*

Segédprogramok (`Utilities`) komponens

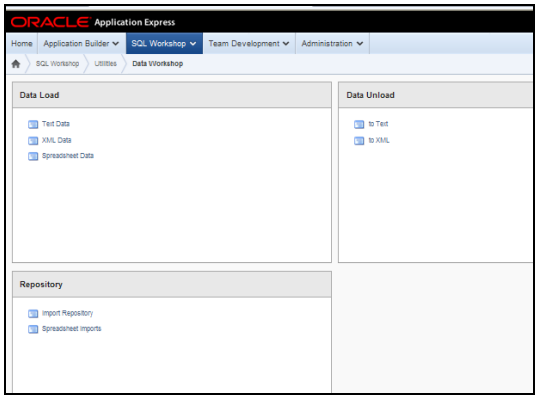
A Segédprogramok (`Utilities`) komponens segítségével egyszerű lekérdezéseket állíthatunk össze, adatokat tölthetünk be és vihetünk ki az adatbázisba/ból, készíthetünk könyvtárat (dll), láthatjuk az objektumok listáit, helyreállíthatjuk az eldobott objektumokat, stb. Kattintsunk az ikonra:



1.62. ábra *Lehetséges segédprogramok*

- A Lekérdezés varázsló (`Query Builder`) használatáról fent volt már szó.

- A második ikon az Adatkezelés (`Data Workshop`):
  Az adatokat ki lehet vinni az adatbázisból, és be lehet tölteni az adatbázisba, azaz:
  – adat betöltése (`Data Load`),
  – adat kivitele (`Data Unload`),
  – raktár (`Repository`).
  Amint azt az 1.63. ábrán láthatjuk, az adattáblából lementett (`Data Unload`) adatok lehetnek txt, és xml tipusúak, míg a betölthetők (`Data Load`) txt, xml és spredsheet (csv) típusúak.

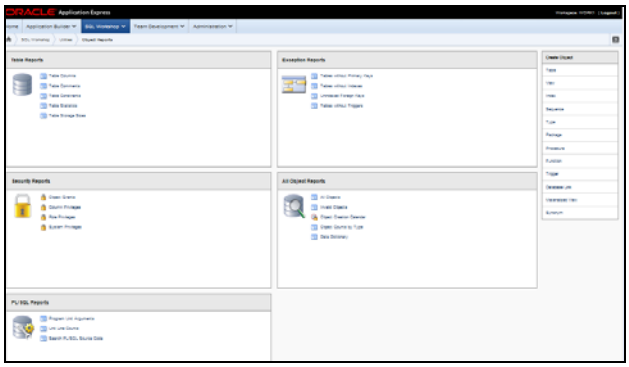


1.63.ábra *Adatkezelés*

- Táblák csomagjait készíthetjük el a `Methods on tables` menüpont segítségével.
- `User Interface Default`: Megváltoztathatjuk az alapértelmezett felhasználói interfész Tartományait (`Region`) és Elemeit (`Item`). A varázsló segítségével specifikálhatjuk, hogy melyik létező felhasználói felületet szeretnénk használni. Varázslóval létrehozott Ürlap (Form) vagy Tartomány az alapértelmezett információkat használja tulajdonságaihoz.
  Két kategóriája van:

Tábla Szótár (`Table Dictionary`): az adatbázisban definiált táblák és oszlopok alapértelmezését határozhatjuk meg
Tulajdonság Szótár (`Attribute Dictionary`): oszlopnevek és tulajdonságainak alapértelmezését adhatjuk meg.

- `Schema Comparison:` Két séma adatbázis objektumait képes összehasonlítani. Mindkettőnek ugyanazon a munkaterületen kell lenni.
- `Object Reports`:



1.64. ábra *Objektumok listázása*

- Listázza az adatbázis tábláin keresztül ennek részleteit az első ablakban (1.64. ábra).
- Listázza a titkosság részleteit, objektumok engedélyezését, oszlopokszerpkörök és a rendszerjogokat.
- `PL/SQL Reports`: megjelenítheti a unitokat, a unit sorainak számát, megkereshetjük a forráskódokat.
- `Exception Reports`: Megjeleníti a nem indexelt Idegen kulcsokat, azokat a táblákat, amelyeknek nincs Elsődleges Kulcsuk, indexük és triggerük.

- Lomtár (`Recycle Bin`): megmutatja és visszaállítja az eldobott objektumokat.
Két kategoriája van:
- (`Dropped Object`) Menedzseli a lomtárban levő objektumokat
- (`Purge Recycle Bin`) :Kiüríthető megjelenítés nélkül a lomtár

- `RESTful Services`:
A REST = Representational State Transfer (REST). A World Web, osztott hipermédia rendszer szoftver architektura stílusa. A szolgáltatás leírja ezt, mint RESTFul, ha az alapkészlete megegyezik a REST -el.
A `REST` a következő karakterisztikát tartalmazza:
- Az adatokat az erőforrás modellizálja. Az erőforrást az URLs határozza meg.
- Egy kis készletet használ a az erőforrás kezeléshez
(például: `PUT`, `POST`, `GET`, `DELETE`)
Maga a szolgáltatás a publikus Web szolgáltatás alkalmazásait kezeli, azokét, amelyek elérhetők más alkalmazások összeköthető adataival. Például beállíthatunk egy RESTful szolgáltatást annak érdekében, hogy ismételten előállítsuk minden dolgozó nevét egy egyéni részleg számára.

Az Application Builder komponens

Az APEXnek az Alkalmazás Varázsló (`Application Builder`) komponense segítségével lehet tetszőleges webes adatbázis-kezelő alkalmazást létrehozni. Ennek használatát, és az ezzel kapcsolatos lehetőségeket a továbbiakban különböző mintapéldákon keresztül fogjuk bemutatni.

E komponens segítségével nyílik lehetőségünk arra, hogy akár gyorsan, néhány kattintással, egyszerűen, a varázslók segítségével hozzunk létre összetett webes adatbázis-alkalmazásokat. Ezeket mi magunk is összeállíthatjuk, azonban lehetőségünk van arra is, hogy egy egyszerű táblázat (CSV, XLS) alapján generáljunk egy egyszerű alkalmazást. E komponens egyúttal lehetőséget ad az alkalmazásaink környezeti paramétereinek menedzselésére, az alkalmazásainkhoz fűződő statisztikák megtekintésére, valamint az alkalmazásaink, és az azokhoz tartozó objektumok egyszerű exportálására és importálására is. Felhasználhatunk formázó webes kódokat, valamint java programokat is az alkalmazásainkban.

2. fejezet. Alkalmazás-létrehozás alapjai

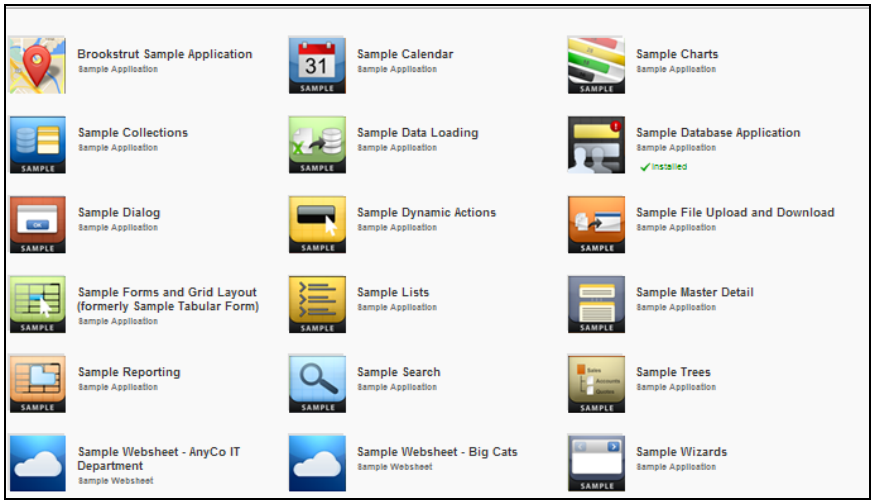
Ez a fejezet az Alkalmazás Varázslóval készülő néhány mintapéldán keresztüli megvalósításával foglalkozik. Nem törekedtünk teljességre. Az alkalmazások készítéséhez szükséges legfontosabb fogalmakat, eszközöket lehetőségeket azonban ismertetjük. Úgy véljük, ezek után egy informatikus el tud jutni a legkülönfélébb lehetőségek megvalósításához is. Továbbiakban az Adatbázis alkalmazások készítésével foglalkozunk.

Megjegyezzük, hogy készíthetők még Websheet Application és Alkalmazási csomagok (`Packaged Applications`)

A `Websheet` alkalmazások interaktiv weblapok kombinációja szöveges adatokkal. Ezek az alkalmazások kiemlten dinamikusak, és a felhasználók által meghatározottak. A Websheet alkalmazsás folyamatos ellenőrzést, felügyeletet is tartalmaz, keresőképessége is van, valamint különböző feljegyzéseket készít. A Websheet alkalmazásokat lehet titkosítani, ugy mint belépési jelszó listája, és tartalmaz mindenféle beépített hitelesítési modelleket.

Alkalmazási csomagok (`Packaged Applications`)(2.1.ábra):

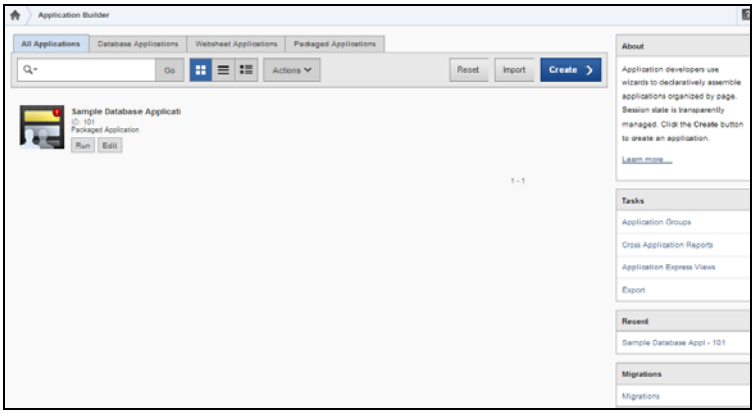
mint az ábrán is láthatjuk az APEX számtalan csomagot tartalmaz. Ezek közül a pipával megjelölt Sample Database Application automatikusan installálódik egy fejlesztő sémájában.A többit nekünk kell varázslóval installálni.



2.1. ábra *Alkalmazási csomagok*

Mintaalkalmazás megtekintése

Kattintsunk az Alkalmazás Varázsló (`Application Builder`) ikonra. Ezzel elindítottuk az Alkalmazás Varázslót. Ebben a verzióban már azonnal megjelenik és elindítható az Oracle mintapéldája.

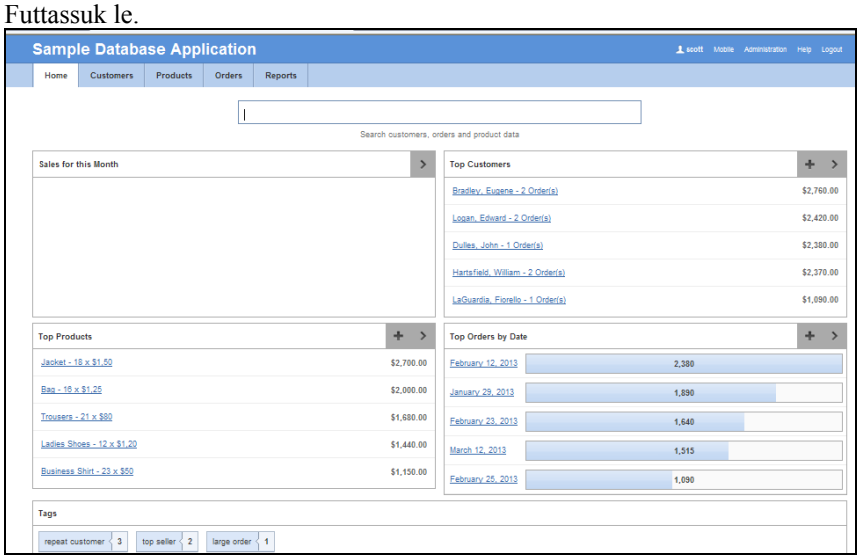


2.2. ábra. *Mintaalkalmazás első lépése*

Az alkalmazás száma 101 (2.2.ábra), a `Run` gombbal futtathatjuk és az `Edit` gomb segítségével szerkeszthetjük az alkalmazást. Táblái mint már láthattuk az Objektum Varázslóban is, a `magyar` sémájában megtalálhatók. Belépési jelszó itt is a fejlesztő felhasználó jelszava `Tigris-1`.

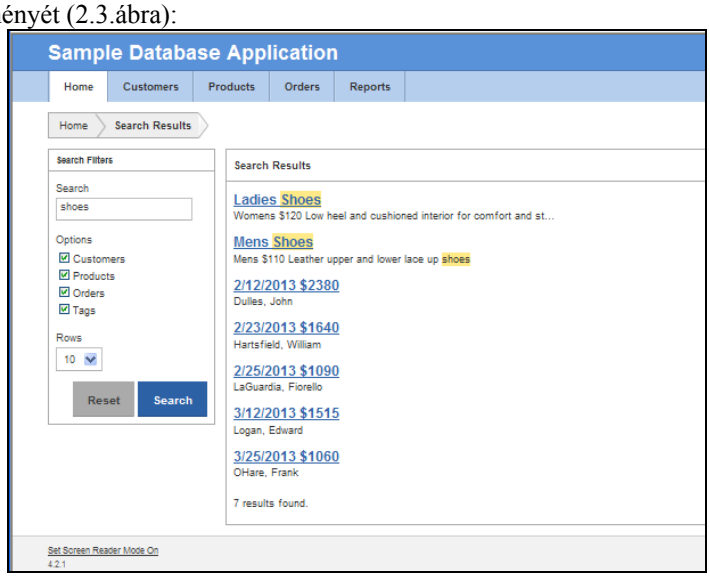
A sikeres belépést követően böngésszünk az alkalmazásban, derítsük fel a lehetőségeit, fedezzük fel a navigációs elemeket, a működését, a megoldásait.

Futtassuk le.



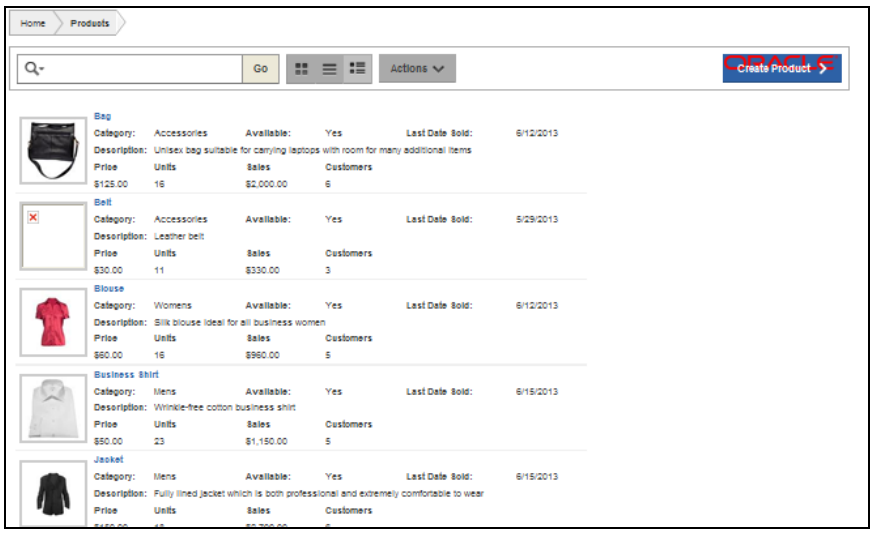
2.3. ábra. *Mintaalkalmazás*

Irjuk be a felső kereső sorba, pl. azt, hogy cipő (shoes) és figyeljük meg a keresés eredményét (2.3.ábra):



2.3. ábra. *Mintaalkalmazás futása*

Itt mehetünk tovább, kereshetünk mást is .kattintsunk a Products fülre, és máris megjelenik a termékek listája(2.4.ábra).

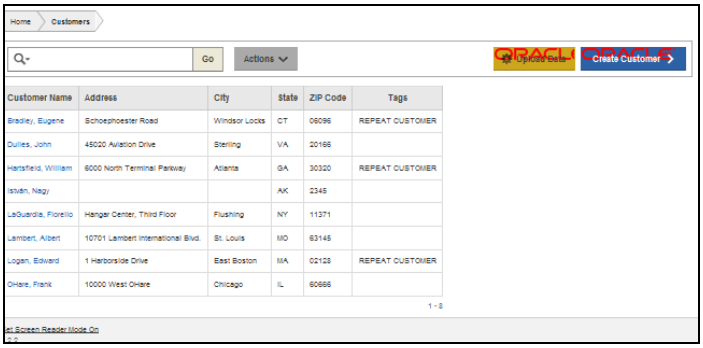


2.4. ábra. *Mintaalkalmazás működése*

Fent baloldalon található az alkalmazás neve, jobb oldalon pedig a felhasználó `magyar` , a `Mobile` és `Administration` menüpont , a `Help` és a `Logout` menüpont.

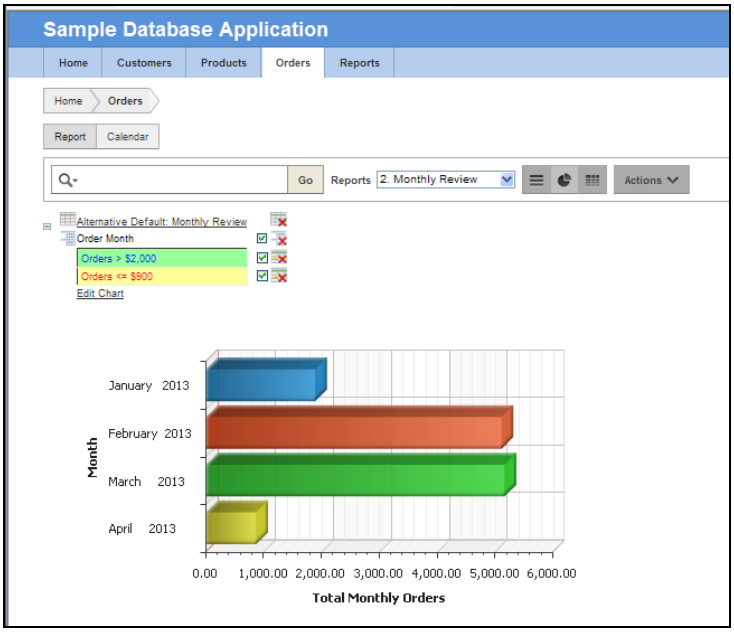
A demo alkalmazásban termékek, vevők és megrendeléseik között böngészhetünk, módosíthatjuk azokat és újakat hozhatunk létre, valamint lehetőségünk van szöveges és grafikus statisztikák megtekintésére.

Vegyünk fel egy új vevőt. Kattintsunk a Customer lapra, majd a jobb oldali Create Customer gombra. Töltsük ki a vevő adatait az ürlapon, de minimálisan azokat a beviteli mezőket, amelyek mellett piros csillag található, majd kattintsunk az Add Customer gombra és végrehajtja az új rekord felvételét.



2.5 *Mintaalkalmazás adatfelvétele*

Az alkalmazásban történő navigációhoz használjuk az úgynevezett Füleket (`Home`, `Customers`, `Products`, `Orders`, `Reports`), valamint a VisszalépőSáv-ot. A megjelenő listák rekordjai előtt található ikonra kattintva, olyan lapra juthatunk, ahol az adott rekordról találunk bővebb információt, vagy az adott rekord adatait módosíthatjuk.



2.6. ábra. *A minta Alkalmazás alkalmazás egy lapja*

Az alkalmazásból való kijelentkezéshez használjuk a jobb felső sarokban található `LOGOUT` linket. Próbálkozzunk bátran, nagyon sokat lehet belőle tanulni.

Tanácsoljuk, hogy használja segítségül a mintapéldát. Lényegében az APEX minden lehetőségére kitér.

Megjegyzés:

Oracle Application Express esetén a grafikus diagrammok megjelenítéséhez szükség van az ADOBE SVG Viewer használatára, melyet az alábbi linken tölthetünk le.

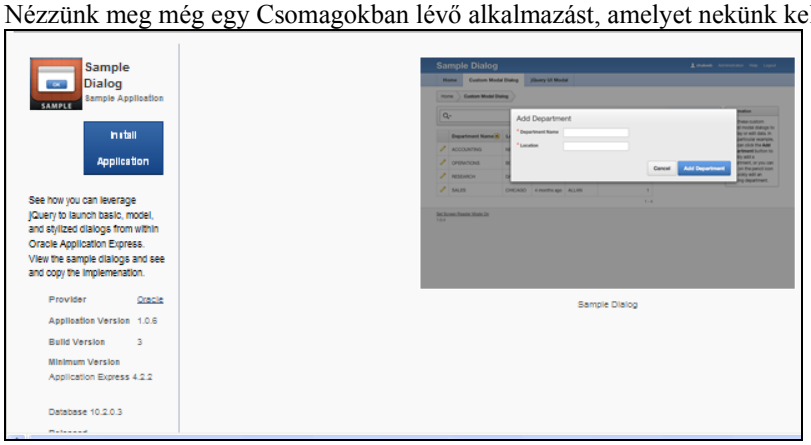
Magyarul:
http://download.adobe.com/pub/adobe/magic/svgviewer/win/3.x/3.03/hu/SVGView.exe

Angolul:
http://download.adobe.com/pub/adobe/magic/svgviewer/win/3.x/3.03/en/SVGView.exe

Bővebb információ az ADOBE SVG Viewer-ről:
http://www.adobe.com/svg

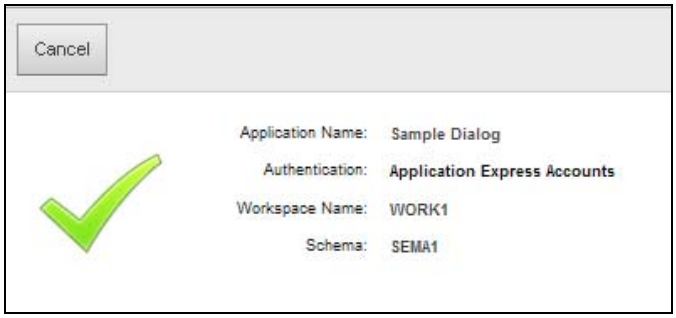
Kattintsunk egy csomagra. Legyen ez például a Sample Dialog. Ez egy modal dialog ablakon keresztül vesz fel új rekordot.

Nézzünk meg még egy Csomagokban lévő alkalmazást, amelyet nekünk kell installálni.



2.7. ábra *Egy csomag installálása*

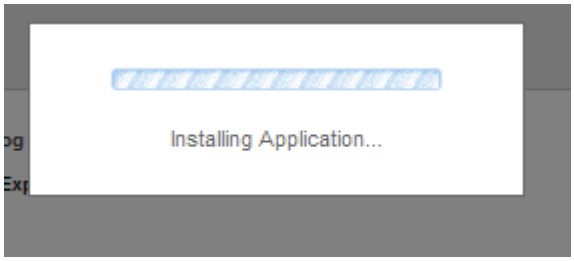
Kattintsunk az `Install Application` gombra (2.7. ábra). A következő ablakban felajánlja, hogy ennek hitelesítése az aplication express accounttal rendelkezzen. Kattintsunk a `Next` gombra.



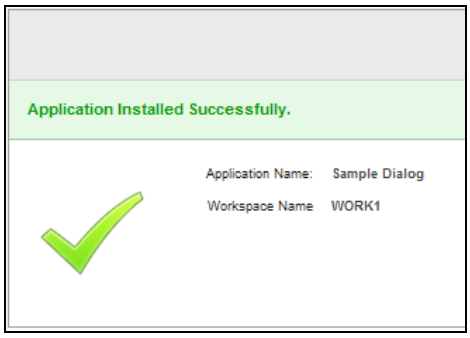
2.8. ábra *Összesítés*

Összesítés (2.8. ábra), majd kattintsunk az `Install Application` gombra.

Türelem az installálás folyamatban van (2.9. ábra)



2.10.ábra *Installáció folymata*



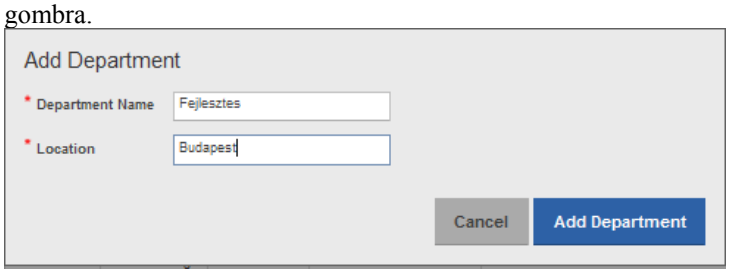
2.11.ábra Sikeres

Sikerült az installálás (2.11. ábra) és kattintsunk a `Run` gombra. Jelentkezzünk be. `magyar/Tigris-1`

2.12. ábra *A működő csomag*

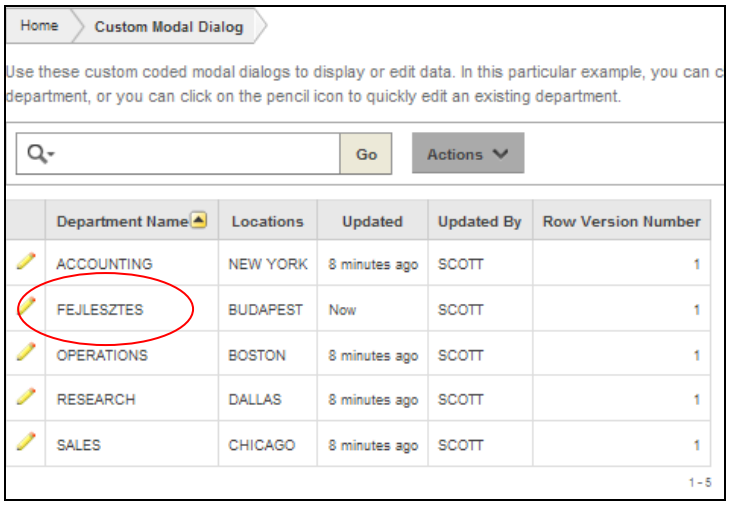
Kattintsunk az `Add Department` gombra.

megjelenik egy interaktív ablak, ezt töltsük ki, majd kattintsunk az `Add Department` gombra.



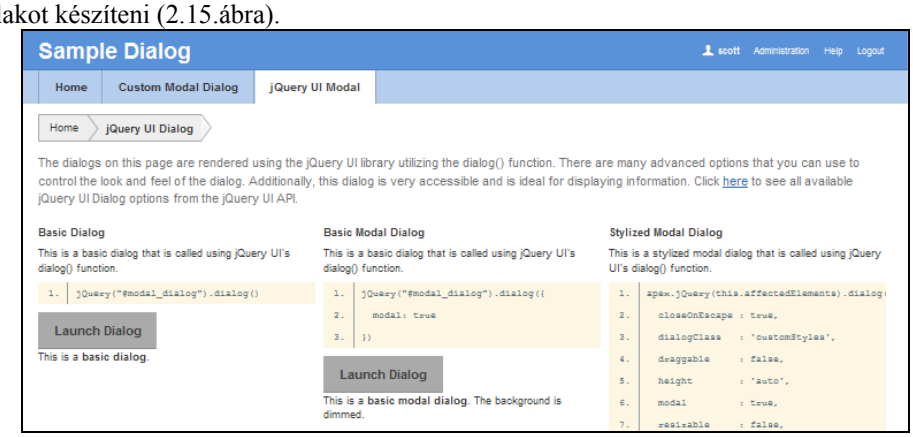
2.13. ábra *Új részleg hozzáadása*

Hozzáadja a beírt új rekordot (2.14. ábra).



2.14.ábra *Sikeres adatbevitel*

A második fülön a mintapélda pedig leírja, hogyan kell jQuery vel különböző dialóg ablakot készíteni (2.15.ábra).



2.15.ábra Harmadik fül tartalma

Most megismerkedünk az alapfogalmakkal, azután ismét ajánljuk, hogy nézegessék a mintapéldákat. A lapokat, folyamatokat, feltételeket,

Tábla megjelenítés
Az első saját alkalmazás készítése

Az Oracle Application Express, az Oracle számára egy gyors, webes alkalmazásokat fejlesztő rendszer. Csak egy web kereső és egy tapasztalt programozó szükséges hozzá, és akkor olyan professionális alkalmazás fejleszthető, amely gyors és biztonságos.

Egy webes adatbázis-alkalmazás fejlesztése három alapvető tevékenységből áll;
- a felhasználói felület (User Interface – UI) létrehozásából,
- valamint az alkalmazási logika és az alkalmazás-komponensek megtervezéséből.

Az Alkalmazás Varázsló (`Application Builder`) mindhárom tevékenységre alkalmas.

Az APEX alkalmazások tulajdonképpen weblapok gyűjteménye. A lapokon különböző navigációs vezérlők találhatók, mint például az ablak-kiválasztó (navigációs) fülek, a gombok és a hiperhivatkozások.

Az alkalmazás alapegysége a lap (`page`). A lap különböző helyzetű tartományokra (`region`) osztható fel.

A tartomány a lap egy olyan területe, mely valamilyen összefüggő tartalmat foglal magába. A tartomány tartalmát a forrása határozza meg. Például a tartomány lehet egy `SQL` lekérdezés eredménye, vagyis egy riport, vagy tartalmazhat statikus `html` szöveget.

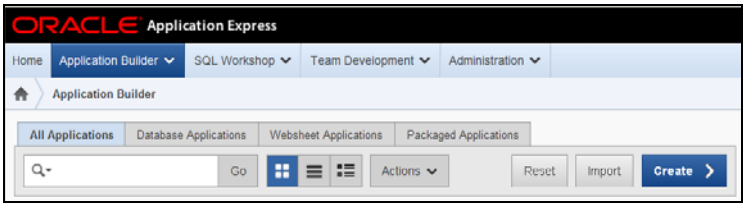
A tartomány grafikus elemei a következők:

- Elem (`Item`), például a szöveg mező (`text`), a (többsoros) szöveg terület (`text area`), a lenyíló szövegmező (`combobox`), stb. Ezt később részletesen tárgyaljuk
- Gomb (`Button`) lenyomásával a felhasználó egy meghatározott lapra, vagy URL címre kerül, mely az információt elküldi és feldolgozza.
- Menü, amely hierarchikus navigációt tesz lehetővé.

A navigációs fülek a tartományokon kívül helyezkednek el, és a lapok közötti váltást teszik lehetővé.

Alkalmazás készítésének kezdete

Lépjünk be az APEX programba a `work1` munkaterületű `magyar/Tigris-1` felhasználó/jelszó párossal. Kattintsunk az `Application Builder` ikonra. Ekkor megjelenik az Alkalmazás Létrehozó (`Create Applcation`) lap (2.16.ábra).



2.16. ábra *Alkalmazás készítésének kezdete*

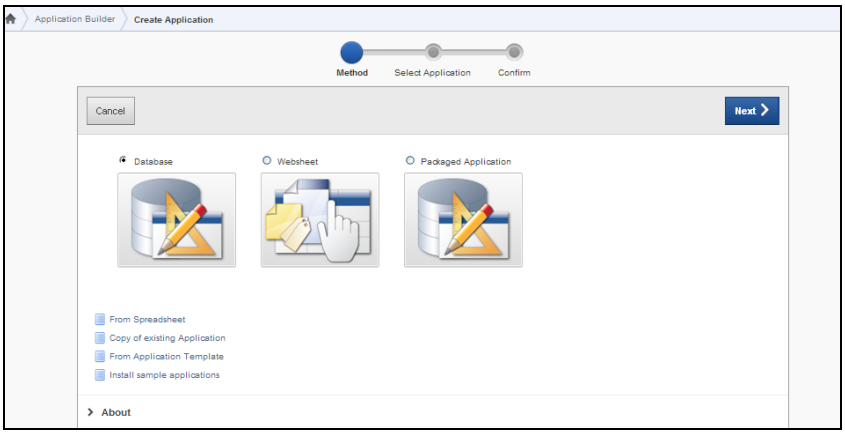
Mi a különbség az adatbázis és a websheet alkalmazás között?

Az alkalmazás egy HTML felület, amelyik az adatbázis objektumokon alapul, úgy mint táblák és eljárások. Mindkét típusú alkalmazást (`Database` és `WebSheet`) az APEX alkalmazás létrehozó varázslóval. A fő különbség a két típusu alkalmazás között az, hogy kinek a számára készül.

Amíg az adatbázis alkalmazás a fejlesztőkhöz kapcsolódik, a Websheet alkalmazást a fejlesztési gyakorlat nélküli végfelhasználók számára tervezik

Mivel egyelőre nincs alkalmazásunk (tehát nem tudjuk azt kiválasztani), kattintsunk a `Create` gombra.

Háromféle módon hozhatunk létre alkalmazást, ahogy az a 2.17. ábrán látható.



2.17. ábra. Az *alkalmazások létrehozási lehetőségei*

- Az első `Database` segítségével olyan adatbázis alkalmazásokat hozhatunk létre, amelyek lapokból állnak, a lapokat mi definiáljuk, biztonságossá tudjuk tenni, specifikálhatjuk a felhasználói felületet. A lapokon táblák, lekérdezések és parancsok eredményei találhatóak. Az alkalmazás létrehozásához szükség van SQL és PL/SQL programozás ismeretére.
- Táblázatos adatokra (`Websheet`) építve hozhatunk létre alkalmazásokat úgy, hogy az tartalmazza az adatok betöltését az adatbázis-kezelőbe, majd egy alapértelmezett riportkészítő és adatszerkesztő alkalmazást állít elő, melyet később igény szerint lehet testreszabni, átalakítani. Az alkalmazás használható lesz az új táblán keresztül történő lekérdezésekre, beszúrásra, módosításra és az analitikus függvények használatára. Használatához nem szükséges SQL programozás ismerete.
- A `Packaged Application` az úgynevezett csomagolt alkalmazások létrehozására szolgál. Egy alkalmazás importálása és installálása bonyolult folyamat. Első lépése a cél adatbázis objektumok és kiinduló adatok létrehozása. Második lépés az installálása és minden képeket, sablonokat tartalmazó, valamint más statikus fájlok installálása.
  A csomagolt alkalmazás használata nagyon egyszerű folyamat. A számos lépéssorozat helyett, meghatározhatjuk a támogatott objektumokat, úgy, hogy az alkalmazást és a szükséges fájlokat áttelepítjük néhány lépés segítségével. Ezt a néhány lépést valósítja meg e kategória.

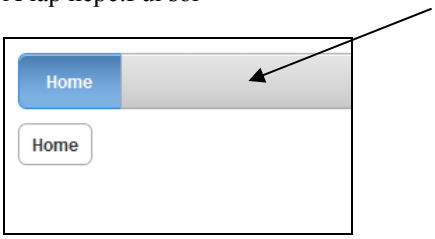
Válasszuk az új alkalmazások létrehozására szolgáló `Database Application` rádiógombot, és kattintsunk a `Next` gombra.

A megjelenő ablakban kötelező megadni a piros csillaggal jelölt mezőket. Az Alkalmazás (`Application`) szövegmezőben a fejlesztőprogram által automatikusan generált szám jelenik meg, melyet megváltoztathatunk. `Name` szövegmezőben adjunk nevet az alkalmazásnak (ez most legyen `Elso`). A séma legyen a felhasználó sémája. A LétrehozásFeltételei (`Create Option`) szövegmezőben választhatunk:

- Indulás a legelejéről (`Start from scratch`)
- Tartalmazza a kezdőlapot (`Include Home Page`)

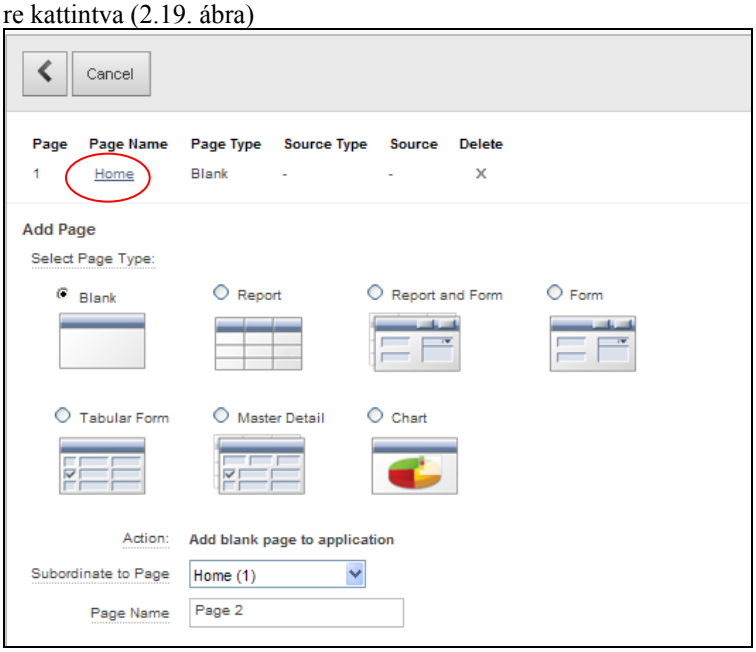
`Include Home Page` választás esetén az első lapunk az úgynevezett `home` lap. Ennek tetején fülek nélkül is elérhető bármelyik itt létrehozott lap, de akármelyik lapról csak a `home` lap érhető el. E sáv felett helyezkedik el a Fülek sora.

A lap képe:Fül sor



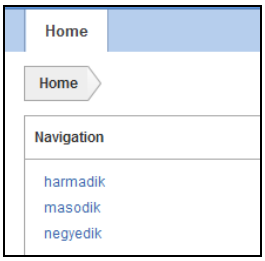
2.18.ábra *Home lap és a fül sor*

A `Home` nevet utólag nem lehet megváltoztatni, csak itt a legelején, a `Home` linkre kattintva (2.19. ábra)



2.19.ábra *Home lap nevének megváltoztatása*

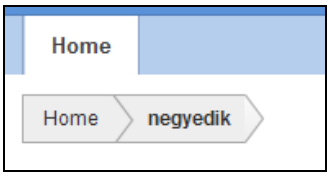
Amennyiben itt, a legelején az alkalmazással együtt hozzuk létre a többi lapot, akkor automatikusan létrehoz egy navigációs sávot, amelyben névsorba rendezve a létrehozott lapok nevei lesznek.



2.20.ábra *Home lap, és a navigáció* ábra

Bármelyik linkre kattintva megjelenik a kijelölt lap, a navigációs sávban `home` és az adott lap lesz elérhető (2.20 ábra).

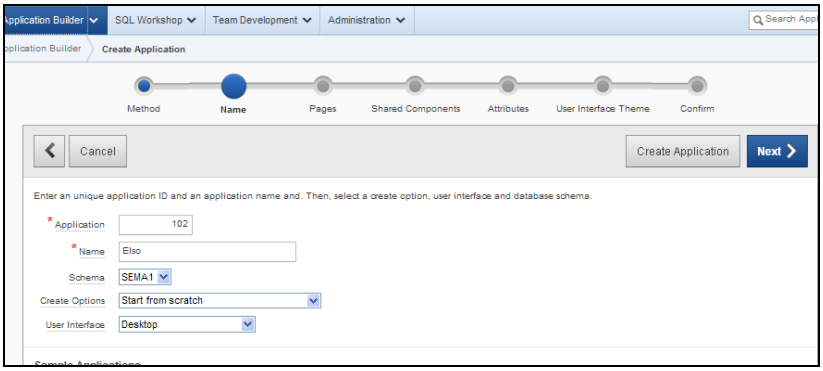
Például kattintva a `negyedik` lapra, megjelenik a `negyedik` lap, a navigációs sávon pedig csak a `home` lap. A többi lap innen nem elérhető (2.21. ábra).



2.21.ábra Home és a kiválasztott lap

Abban az esetben, amikor az új lap hozzáadása nem itt történik (hanem csak az alkalmazás létrehozása után), akkor nem hoz létre navigációs sávot, és a másik lap csak fül vagy gomb segítségével érhető el.

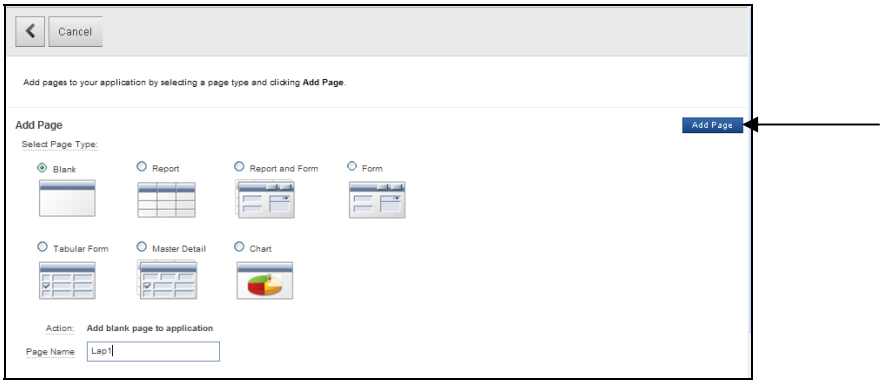
- Használjuk egy korábban már létrehozott modelt (`Use previously created applicatin model`)
- A felhasználói felület (`User Interface`) lehet a Desktop, vagy intelligens mobil lekérdezés.



2.22. ábra. *Az elkészítendő alkalmazás neve és azonosító száma*

Válasszuk `Start from scratch`, és a `Desktop` beállítást, majd Kattintsunk a `Next` gombra.(2.22. ábra)
A megjelenő ablakban kiválaszthatjuk alkalmazásunk elsőként létrehozandó lapjának típusát (2.23.ábra.). A lap típusa lehet:

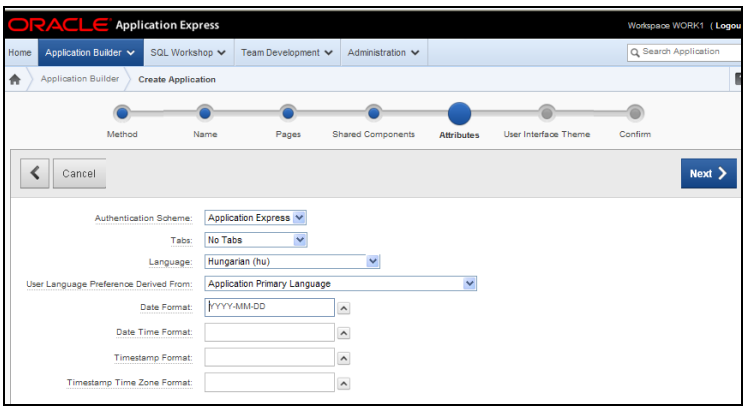
`Blank` :	üres, nincs beépített funkciója,
`Report`:	egy lekérdezés formázott eredménye,
`Form`:	a forrás adattábla egy sora jelenik meg módosítás céljából,
`Tabular Form`:	a forrás adattábla több sorát lehet módosítani, törölni,és beszúrni (a táblának Elsődleges kulcs, azaz Primary Key megszorítással kell rendelkeznie),
`Report and Form`:	ez a lap típus két lap a `Report` és a `Form` kombinációjából épül fel. Az első lapon a felhasználó kiválaszt egy módosítandó sort, a második lapon módosítani tudja a kiválasztott táblát vagy nézetet. A forrás adattáblának és az itt szereplő `Tabular Form`-nak Elsődleges kulcs megszorítással kell rendelkeznie.
`Master Detail`	
`Chart`:	Grafikon laptípus.



2.23. ábra. *A választható lap típusok*

Válasszunk `Blank` típusú lapot, és adjunk nevet a Lapnak. Legyen Lap1, majd és kattintsunk az `Add Page` gombra, utána pedig a `Next` gombra.

A következő megjelenő ablakban kiválaszthatjuk, megosztott komponenst választunk-e egy már létező alkalmazásból, vagy nem. Válasszuk a `NO` rádiogombot. és `Next`.



2.24. ábra. *Hitelesítés, nyelv, és felhasználói nyelv választása*

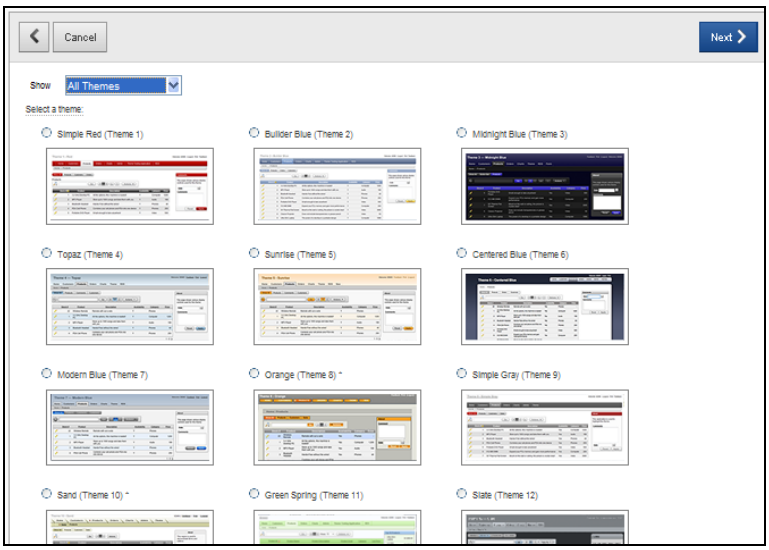
- Megjegyezzük, hogy az `Authentication Scheme` esetén három választásunk van:
  - APEX azonosítás (`Application Express Authentication`), ekkor egy virtuális felhasználó jön létre,
  - nincs belépési jelszó (`No Authentication`), ekkor a belépés nyilvános,
  - az adatbázis jelszavával léphetünk be (`Database Account`), ekkor saját jogon érjük el az adatbázis motort.
- `Tabs` mezőben szintén három lehetőségünk van:
  - `No Tabs` ne legyenek fülek
  - `One Level Tabs` egy szintű fülek legyenek
  - `Two Level Tabs` kétszintű fülsor legyen.

  Válasszuk a `No Tabs` lehetőséget.
- A Nyelv (`Language`) mezőben válasszuk a magyar nyelvet.
- User Language Preference Derived From mezőt ne változtassuk meg,
- Dátum és idő formátumot megválaszthatjuk, de nem kötelező.

Ezután (2.24. ábra) kattintsunk a `Next` gombra.

A megjelenő ablakban választhatunk megjelenítési Felület Mintát (Theme). Ez azt határozza meg, hogy milyen legyen a felhasználói felület általános kinézete, szine, a fülek pozíciója, a kilépés helye, stb. (2.25. ábra).

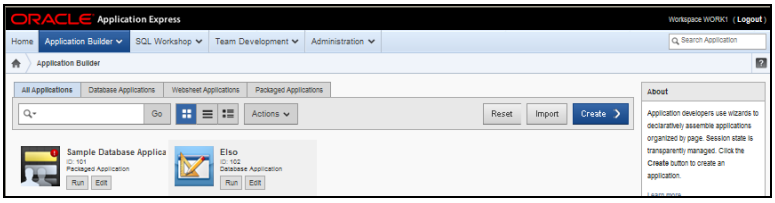
Itt 4 kategória közül választhatunk. Ezek: minden téma, szokásos téma, egyéni téma, származtatott téma (`All Themes`, `Standard`, `Custom` és `Legacy Theme`). Válasszuk ki a szokásos kategoriából a 22-es (`Bluejay`: kékmadár) Felület Mintát, és kattintsunk a `Next` gombra. Megjegyezzük, hogy a Felület Mintát utólag is megváltoztathatjuk, lecserélhetjük.



2.25 ábra. *A lehetséges Felület Minták egy része.*

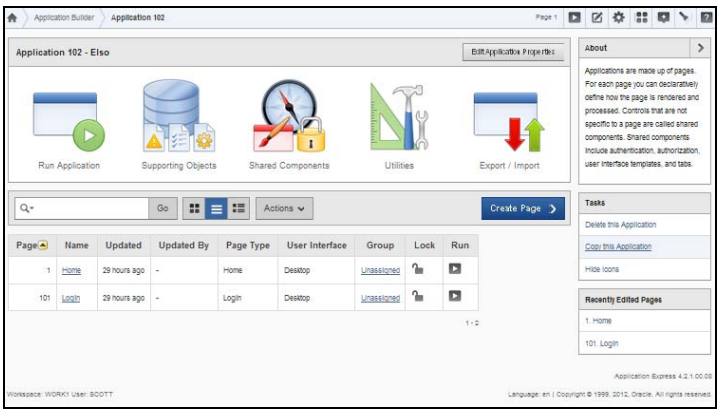
A létrehozandó alkalmazásnak a felhasználó által beállított, illetve a varázsló által automatikusan generált paraméterei (például az alkalmazás azonosítója, mely esetünkben 101) láthatóak a következő ablakban. Ha ez megfelel, akkor kattintsunk a `Create Application` gombra.

Az alkalmazás létrehozása sikerült (2.26. ábra). Az alkalmazásunknak két lapja van, a felhasználói funkcióknak szánt `1-Page 1` lap, és a bejelentkező lap (`101-Login`)



2.26. ábra. *Sikeres alkalmazás-létrehozás*

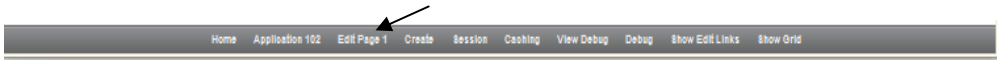
Futtassuk le a már elkészült alkalmazást. Futtatás a Run gombbal, vagy rákattintunk az alkalmazásra , ekkor a következő ablak jelenik meg:



2.27. ábra. *Az alkalmazás két lapja*

Lefuttathatjuk az alkalmazást a `Run Application` ikonnal. Az alkalmazás egyes lapjait pedig a `Run` oszlopban levő nyillal (2.27. ábra)

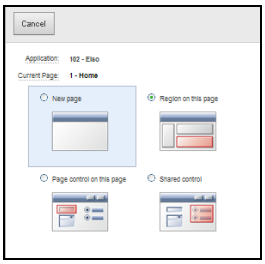
Először is kéri a belépési felhasználó nevet és jelszót, amelyet megadva, megjelenik az alkalmazásunk első lapja. Ez egyenlőre egy üres lap. A lap alján pedig a `Fejlesztői eszköztár` látható.(2.28. ábra)



2.28 ábra. *Fejlesztői eszköztár*

Viszatérhetünk a kiinduló állapothoz (`Home`), az alkalmazásokhoz (`Application`), szerkeszthetjük a megfelelő oldalakat (`EditPage 1`).

Létrehozhatunk (`Create`) új lapot a lapon `Tartományt`, vezérlő elemet ezen a lapon, valamint osztott komponenst, ahogy az 2.29. ábrán láthatjuk. A többi menüponttal most nem foglalkozunk.

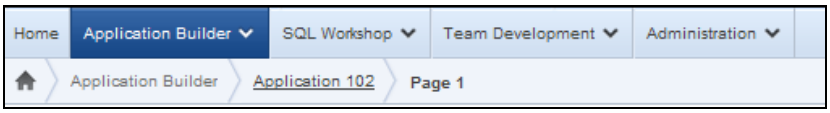


2.28 ábra. *Fejlesztői eszköztár Create gombjára kattintva*

A következő lépés a lap tartalmának létrehozása lesz. Kattintsunk alul, a fejlesztői eszköztáron (2.28. ábra) az `Edit Page 1` gombra. Így jutunk az aktuális lap szerkesztő környezetébe.

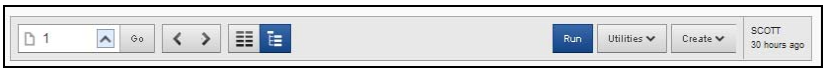
Lap (Page) felépítése, szerkesztése

Nézzük meg, milyen funkciókat is tehetünk a felhasználói funkcióknak szánt lapra. Melyek is egy lap jellemzői? Ez a lap tulajdonképpen egy html kód által generált windows ablak. Nincs mérete, a nyomtatás a szokásos `Page Setup` segítségével történhet.
A legfelső sorban láthatjuk a felhasználó számára megengedett tevékenységeket. Alatta a Visszalépő Sáv (`Breadcrumb`) mindig megjelenik minden egyes lap tetején az Application Builderben annak érdekében hogy a lapok közötti navigációs út létrejöjjön. Pillanatok alatt egy másik lapra lehet navigálni a segítségével ahogy a Visszalépő Sáv megfelelő belépési pontjára kattintunk (2.29. ábra).



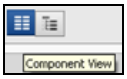
2.29 ábra. *Vissszalépő Sáv (Breadcrumb)*

A Visszalépő Sáv alatt helyezkedik el a Lap Navigációs Menüje (2.30. ábra). Erről a következő műveletek érhetők el:



2.30. ábra. *A lap navigációs menüsora*

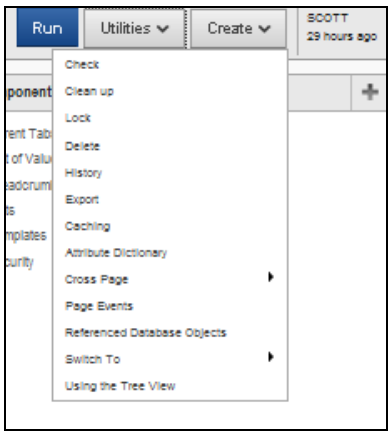
- Bal oldalon megjelenik illetve egy legördülő listából kiválaszthatjuk a szerkeszteni kívánt lapot (lapszám alapján), természetesen a kiválasztás után a `Go` gombra kell kattintani.
- Utána a két relációjel a megelőző lapot és a következő lapot adja fel.
- Fontos a következő két gomb, amely a komponenseket láttatja, illetve a megfelelő fa szerkezetüket mutatja meg (2.31. ábra).



illetve

2.31. ábra. *A lap kétféle megjelenítése*

- Jobb oldalon található gombok pedig a lapot Futtató (`Run`) gomb.
- Azután következik egy `Utilities` gomb, mely segédprogramok gyüjteménye:



Ellenőrzés
Takarítás
Zárolás
Törlés
Története
Export

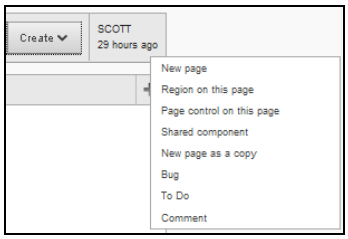
Szótár tulajdonságai

Lap eseményei

Nevek változtatása
Fa szerkezet

2.32. ábra *Segédprogramok (aktuális)*

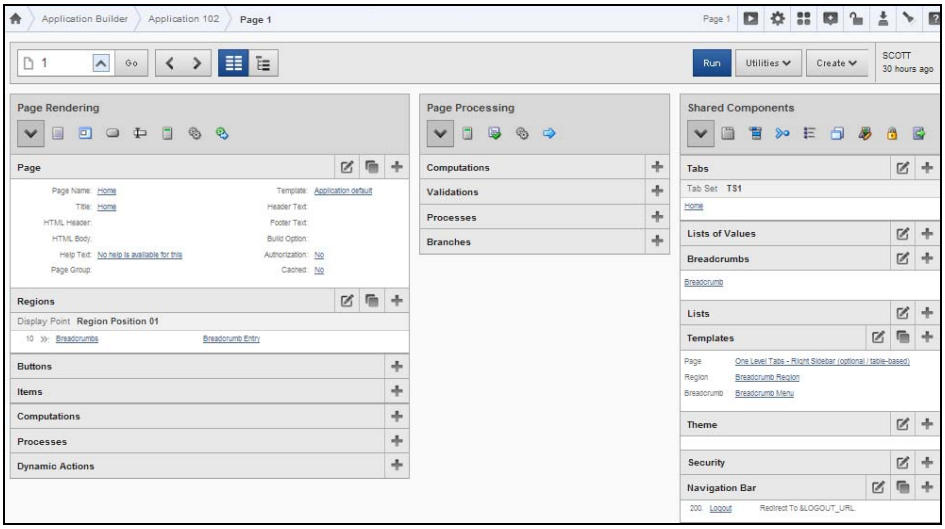
A következő gomb a Létrehozás (`Create`)gomb, amely a következőket tudja létrehozni:



2.33. ábra A Létrehozó (Create) gomb aktualitása

- Végül láthatjuk, hogy a felhasználó mikor módosította az alkalmazását (2.33. ábra).

Részletesen tárgyaljuk az ablak nagy részét kitevő `Lap szerkesztés (Page Rendering)` részt, amely lehet komponens vagy fa nézetű.



2.34. ábra. *A lap komponens nézete*

A 2.34. ábrán látható Lap (`Page 1`) ablak komponens nézetében az alkalmazói lapokat meghatározó objektumok három oszlopba csoportosítva láthatóak.

Baloldali oszlop:

Lap-Megjelenítési csoport (`Page Rendering`). Itt találhatóak a *Lap* (`Page`) tulajdonságait és megjelenését meghatározó komponensek, valamint *Tartományokat* (`Regions`), *Gombokat* (`Buttons`), Elemeket (`Items`), valamint a megjelenítéshez kapcsolódó, úgynevezett *Lapszintű Számítási Folyamatokat* (`Computations`) és *Lapszintű Általános Folyamatokat* (`Processes`) valamint *Lapszintű Dinamikus Tevékenységeket* (`Dynamic Actions`) lehet itt létrehozni és módosítani.

Középső oszlop:

Folyamat-Definíciós csoport (`Page Processing`). Itt találhatóak a lap használatához, alkalmazási logikájához kapcsolódó komponensek, az úgynevezett *Alkalmazásszintű Számítási Folyamatok* (`Computations`), a *Validációs Adatok* (`Validations`), az *Alkalmazásszintű Általános Folyamatok* (`Processes`), valamint a *Kapcsolati Folyamatok* (`Branches`).

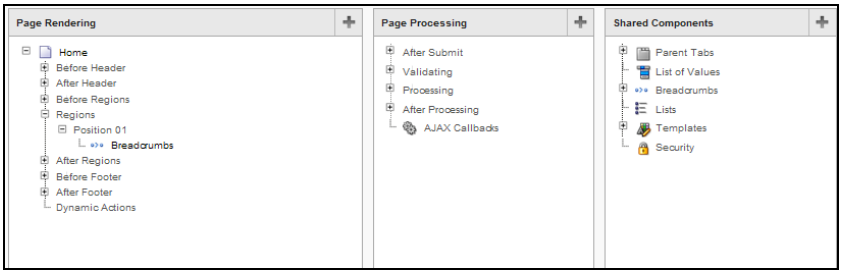
Az itt szereplő Számítási és Általános Folyamatok gyakran tartalmaznak valamilyen SQL, vagy PL/SQL utasítást, általában a Validációs Adatokkal adjuk meg az adatbevitelhez az ellenőrzési feltételeket, a Kapcsolati Folyamatok segítségével pedig a megjelenítendő új (alkalmazásbeli, vagy külső, URL-címével adott) lapot jelöljük ki.

Jobboldali oszlop:

Megosztott komponensek csoportja (`Shared Components`). Itt hozhatók létre, illetve módosíthatóak a *Fülek* (`Tabs`), az *Érték Listák* (`Lists of Values`), a *Visszalépő Sáv* (`Breadcrumbs`), a *Listák* (`Lists`), a *Felület Minta* (`Theme`), a *Sablonok* (`Templates`), a *Hitelesítés* (`Security`) és a *Navigációs Sáv* (`Navigation Bar`). Ezeket a komponenseket az alkalmazás minden lapja eléri, közösen használhatják (tehát bármelyik lap definíciós ablakából szerkeszthetjük.

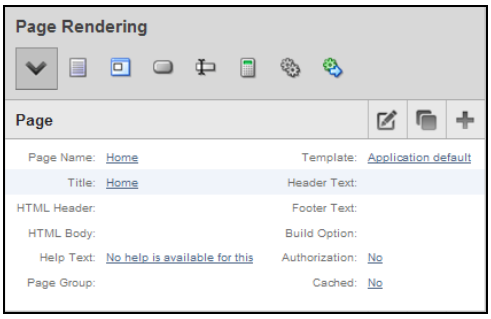
A lapokon minden komponens látható és szerkeszthető. Másolni lehet egy komponenst az egyik lapról a másikra. Az aktuális lapra lehet új komponenst is helyezni.

Nézzük meg a fa szerkezetet is. A fa szerkezetű nézet megjeleníti a tartományokat, a lap elemeit, az alkalmazási logikát, mint a facsoport csomopontjai. A fa csoport minden komponense az események sorrendjén alapul, vagyis az APEX folyamatai hogyan hajtják végre a lap előkészítését, és a végrehajtást. mindkét nézeten kialakítható és változtatható a felhasználói felület.



2.35. *A lap fa szerkezete*

A komponens nézet *Lap-Megjelenítési* csoportja, szekciója tartalmazza a lap szerkesztését, másolását és új lap létrehozását is (2.36. ábra).



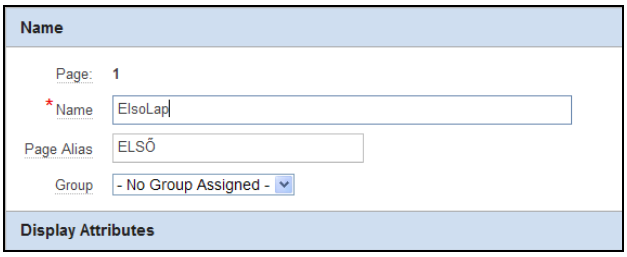
2.36. *Magának a lapnak a szerkesztése*

Szerkesztő gomb `Edit` gomb

Másolás `Copy`

Létrehozás `Create`

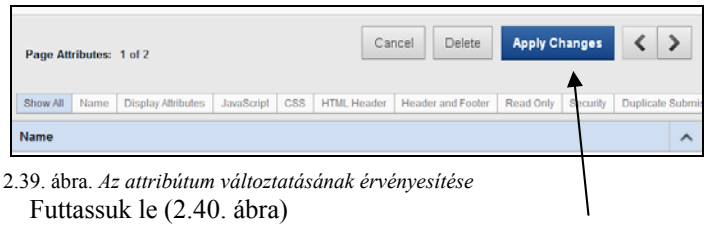
Változtassuk meg a lap fejlécét a szerkesztő gombra való kattintással. Adjunk nevet a lapnak (2.37. ábra):



2.37. ábra A Lap névadása

Változtassuk meg a `Header and Footer` attribútumot, és adjunk meg valamilyen szöveget fejlécnek, illetve láblécnek (2.38. ábra), majd menjünk vissza a gördítősávon a lap tetejére, és az `Apply Changes` gombra való kattintással érvényesítsük a módosításokat, amint az a 2.39.ábrán látható. Ellenőrizzük és futtassuk le a lapot.

2.38.ábra. *Fej és lábléc megírása*

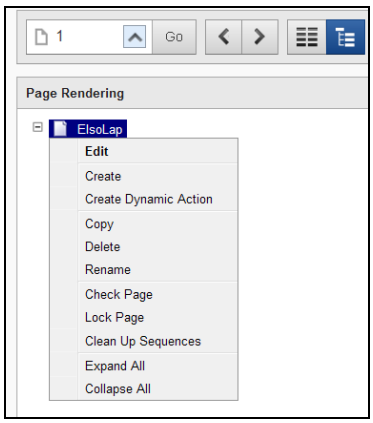


2.39. ábra. *Az attribútum változtatásának érvényesítése*

Futtassuk le (2.40. ábra)

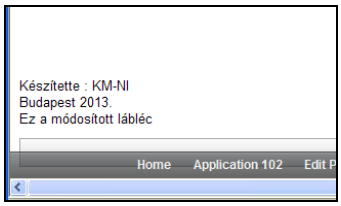
2.40. ábra *Az első lap megjelenítése*

Ugyanez megoldható a Fa nézeten is (2.41. ábra).



2.41. ábra Szerkesztés Fa nézetben

Kattintsunk a fa nézetre. Jelöljük ki az első lapot és kattintsunk a jobb gombra. A felbukkanó menüpontok közül válasszuk ki a megfelelőt, most például az `Edit` menüpontot, és módosítsuk a fej vagy láblécet. Érvényesítsük a változtatásokat és futtassuk le.

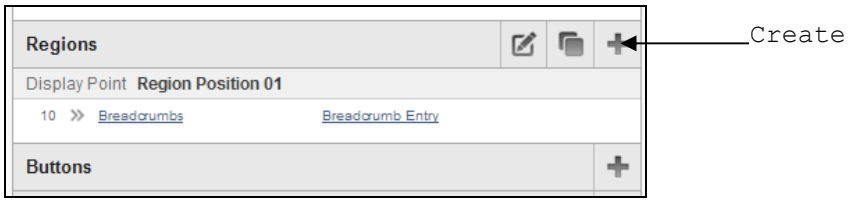


2.42. A módosított lábléc

Tartomány létrehozása

A lapon létrehozhatunk tartományokat (`region`), amelyek kiválasztott sablont használnak, és html tartalmat generálnak. Egy lapon több tartomány is létrehozható. A tartomány a komponensek csoportjának gyűjtőhelye, például szövegeket, riportokat, gombokat, és egyéb Elemeket tartalmazhat. Készíthetünk egyszerű tartományt is, amely nem készít kiegészítő html kódot, és készíthetünk szépen kidolgozott tartományt, amely lefordítja a tartalmat és kiegészítő html kódot készít.

Nézzünk egy példát. Kattintsunk a `Page Definition` lapon a `Page Rendering` oszlop `Regions` szekciójában a Létrehozás (`Create`) funkciójú + ikonra (2.43. ábra.).

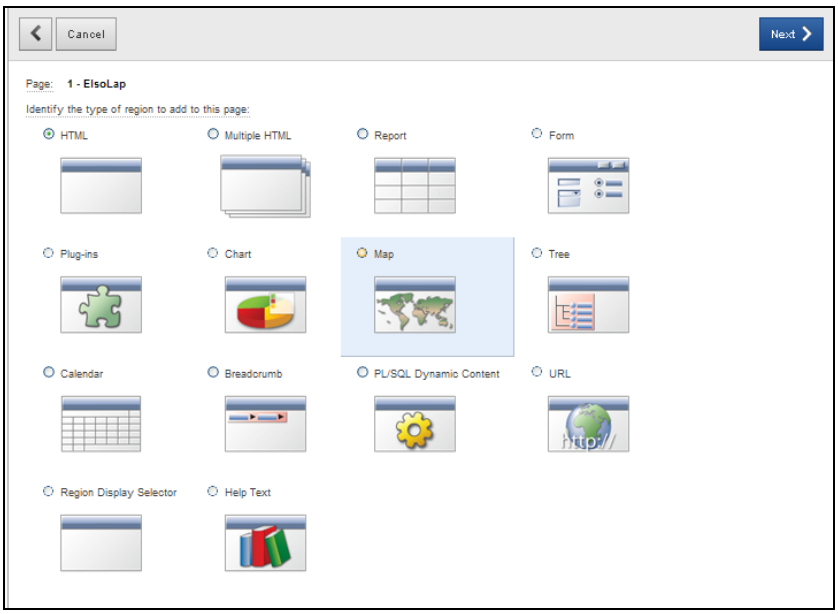


2.43. ábra. *Új tartomány létrehozása*

A következő típusú tartományok választhatók (2.44. ábra):

1. `HTML`: Az Elemek gyűjtőhelyeként funkcionál, és magába foglalja azokat a html tartalmakat, amelyekkel elláttuk.
   Lehet html szöveg, amely helyettesítő és speciális karaktereket tartalmaz.
2. `Report`: SQL lekérdezések eredményét, vagy varázslóval előállított SQL lekérdezéseket állít elő, amelyeket nem nekünk kell megírni, és ennek eredményét jeleníti meg..

A többit nem részletezzük, mert az egyes tartomány típusokat szemléltető képek elég egyértelműek (2.44. ábra).



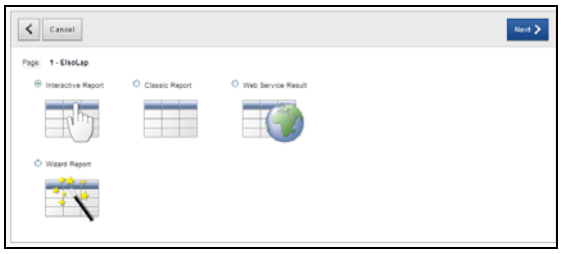
2.44. ábra. *A választható tartományok*

Riportok, Tábla-megjelenítés

Jelenítsük meg az `emp` tábla tartalmát az alkalmazásunk első lapján.

Első lépésként válasszuk amegfelelő `Report` tartományt, a megjelenő lehetőségek közül.

Választható(2.45. ábra):



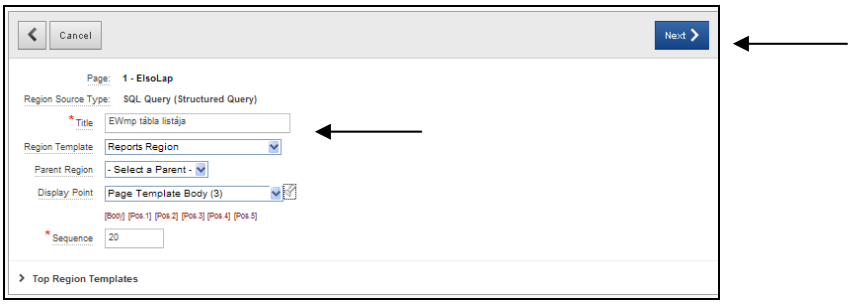
2.45. ábra. *A választható tartományok*

- *Interaktív riport* (`Interactive Report`)
  Az interaktív riport általunk írt SQL lekérdezésen alapuló interaktív riportot készít. Az SQL lekérdezés segítségére használjuk a Lekérdezés Varázslót (QueryBuilder). Ez a riport csak a munkaállomáson futó (desktop) lapon támogatott, és laponként csak egy interaktív riportot határozhatunk meg. A végfelhasználó testre szabhatja a riport elrendezését, és az adatok az Action menűből kiválasztott opciónak megfelelően jelennek meg. A riport megjelenése az előre elkészítettriport sablon alapján történik, amelyben karaktersorozat szerinti szűrés megvalósítható.

- *Klasszikus riport* (`Classic Report`)
  A Klasszikus riport a szokásos SQL lekérdezésen vagy olyan PL/SQL függvényen alapszik, amelyik egy SQL utasírtással tér vissza. Az SQL lekérdezés előállítására segítségül használjuk a Lekérdezés Varázslót.
- *Web szolgáltatás eredménye* (`Web Service Report`)
  A riport a Webszolgáltatás eredményén alapul.
- *RiportVarázsló* (`Wizard Report`)
  A riportot a hagyományos SQL lekérdezés varázslóval állítja elő, amelyhez nem szükséges SQL ismeret.Kiválaszthatjuk a megfelelő sémát, táblát oszlopokat és az eredmény megjelenését.

Válasszuk első esetben a legegyszerűbb riportkészítést a varázslóval:

A varázsló első lépésében meg kell adni a riport Fejlécét (`Title`), és a sorozatszámát (`Sequence`). A többi választható(2.46. ábra)



2.46.ábra. *Tartomány létrehozása*

A tartomány – mint látható – a több lehetséges elhelyezési pozíció (`Display Point`) közül a 3. sorszámút ajánlja fel (`Page Template Body t (3))`. Ne fogadjuk el ellenőrzés nélkül, hiszen lehet, hogy számunkra nem mindig megfelelő. A lapon az elhelyezkedési térképet az elemlámpára kattintva kapjuk meg, és kattintsunk a `Next` gombra.

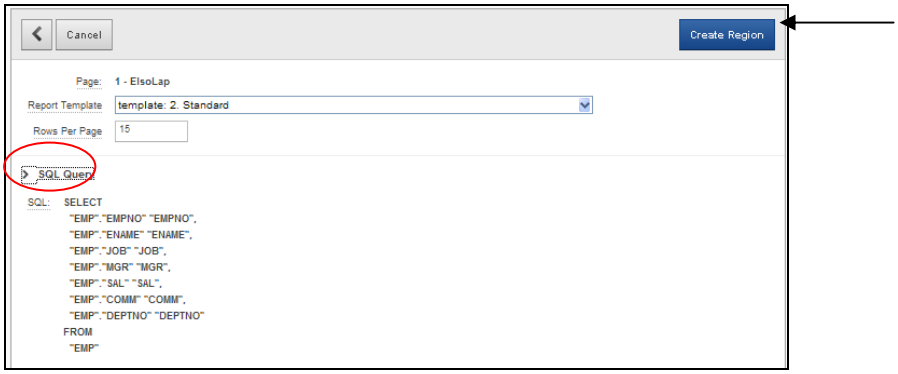
Megjegyzés

A sorozatszám (`Sequence`) mezőt minden objektumnál kötelező kitölteni illetve elfogadni a rendszer által felajánlott értéket. Természetesen meg lehet változtatni a felajánlott értéket. A rendszer ezt a sorozatszámot a létrehozás sorrendjében adja ki, és jelzi a lapon való megjelenés illetve végrehajtás sorrendjét.

A megjelenő lapon válasszuk ki az `emp` táblát, és a `hiredate` attribútum kivételével a többit (2.47. ábra), majd kattintsunk a `Next` gombra.

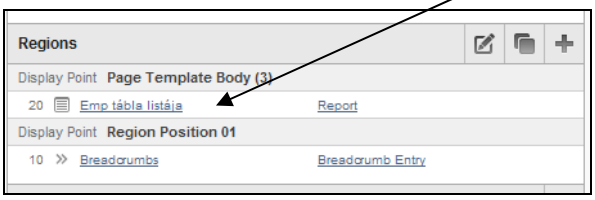
2.47. ábra.. *Az SQL lekérdezés elkészítése varázslóval*

A megjelenő ablakban ellenőrizzük az imént kijelölt oszlopokat megjelenítő SQL lekérdezést, ezért kattintsunk az `SQL Query` pontra. Az előállított lekérdezést a 2.48. ábrán láthatjuk. A megjelenés külalakját utólag is megváltoztathatjuk a megfelelő *Riport Sablon* választásával (`Report Template`).



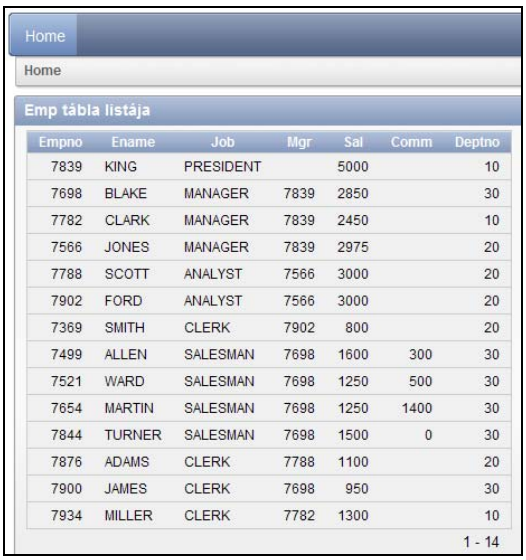
2.48. ábra. *Az előállított SQL lekérdezés*

Kattintsunk a `Create Region` gombra. A megjelenő `Page` ablakban láthatjuk, hogy elkészült a struktúra. Megjelent egy új tartomány, ahogy azt a 2.49. ábrán láthatjuk.



2.49. ábra. *Az elkészült új tartomány*

Futtassuk le az alkalmazást. A Visszalépő Sávban rákattintunk az `Application101` pontra (ez az `elso` nevű alkalmazás azonsítója), majd a Run ikonra. A futási eredmény a 2.50-as ábrán látható.



Empno	Ename	Job	Mgr	Sal	Comm	Deptno
7839	KING	PRESIDENT		5000		10
7698	BLAKE	MANAGER	7839	2850		30
7782	CLARK	MANAGER	7839	2450		10
7566	JONES	MANAGER	7839	2975		20
7788	SCOTT	ANALYST	7566	3000		20
7902	FORD	ANALYST	7566	3000		20
7369	SMITH	CLERK	7902	800		20
7499	ALLEN	SALESMAN	7698	1600	300	30
7521	WARD	SALESMAN	7698	1250	500	30
7654	MARTIN	SALESMAN	7698	1250	1400	30
7844	TURNER	SALESMAN	7698	1500	0	30
7876	ADAMS	CLERK	7788	1100		20
7900	JAMES	CLERK	7698	950		30
7934	MILLER	CLERK	7782	1300		10

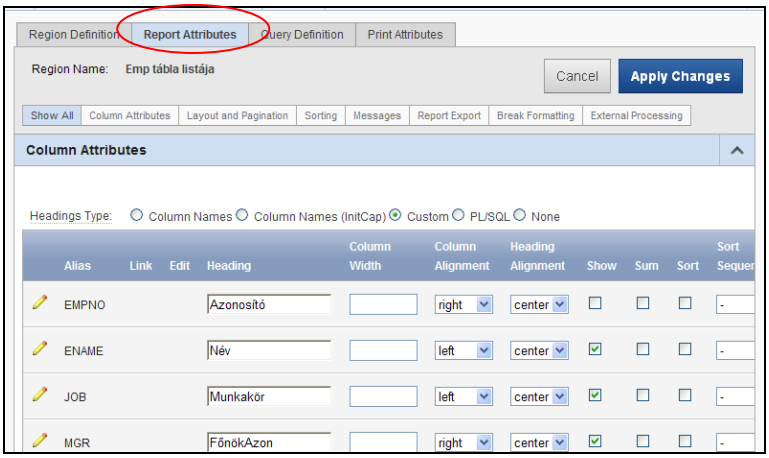
1 - 14

2.50. ábra. *A lap futási eredménye*

Ha valamit elfelejtettünk beállítani, akkor menjünk vissza `Edit Page 1` a *Fejlesztői Eszköztáron*, a Lapra, és kattintsunk a Tartomány részben `Az emp tábla listája` hiperhivatkozásra illetve a `Report hivatkozásra.`

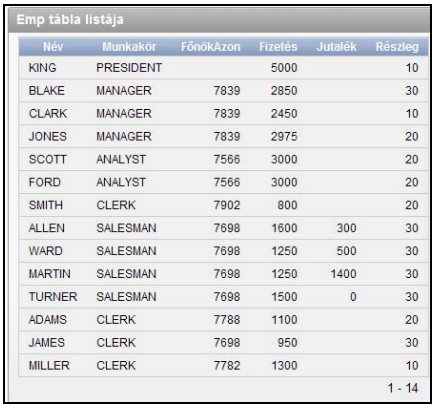
2.51. ábra *A Tartomány szekcióban megjelent a létrehozott riport*

Majd utána felül `Report Attributes` fülre. Változtassuk például a megjelenő lista fejlécét magyarra olymódon, hogy a `Custom` rádiógombot bekapcsoljuk, és átírjuk az oszlopok fejlécneveit (`Heading`), amint azt a 2.70-ábrán láthatjuk. Az oszlopokat (`Column alignment`) igazíthatjuk a szokásos módon: balra (`left`), jobbra (`right`), középre (`center`). Az oszlop fejlécét (`Heading alignment`) is ugyanígy igazíthatjuk. Ha nem jelöljük be az oszlop láthatóságot (`Show`), akkor nem látjuk a megjelenítésnél. Képezhetünk egyes oszlopokból öszeget (`Sum`), a bejelölt oszlopok szerint rendezhetjük (`Sort`) a listát, és megadhatjuk (1, 2, vagy 3) a rendezési sorrendeket (`Sort Sequence`) is. A le- és felfelé mutató nyilakra kattintva lehetővé válik az egyes oszlopok megjelenési sorrendjének beállítása. Mindez látható a 2.52. ábrán.



2.52. ábra. *A Riport jellemzőinek megváltoztatása*

Lépjünk vissza a Visszalépő Sávon az általunk létrehozott (`elso` nevű) alkalmazásra, és futtassuk le ismét (2.53. ábra).



Emp tábla listája

Név	Munkakör	FőnökAzon	Fizetés	Jutalék	Részleg
KING	PRESIDENT		5000		10
BLAKE	MANAGER	7839	2850		30
CLARK	MANAGER	7839	2450		10
JONES	MANAGER	7839	2975		20
SCOTT	ANALYST	7566	3000		20
FORD	ANALYST	7566	3000		20
SMITH	CLERK	7902	800		20
ALLEN	SALESMAN	7698	1600	300	30
WARD	SALESMAN	7698	1250	500	30
MARTIN	SALESMAN	7698	1250	1400	30
TURNER	SALESMAN	7698	1500	0	30
ADAMS	CLERK	7788	1100		20
JAMES	CLERK	7698	950		30
MILLER	CLERK	7782	1300		10

1 - 14

2.53. ábra. *Az emp tábla listája*

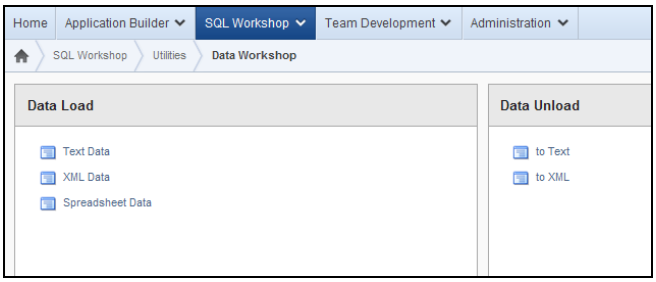
Dolgozó tábla létrehozása táblatartalmak másolásával

Legyen a feladat a következő; helyezzünk az előbbiekben létrehozott lapra az `emp` tábla listája mellé a dolgozóknak egy olyan `dolgozó` nevű listáját, amely a telephely szerint van rendezve, és eszerint a lista legyen ismétlődésmentes, valamint legyen másmilyen a riport kinézete is.

Lépjünk vissza a `Home` linkre

A táblákat (`dolgozo`, `reszleg`) statikusan elkészíthetjük az `SQL` komponens `SQL Commands` komponense segítségével a megfelelő `CREATE` utasítások beírásával és lefuttatásával (`CREATE TABLE dolgozo AS SELECT * FROM emp`, és `CREATE TABLE reszleg AS SELECT * FROM dept`), de elkészíthetjük a táblatartalmak átmásolásával az APEX grafikus komponensei segítségével. Most ezen a folyamaton megyünk végig. A műveletsor két lépésben történik. Előbb az `emp` tábla tartalmát egy átmeneti (text típusú) szö-

vegállományba tesszük, majd innen betöltjük egy `dolgozo` nevű táblába. Ehhez először vissza kell lépnünk a `Home` főablakba, amit a fenti ablak Fejlesztő Eszköztár `Home` menüpontjára való kattintással érhetünk el. A `Home` főablakban válasszuk ki a SQL Műhely (`SQLWorkshop`) komponensen belül a `Utilities` komponenst, abból pedig az Adat Műhely (`Data Workshop`) komponenst. Itt találjuk meg Kivitel és Betöltés (`Load/Unload`) folyamatot. Ennek révén visszük ki az `emp` tábla tartalmát egy text típusú fájlba (2.54. ábra)

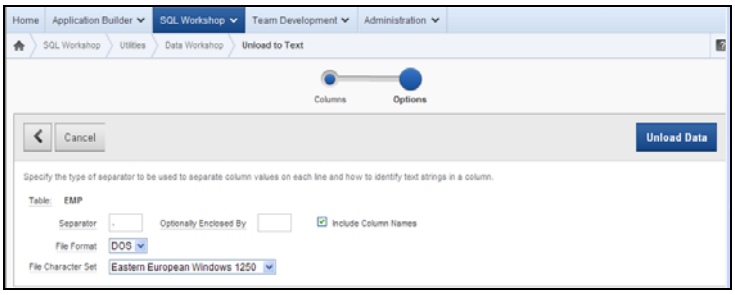


2.54. ábra. *A tábla kivitele Text vagy XML típusba*

Kattintsunk a Kivitel (`Data Unload`) ablakban a `to Text` ikonra.

2.55. ábra. *A kivitel lépéseinek sora*

A kivitel lépéseinek sorozata látható a 2.55. ábrán. Válasszuk ki a sémát, és a megfelelő táblát, valamint a jelöljük ki a megfelelő oszlopokat. `Next`.



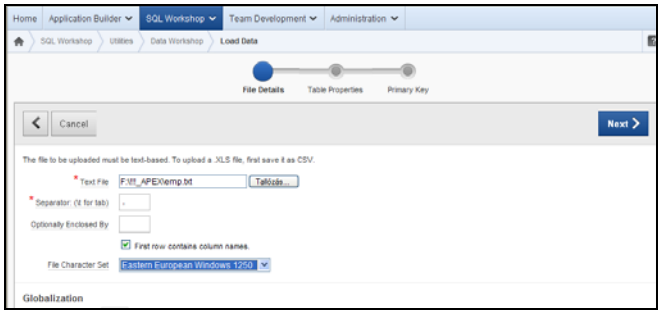
2.56. ábra *A fájl formátum, és esetleg a karakterkészlet (!) kiválasztása*

Az itt látható utolsó ablakban a jelölőnégyzet megjelölésével elérjük, hogy a text fájl tartalmazza az oszlopfejléceket is. Ügyeljünk a Fájl Formátum (`File Format`) mező és a Fájl Karakterkészletének (`File Charakter Set`) mező megfelelő beállítására.

Kattintsunk az `Unload Data` gombra.Ha sikerült a tábla kivitele text fájlba, valamint az általunk megjelölt könyvtárba, akkor onnan töltsük ezt vissza egy dolgozó nevű táblába (2.57. ábra).

2.57 ábra. *Új tábla létrehozása text fájl betöltésével.*

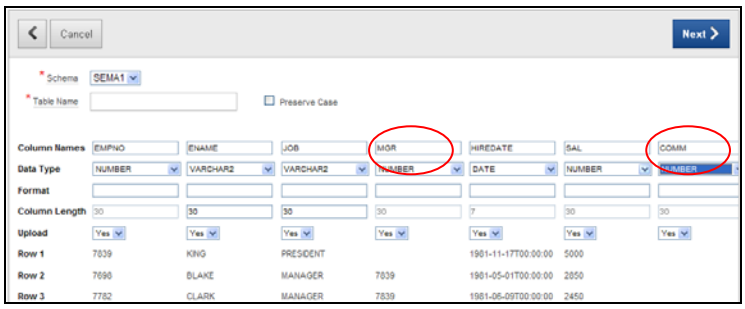
Lépjünk vissza a Visszalépő Sávon a `Data Workshop` mezőre és a lépések sorának megfelelően töltsük ki illetve válasszuk ki a megfelelő adatokat.kattintsunk a `Data Load` ablakrészben a `Text Data` ikonra. Új táblát hozunk létre, vesszővel elválasztott adatokkal, majd `Next`.



2.58. ábra *A fájl betöltése*

Tallózuk a fájlt (2.58 ábra), állítsuk be a elválasztó (`separator`) karaktert és Karakterkészletet (`File Charakter Set`) mezőt, majd `Next`.

Felhívjuk a figyelmet arra, hogy a tábla tulajdonságainak beállításánál, az oszlopneveket meg is tartotta a visszavitel, azonban az adattípusokat nem teljesen. Tehát azt feltétlenül állítsuk be az eredeti tábla oszlopmezőinek megfelelően, és adjuk meg az elsődleges megszorítást is. Beállíthatjuk az oszlop szélességét,valamint az `Upload` sorban azt az oszlopot hagyjuk `YES` értéken, amelyiket be akarjuk vinni (2.59. ábra).



2.59. ábra *Az adatok betöltésének következő lépése*

A teljes ellenőrzés, és beállítás után kattintsunk a `Next` gombra. itt állíthatjuk be az Elsődleges Kulcsot (2.60. ábra).

2.60. ábra *Az elsődleges kulcs beállítása*

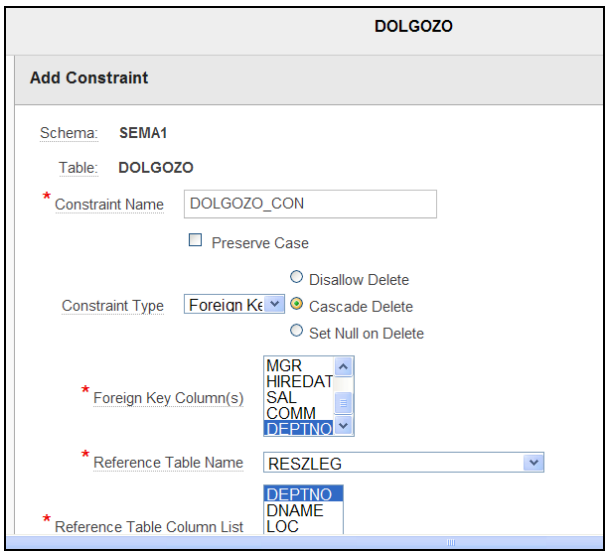
A mezők választása jelen esetben a következő:

- `Primary Key From` : Válasszuk a használjuk a létező oszlopokat (`Use an existing column`) rádiógombot. Ne hozzunklétre új oszlopot!
- `Primary Key`: Válasszuk elsődleges kulcsnak az `empno` oszlopot.
- `PK Constraint Name`: Elfogadhatjuk a felajánlottat de adhatunk más nevet is.
- `Primary Key Population`: Ne generáljunk új szekvenciát. Válasszuk tehát a `Not generated` rádiógombot.

Majd kattintsunk a `Load Data` gombra.

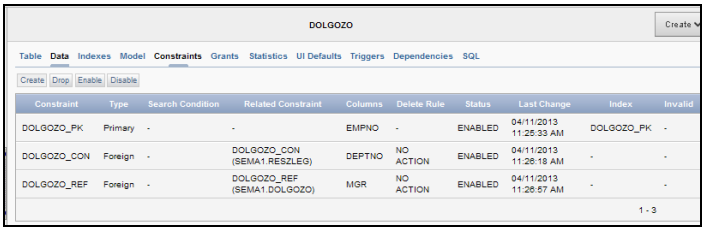
Ugyanígy két lépésben készítsük el `reszleg` táblát is. Elkészült tehát a `dolgozo` és a `reszleg` táblánk, azonban nem teljesen egyeznek meg az `emp` és `dept` táblákkal. Hiányzik az `emp` tábla idegen kulcs megszoritása, amely a `dept` tábla `deptno` oszlopára vonatkozik. Hiányzik az `empno` és `mgr` oszlopokat összerendelő hivatkozási megszoritás is. Ezeket az objektum megjelenítő (`Object Browser`) komponens segítségével hozhatjuk létre.

A `dolgozo` táblát jelenítsük meg az `OB` objektum megjelenítő ablakban. A felső menüsorban találhatók a megszorítások (`Constraints`), először erre kattintsunk, majd az alsó menüsorban a `Create` menüpontra.(2.61. ábra). Töltsük ezt ki megfelelően. `Next`. Mmajd ellenőrizzük a létrejövő kulcs adatait és Finish.



2.61. ábra. *A dolgozo tábla idegen kulcs megszoritásának léterhozása*

Ugyanígy `Foreig Key` típus kiválasztással hozzuk létre a hivatkozási megszorítást, melyben a referencia tábla természetesen szintén a `dolgozo` tábla. (mgr -empno)
Nézzük meg a kulcsokat:



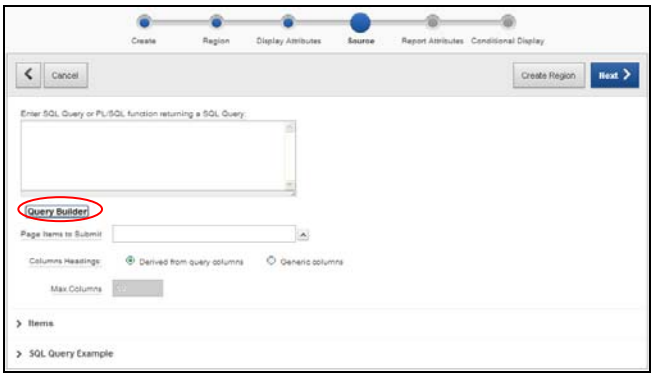
2.62. ábra A létrehozott kulcsok

Ezután térjünk vissza az alkalmazáshoz.
Egészítsük ki az általunk létrehozott (`elso` nevű) alkalmazás első lapját egy új tartománynyal. Kattintsunk a `Region` szekcióban a Létrehozás (`Create`) ikonra (a plusz jelre), válasszuk ismét a `Report` tartományt, majd a riport megvalósításánál, most ne a varázslót, hanem a Klasszikus (`Classic Report`) riport komponenst, majd `Next`.

.Írjuk be a kötelezöen megadandó cimkét (`Title`), legyen a tartomány sablonja `Report Region` , egyenlőre mást ne változtassunk meg, majd `Next`.

Ebben a tartományban a telephely szerint a dolgozók listáját szeretnénk elkészíteni. Itt is két lehetőségünk van, a varázsló használata és az utasítás beírás (2.63. ábra). Fontos, hogy szintaktikailag helyesen írjuk meg a lekérdezést, ezért amikor csak tehetjük, használjuk a varázslót. (Ha nem használjuk a `Query Builder`-t, akkor természetesen nekünk kell beírni a lekérdezést. Nézzük meg, hogy milyen a lekérdezés szintaktikája egy kicsit lejjebb, az `SQL Query` feliratra kattintva, és utána írjuk meg a lekérdezést. Az SQL utasítás megírása után ellenőrizzük szintaktikailag, majd kattintsunk a `Create Region` gombra.)

Most viszont már használjuk a lekérdezés tervezőt. Kattintsunk a `Query Builder` gombra (2.63. ábra).



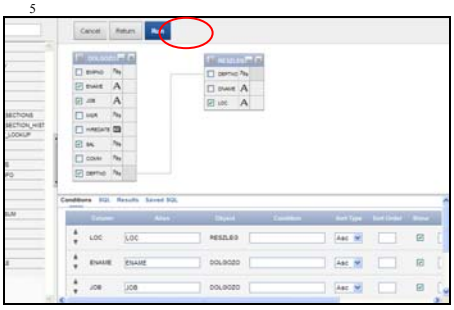
2.63. ábra. *A lekérdezés megírásának lehetőségei*

A megjelenő lap baloldalán kattintsunk azokra a táblákra, amelyekből elő szeretnénk állítani a lekérdezést. Ha egy tábla nevére rákattintottunk, akkor kis idő múlva a tábla szerkezete megjelenik a tervező felületen.

Jelöljük meg azokat az oszlopokat, amelyeket szerepeltetni szeretnénk a lekérdezésben. Elsőként a `Loc` (telephely) attribútumot jelöljük ki, annak érdekében, hogy a riportban ez legyen az első megjelenő oszlop. A `Query Builder` komponens használatát már ismertettük.

Kattintsunk mindkét táblán a `deptno` attribútumot szimbolizáló sor jobboldali négyzetébe, így kapcsoljuk össze a két táblát. Nézzük meg az SQL lekérdezést. Kattintsunk az alsó részen az SQL linkre. Futtassuk le, azaz kattintsunk a `Run` gombra (2.64. ábra).

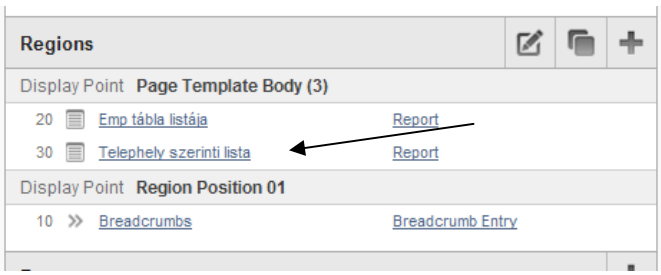
5



2.64. ábra *Lekérdezés Varázsló*

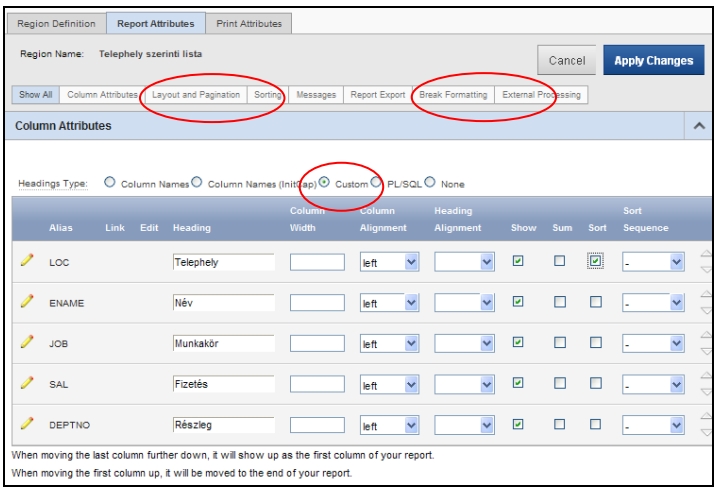
Ha a lekérdezés megfelelő eredményt ad, akkor `Return` gombra kattintva visszatérünk a `Create Region` lapra. Itt utólag is megváltoztathatjuk az SQL utasítást (a riportot). Kattintsunk a `Create Region` gombra. Itt megtalálható a `Next` gomb is , akkor feltételt is adhatunk a riport megjelenéséhez. `Next` gombra.

A `Regions` szekcióban megjelent a létrehozott riportot tartalmazó tartomány korábban megadott neve a `30`-es számmal sorszámozva (2.40. ábra).



2.65. ábra. *A lap megvalósításának tartomány szekciója*

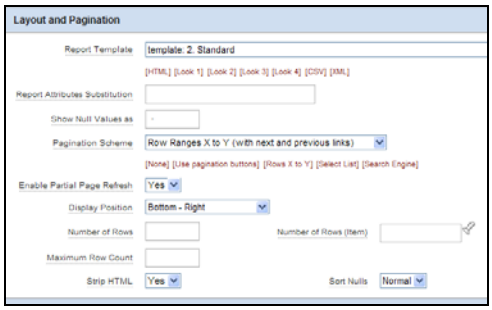
A `Telephely szerinti lista`" sorában a `Report` linkre kattintva (2.65. ábra), módosítsuk a riportot úgy, hogy telephely (`Loc`) szerinti rendezést tartalmazzon, a feliratok magyar nyelvűek legyenek, valamint biztosítsuk az ismétlődésmentes kiíratást a telephely oszlopban. Mindezt a megjelenő `Report Attributes` lapon tehetjük meg.



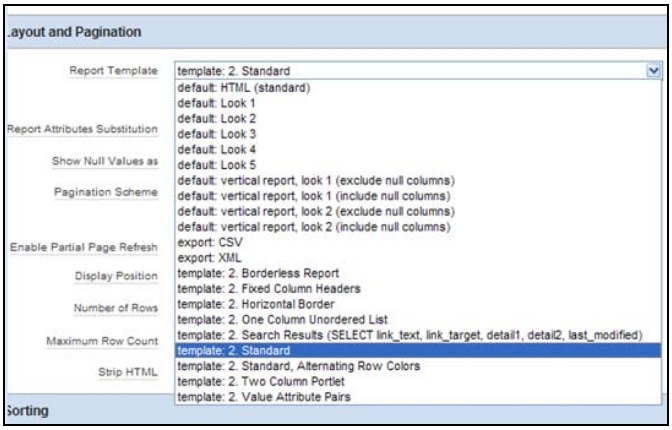
2.66. ábra. *A riport utólagos módosításanak elérése, és lehetőségei*

Itt lehet beállítani a riportok jellemzőit: (2.41. ábra)

- Az oszlop jellemzőket (`Column Attributes`)
- Az elrendezéseket és a laptörést (`Layout` and `Pagination`)
  (Például itt állíthatjuk be a riport kinézetét 2.67. ábra- 2.68. ábra)

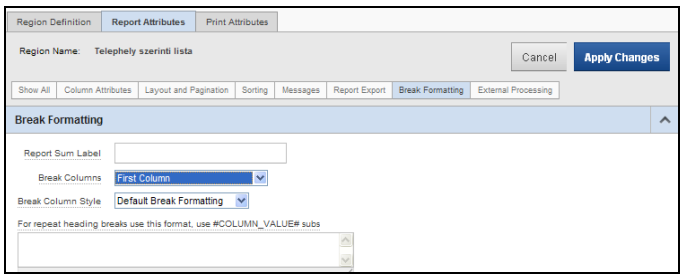


2.67. ábra A riport megjelenésének beállítása



2.68. ábra. *A riportok külalakjai*

- A rendezést (`Sorting`)
- Üzeneteket (`Messages`)
- A riport exportálását (`Report Export`)
- Az ismétlődésmentes kiíratást (`Break Formatting`) (2.69. ábra)

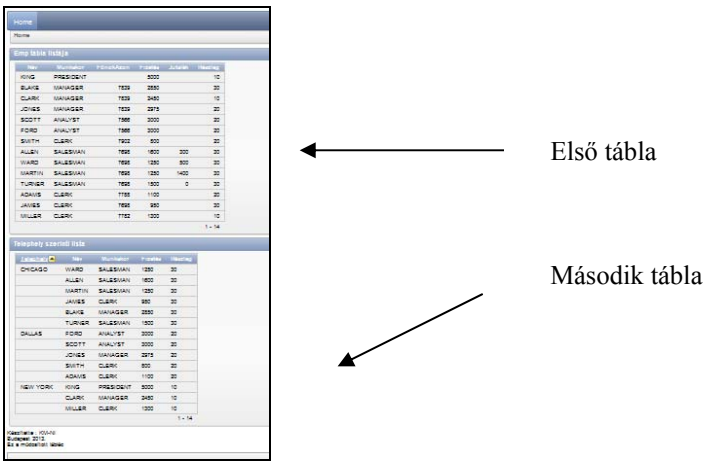


2.69.ábra *Ismétlésmentes beállítás*

- A külső folyamatokat (`External Processing`)

Módosítsuk a listát oly módon, hogy az egymás alatt ismétlődő telephelyeket csak egyszer írja ki. Ennek érdekében beállítjuk a Megszakított Listázás formázási paramétereit (`Break Formatting`). Először a `Breaks` paramétert állítjuk be (`First Column`, ekkor csak az első oszlopra alkalmazza a Megszakított Listázást), majd a Megszakított Listázás típusaként (`Identify how you would like your breaks to displayed`) az alapértelmezett esetet (`Default Break Formatting`). Ez biztosítja azt, hogy az egymás alatti azonos mezőértékek csak egyszer jelenjenek meg (2.69. ábra).

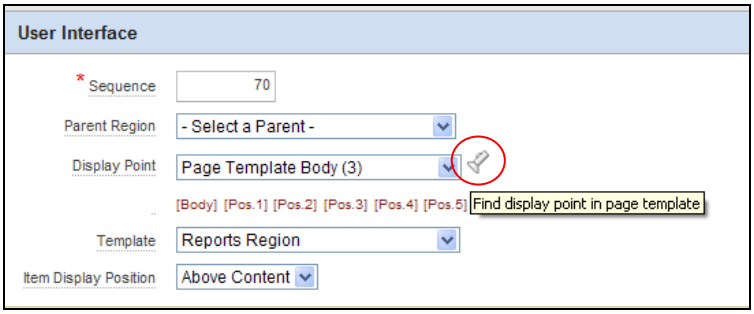
Megjegyezzük, hogy a másik esetben (`Repeat headings on Break`) az első oszlop azonos értékei szerint egymás alatt elhelyezkedő résztáblázatokat készít.
Alkalmazzuk a módosításokat (`Apply Changes`), figyeljük a hibaüzeneteket, majd futtassuk le a lapot. Láthatjuk, hogy a másik tábla alá helyezi el (2.70. ábra)).



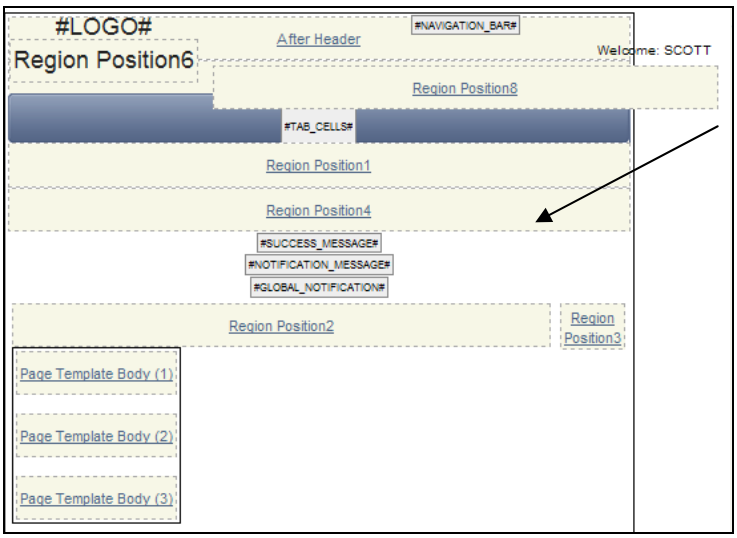
2.70.ábra *A megjelenő futási kép*

Ez így nem nagyon szép. Tulajdonképpen a két táblát egymás mellé kellene tenni. Ha ezt szeretnénk, menjünk vissza a Fejlesztői Eszköztáron az `Edit Page1`-re. Kattintsunk a `Telephely szerinti lista` hiperhivatkozásra, és az alapértelmezés szerint megjelenő `Region Definition` lapon és a `User Interface` jellemzőnél vizsgáljuk meg a `Diplay Point` gördülő listában a lehetőségeinket. Kattintsunk az elemlámpára, a felbukkanó segédablakban láthatjuk a lehetőségeket.(2.4. ábra)

Az elemlámpára kattintás után megjelennek az adott sablonnál lévő tartomány elhelyezkedési lehetőségek. Kijelölhetjük a nekünk tetsző tartomány pozíció-tipust, valamint oszlopszámot. Ez utóbbi révén hozhatunk létre egymás melletti régiókat. Választhatjuk a `Region Pos03`-at és ennek első oszlopát, de lehet az eredeti `Page Template Body(3)` beállításnál is például a 2. oszlopot választani. Mindkét esetben a régiók egymás mellett lesznek.

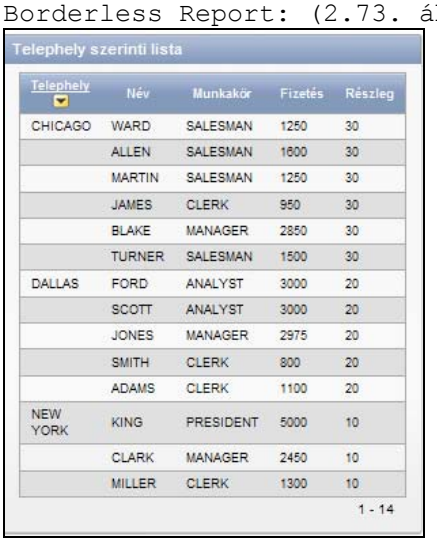


2.71. ábra. *A elemlámpa (a tartomány elhelyezés kiválasztása)*



2.72. ábra. *A tartomány pozíciókijelölési segédlete*

Állítsuk be a 3. pozíciót (2.72. ábra). Alkalmazzuk a módosításokat, Módosítsuk a riport kinézetét is. Kattintsunk a `Report Attributes` fülre, A `Layout and Pagination` részben állitsunk be más sablont. `pl template 2: Standard, Alternatíve Row Colors` vagy álaszthatunk más külső megjelenítési formát is pl `template 2: Borderless Report: (2.73. ábra)`



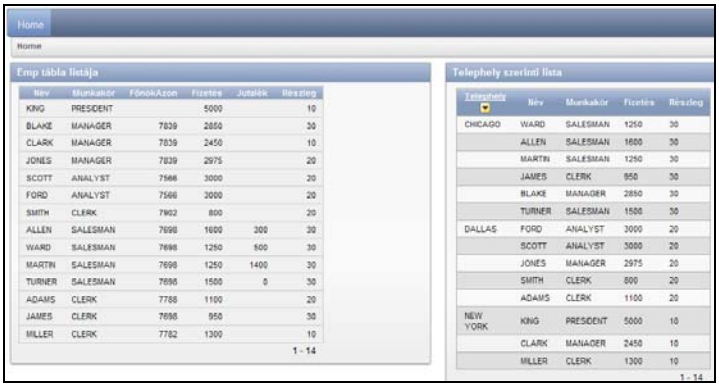
Telephely szerinti lista

Telephely	Név	Munkakör	Fizetés	Részleg
CHICAGO	WARD	SALESMAN	1250	30
	ALLEN	SALESMAN	1600	30
	MARTIN	SALESMAN	1250	30
	JAMES	CLERK	950	30
	BLAKE	MANAGER	2850	30
	TURNER	SALESMAN	1500	30
DALLAS	FORD	ANALYST	3000	20
	SCOTT	ANALYST	3000	20
	JONES	MANAGER	2975	20
	SMITH	CLERK	800	20
	ADAMS	CLERK	1100	20
NEW YORK	KING	PRESIDENT	5000	10
	CLARK	MANAGER	2450	10
	MILLER	CLERK	1300	10

1 - 14

2.73.ábra. *Egy másfajta lista kinézete.*

Most futtassuk le a lapot. (



2.76. ábra A két riport egymás mellett

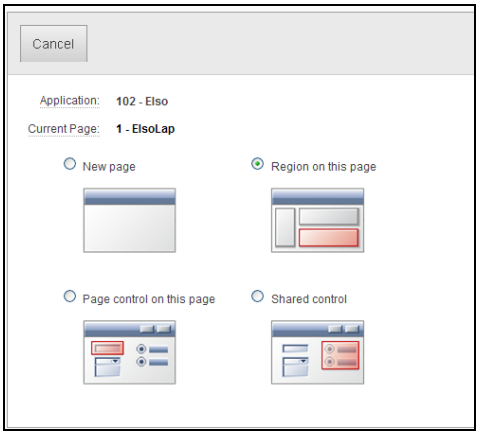
A két lista a teljes lapon elég messze van, ha a `Region Pos03`-at választottuk.

Megjegyzés. A tartományok elhelyezkedése a választott Mintától (`Temlate`)-től függ. ha más mintát választunk, akkor lehet, hogy másképp kell megválasztani a Megjelenési pontot (`Display Point`). Erre nagyon figyeljünk.

Következő Feladatunk A telephelyek szerinti átlagot jelenítsük meg.:

Szeretnénk egy lekérdezés eredményét a kettő riport közé helyezni helyezni. A következőképpen tehető meg. Hozzunk létre egy új tartományt.

A Fejlesztői Eszköztáron kattintsunk a `Create` komponensre.Megjelenik a Lap szerkesztő ablaka.

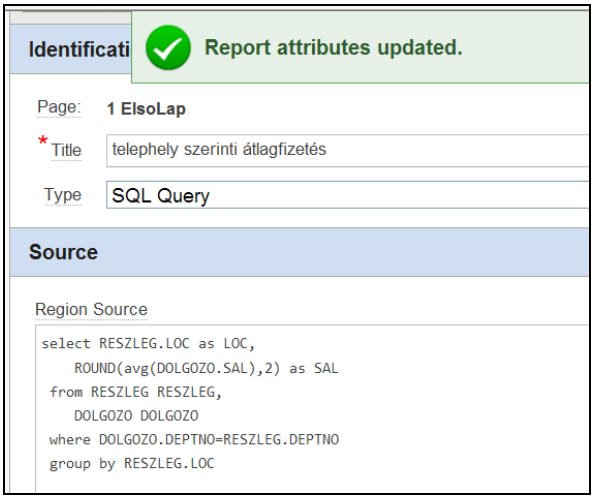


2.77. ábra. *A fejlesztői sávon a Create menüpontra kattintva ez a lap jelenik meg*

Az ekkor a megjelenő ablak látható a 2.77. ábrán. A `Region on this page` rádiógombot választva, kattintsunk a `Next` gombra. Válasszuk a `Report` rádiógombot, majd újra Klasszikus riport komponenst, a `Next` gombot, majd válasszuk az `SQL` lekérdezés, a `Next` gombot. A megjelenő `Create Region` ablakban írjuk be a `Title` mezőbe a riport fejlécét, valamint fogadjuk el a felajánlott szekvencia értéket (`sequence`), és állítsuk be a riport helyzetét, a vizszintes elhelyezkedést.

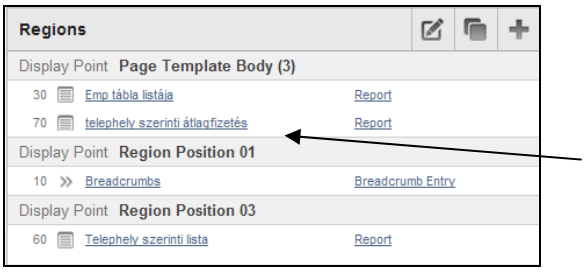
User Interface legyen az első riport tartományában, de mellette. Ezért válasszuk a Belépési Pontnak a Page Temlate Body(3)-t. Next.

Itt válasszuk a Query Builder-t. Állítsuk össze a lekérdezést, és futtassuk le. Problémát okoz az, hogy az átlag értéke valós törtként jelenik meg. Itt ezt nem tudjuk megváltoztatni, ezért térjünk vissza és az SQL lekérdezésnél a Region Source-hoz, és az AVG függvény elé írjuk a ROUND függvényt, megadva hozzá a 2 tizedes pontosságot (2.78. ábra). Kattintsunk az Create Region gombra a lap tetején.



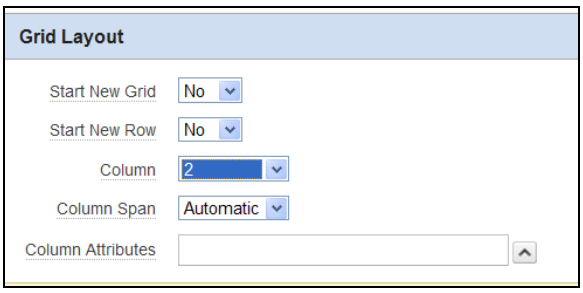
2.78. ábra. *A forrást meg lehet változtatni*

Kattintsunk a telephely szerinti átlagfizetés linkre, akkor módosítani tudju a lap megjelenésének kinézetét.



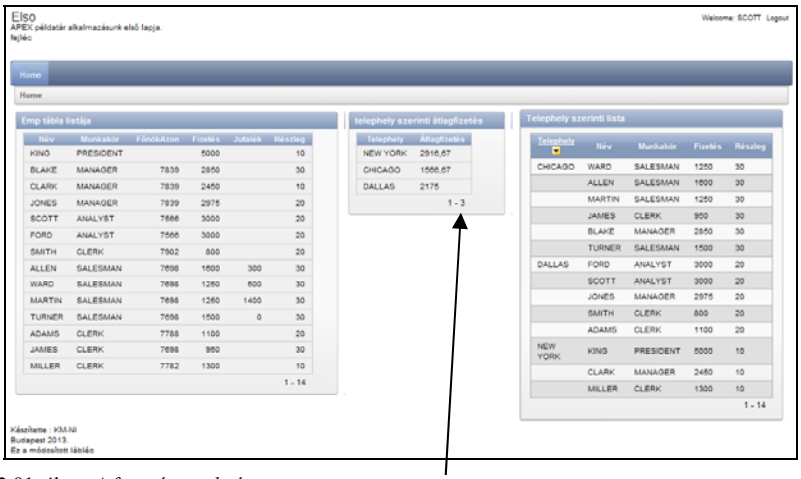
2.79. ábra A *Tartomány szekció*

A Rács (Grid)helyzetének pedig
- ne kezdjen új rácsvonalknál Start New Grid legyen NO
- ne kezdjen új sorban Start New Row legyen NO
- a terület második oszlopába helyezzük Column legyen 2
- A többit ne változtassuk (2.80. ábra)



2.80. ábra. *Új tartomány létrehozása az az első belépési pont második oszlopába*

Futtassuk le a lapot. Egy futási eredmény a 2.81. ábrán látható. A megjelenítés és az elhelyezkedés tehát igen változatos lehet.



2.81. ábra. *A futtatás eredménye*

Ugyanazon tartományi pozíció 2. oszlopában levő lista

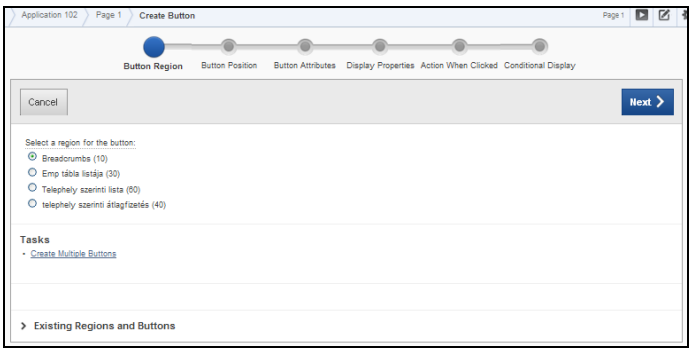
Gomb (Button)

A gomb a felhasználói felületen elhelyezett olyan grafikai objektum, melynek segítségével a felhasználó valamilyen azonnali tevékenységet tud kezdeményezni.

Az elvégzendő tevékenység jellemzően valamilyen feldolgozás, az alkalmazás egy másik lapjára való átlépés, vagy a vezérlés egy megadott web-cimre való átirányítása.

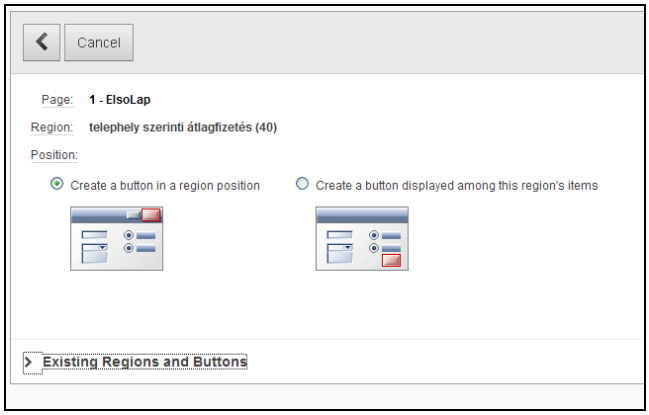
A gomb létrehozása a Komponens nézetben.

A gombok létrehozását is egy varázsló segíti, melyet a `Page Lap` lapon a `Buttons` szekcióban a Létrehozás (plusz jelű, `Create)` ikonra való kattintással indíthatunk el. Az alábbiakban ennek főbb lépésein megyünk végig (2.82. ábra).



2.82.ábra. *A gomb létrehozásának lépései*

1. Ki kell választani a tartományt, ahol a gomb megjelenik.
2. Meg kell határozni a tartományon belüli helyzetét (2.83. ábra). Például ha a gombot közvetlenül a szövegmező mellett szeretnénk látni, azaz tartományon belül, akkor „`Create button in a region position`” helyzetet állítsuk be.A helyzet meghatározott a tartomány sablonja által.

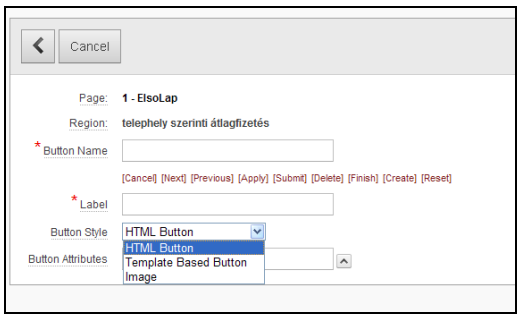


2.83. ábra. *A gomb pozíciójának kétféle lehetősége*

A másik (`Create button displayed among this region's items`) beállítás a gombot úgy kezeli, mint egy Elemet, és helyzetét is így határozza meg, egyúttal megjelenítve az Elem-definíciós szekcióban (`Items`) is. Nevezzük ezért az ilyenmódon pozícionált elemeket *Elem-típusú* gomboknak, míg az előzőeket *nem-Elem-típusú* gomboknak. Bár a két típus között különbség látszólag csupán az elhelyezésben van, több lényeges tulajdonságuk is más.

3.A.)Ha a nem-Elem gombtípust választjuk (és kattintunk a `Next` gombra), akkor a következő ablakban adhatjuk meg a gomb nevét (mint változó nevet, `Button Name`).Ez a gombnév a lap tevékenység (submit page) számára jelent azonosító értéket.(pl ha a gomb neve `RESET` (flow reset button) akkor a gomb lenyomásának célja, hogy tevékenység nélkülvisszaállítsa a lapot.

Állítsuk be a címkéjét (mint megjelenő nevet, `Label`), és a megjelenés stílusát (`Button Style`), amely három féle lehet: `HTML Button`, `Template Based Button` és `Image`. (2.84. ábra)



2.84. ábra. *A gomb nevének típusának és Akciójának beállítása*

Ha a gomb egy kép, akkor a kép nevét kell megadni, ha a gomb sablon stílusú (`Template`) akkor a sablon nevét kell kiválasztani:

4.A.)A varázsló következő ablakában a gomb elhelyezkedését és sorrendjét adhatjuk meg.

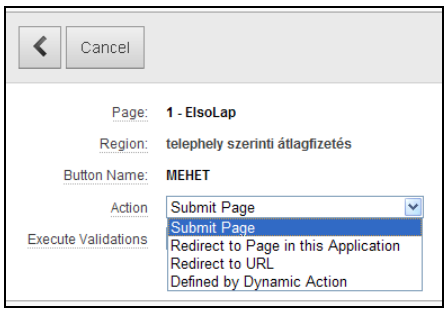
2.85. ábra A gomb pozíciója, és igazítása

Itt megadhatjuk a sorrendet (`sequence`), a tartományban elfoglalt helyzetét (`Position`), és végül igazíthatjuk Balra (`Left`) vagy Jobbra (`Right`) (2.85. ábra).

5.A.)Végül az `Action` beállítással döntjük el azt (2.86. ábra), hogy a gomb által egy a web-szerver számára feldolgozásra átadandó folyamatot akarunk-e kezdeményezni (`Submit page`), vagy egy másik alkalmazás-felületi lapra, vagy tevékenység nélkül átirányítjuk a vezérlést. Az első csoportba tartozó gombokat *Feldolgozó* típusúaknak, míg a másik csoportba tartozókat *Átirányító* típusúaknak nevezzük. Emlékeztetünk arra, hogy az egyes beállító mezők mellett a piros csillag jelzi, ha beállításuk kötelező.

Választhatunk :

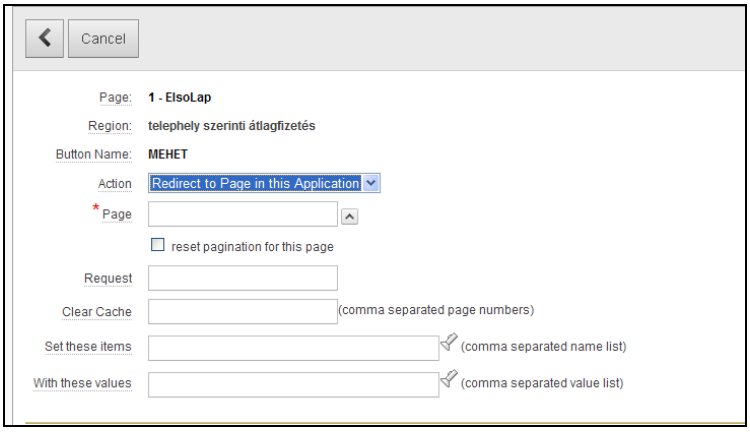
- A lap adatbázisba küldéséről (`Submit Page`)



2.86.ábra *A Gomb tevékenységei (Action)*

- A lap átirányításárol az alkalmazás másik lapjára (`Redirect to Page in this Application`)

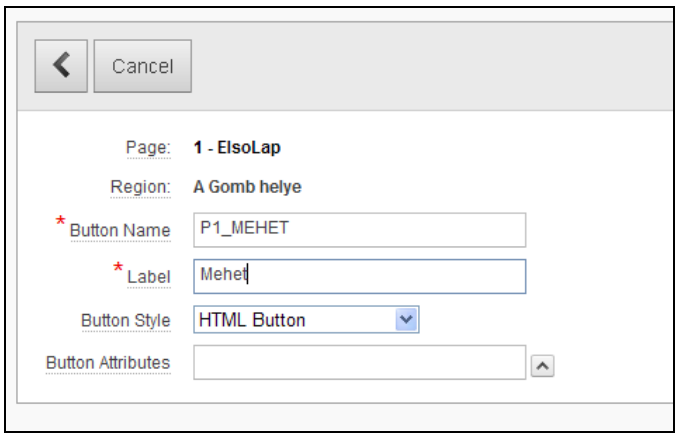
Ha ezt választjuk kinyílik az ablak, és meg kell adnunk hogy melyik lapra ugorjon. Bejelölhetjük a lapszámozást (`reset pagination for this page`), beállíthatjuk a Kérés azonosítóját (`request`), törölhetjük az állományt (`cache`),stb.



2.87.ábra *Az Átirányító gomb jellemzői*

- Az URL címre irányításáról
- Dinamikus Tevékenység határozza meg (`Defined by Dynamic Action`)

3.B.)Ha az Elem gombtípust választjuk (`Create button displayed among this region's items`, és kattintunk a `Next` gombra), akkor az alábbi ablak jelenik meg (2.88. ábra)



2.88. ábra. *Elem gombtípus tulajdonságai, request megadása*

Itt az Elemek nevéhez hasonló szerkezetű névmegadást kínál fel a varázsló. A megjelenítési és elhelyezkedési sorrendet követően adhato meg a megjelenítendő név (`Label`). `Next`.

A következő ablakban elfogadjuk, vagy beállítjuk a sorrendet (`Sequence`). `Next`.

Megjegyzés

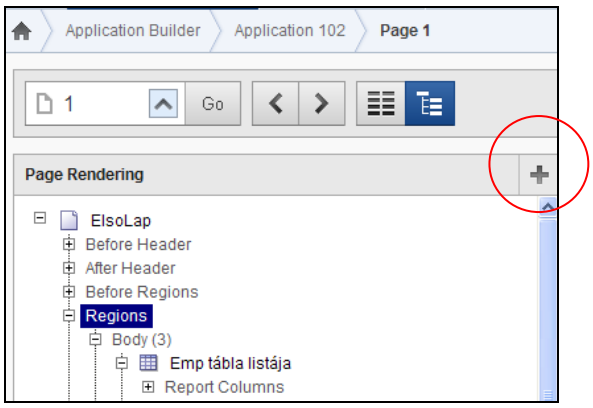
A gomb sorozatszámát (`Sequence`) elfogadhatjuk. Változtatni akkor szükséges, ha utólag szertnénk olyan gombot létrehozni, amelyik a lapon fizikailag előbb jelenik meg, tehát balrább van egy azonos mezőn belül. Tehát a gombok elhelyezkedési sorrendjét befolyásolhatjuk a soirozatszám megváltoztatásával. Ez bizonyos esetekben a külső megjelenésnél nagyon fontos, hogyan helyezkednek el a gombok egymás után. A gomboknak a lapon nem megfelelő helye sok esetben zavaró lehet, vagy egyáltalán nem jelenik meg.

5.B.)A megjelenő ablakban állitjuk be a tevékenységet, amely azonos a fent leírt nem_Elem típusu gomb lehetőségeivel

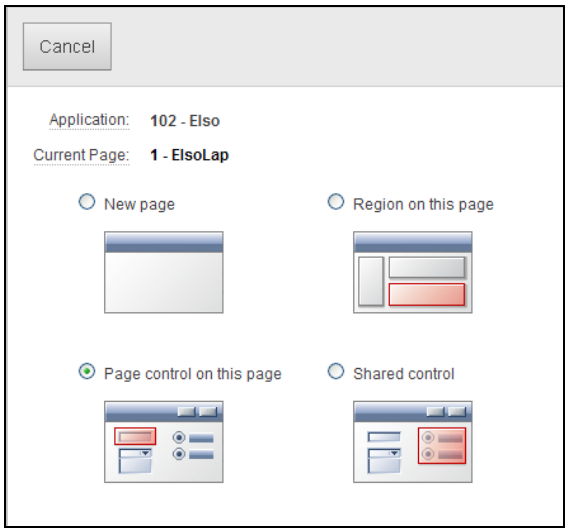
Itt a az `Action : Submit Page` legyen, azaz küldje az adatbázisba és ott dolgozza fel.

A gomb létrehozása a Fa nézetben.

Lépjünk be a Lap Definiciós nézetbe, ahol választhatunk a Komponens illetve fa nézet között. Válasszuk a fa nézetet, és kattintsunk a Létrehozás (+) (`Create`), jelre.

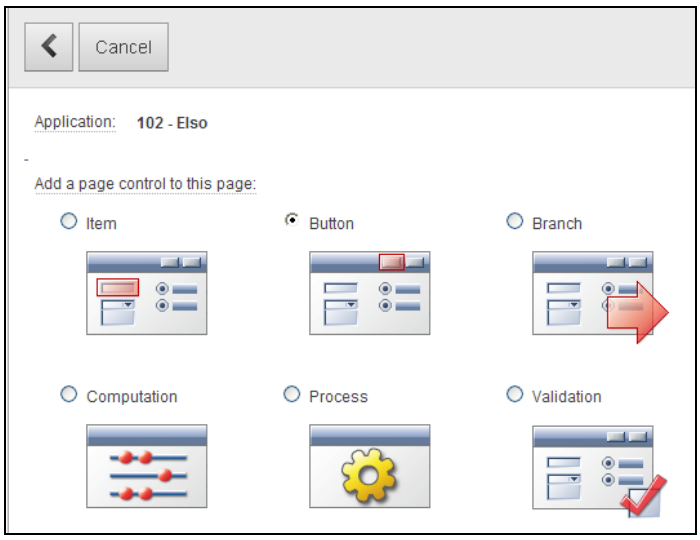


A megjelenő ablakban válasszuk a létrehozandó komponensnek megfeleló rádiógombot. Jelen esetben a Lap komponens létrehozása a lapon (`Page control on this page`), majd `Next`.



2.89.ábra *Fa nézetben létrehozhato új komponensek*

A következő ablakban válasszuk ki a lehetséges elemek közül a megfelelőt. Legyen most a Gomb (`Button`), és `Next`.

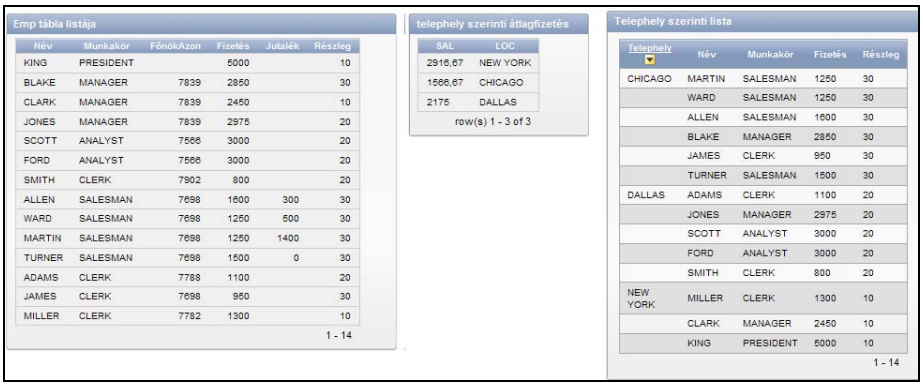


ábra

Ezek után a többi ablak azonos a komponens nézet által létrehozott Lap komponensek létrehozásával, amit fent már leírtunk.

3. fejezet
Saját alkalmazás készítésének folytatása

Folytassuk a korábban elkezdett web-es adatbázis-alkalmazásunk létrehozását. Alkalmazásunk használja az `emp` és `dept` táblákból létrehozott `dolgozo` és `reszleg` táblákat. Vegyünk fel a már korábban elkészített lapon egy gombot, amelyre való kattintás hatására megjelenik egy másik lap egy kéttáblás lekérdezés eredményét megjelenítve. A második lapon is legyen egy gomb, amivel visszatérünk az első, kiinduló lapra. Hozzunk létre egy harmadik lapot, de ezt csak fülre való kattintással érjük el. Itt helyezzünk el egy riport tartományt, amelyben legyen egy, a `dolgozó` táblát megjelenítő riport, fölötte pedig egy legördülő lista, mely a dolgozók munkaköreit tartalmazza, és egy olyan gomb, amelynek hatására a riportban a kiválasztott munkakörnek megfelelő doplgozók listája jelenjen meg. A legördülő listában legyen egy olyan elem, melynek választásakor az összes dolgozó jelenik meg.
A korábbiakban létrehozott alkalmazás egy lapot tartalmazott és azon a 3.1. ábra szerinti három riportot.



3.1. ábra. *Alkalmazásunk első lapja*

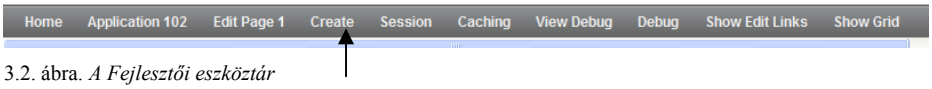
A feladat megoldásának érdekében először létrehozunk egy második lapot, majd egy gombot az első lap közepén, mely elérhetővé teszi a második lapot, utána pedig egy gombot a második lapon, melynek segítségével vissza lehet jutni az első lapra.

Interaktív riport létrehozása.

1. Először tehát hozzuk létre a második lapot, amelyen legyen `dolgozo` tábla interaktív riportja. A felhasználónak megjelenéskor mindig más és más lekérdezésekre van szüksége.

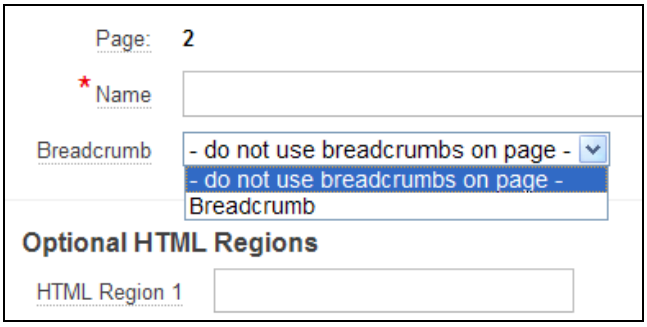
Lépései:

- A felhasználói eszközsoron kattintsunk a `Create` menüpontra (3.2. ábra):.



3.2. ábra. *A Fejlesztői eszköztár*

- A megjelenő ablakban válasszuk az Új lap (`New Page`) rádiógombot, majd `Next`.
- A következő ablakban jelöljük meg a `HTML` lap rádiógombot és `Next`
- Itt kötelező kitölteni (vagy meghagyni a felajánlottat), vagy megadni a Lap számát, `Next`.
- Az újabb ablakban kötelező megadni a Lap nevét (`Name`). A `Breadcrumb` mezőben választhatunk, hogy legyen e visszalépő sáv, vagy nem a Lapon. A Breadcrumb is törölhető. (3.3.ábra)



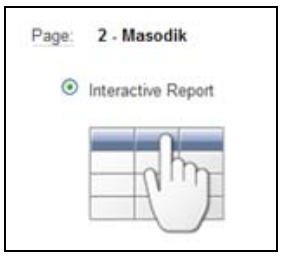
3.3. ábra *A Lap nevének megadása*

- Az újonnan megjelenő ablakban a Fület (`Tab`) állíthatjuk be, most még ne legyen Fül (`Do not use tabs`), és `Next`.
- Ellenőrizzük az összesítést (`Confirm`), majd `Finish`.

Kattintsunk létrejötte után a Lap szerkesztésére gombra.

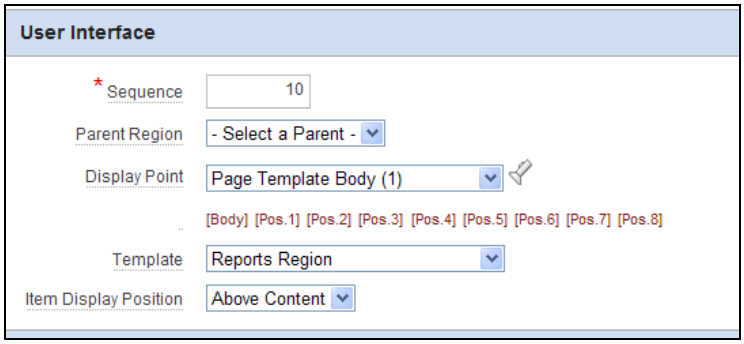
Hozzunk létre egy új tartományt az interaktív riport számára.

- Válasszuk a `Report` tartományt, `Next`
- Válasszuk megint a `Interactive Report` rádiógombot (3.4. ábra), `Next`



3.4. ábra Interaktív Riport választás

- Adjuk meg a riport nevét, válasszuk ki a `Report Template`-t, és a `Riport` belépési pontját (3.5. ábra), majd `Next`.



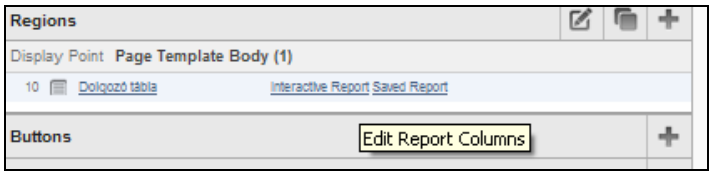
3.5. ábra *A riport felhasználói felületének beállítása*

- Adjuk meg a riport forrásást (`Source`), használjuk a Query Buildert. onnan `Return`-nel térünk vissza, feltételt még nem adunk, majd `Create Region`.

Futtassuk le, a riport megjelenik. Ha rossz belépési pontot választottunk, akkor nem mindig jól jelenik meg a riport.

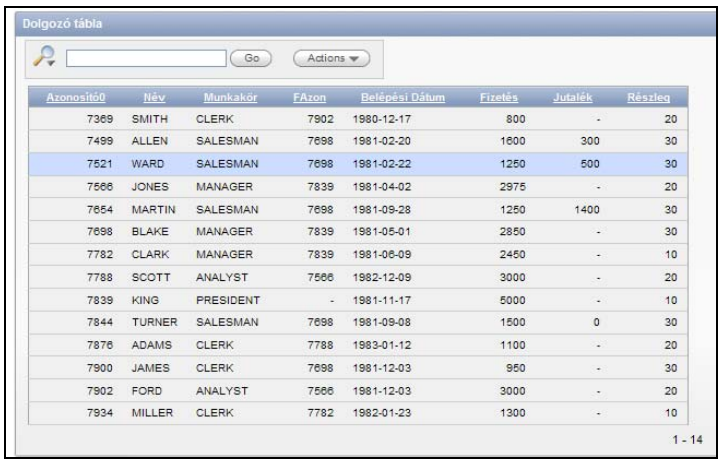
Megjegyezzük: hogy a Position 2.-belépési pontot akkor használjuk, ha több riportot szeretnénk egymás mellett megjeleníteni

Változtassuk a fejléceket magyarra (3.6. ábra).
`Edit Page 2`, majd `Edit Report Columns`



3.6. ábra *A riport oszlopainak formázása*

Írjuk át az oszlopneveket magyarra, majd futtassuk le.



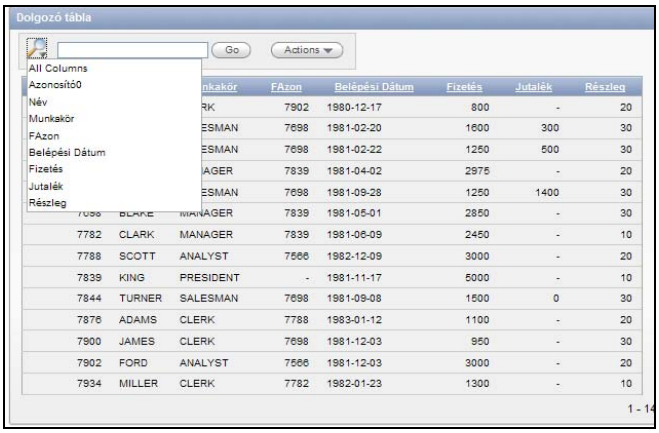
Azonosító0	Név	Munkakör	FAzon	Belépési Dátum	Fizetés	Jutalék	Részleg
7369	SMITH	CLERK	7902	1980-12-17	800	-	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02	2975	-	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850	-	30
7782	CLARK	MANAGER	7839	1981-06-09	2450	-	10
7788	SCOTT	ANALYST	7566	1982-12-09	3000	-	20
7839	KING	PRESIDENT	-	1981-11-17	5000	-	10
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1983-01-12	1100	-	20
7900	JAMES	CLERK	7698	1981-12-03	950	-	30
7902	FORD	ANALYST	7566	1981-12-03	3000	-	20
7934	MILLER	CLERK	7782	1982-01-23	1300	-	10

3.7.ábra *Az interaktív riport futtatása*

Figyeljük meg, hogy az interaktív riportban az oszlopnevek felett található egy kereső mező egy `Go` gomb, amely a keresést, valamint a tevékenységeket végrehajtja, és egy `Action` gomb.

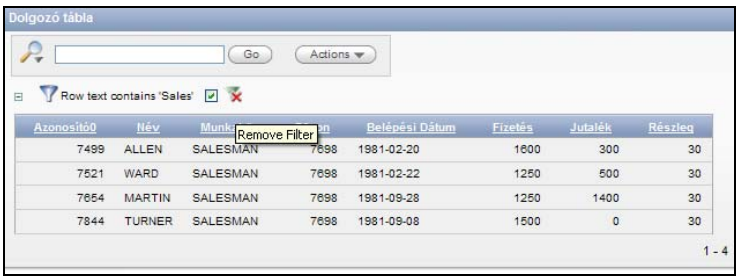
Milyen Tevékenység végezhető el az `Action` gombbal.

- Keresés- nagyítóval történő keresés:
  A nagyító melletti kis háromszögre kattintva oszlopokban keres (3.8. ábra)



3.8.ábra *Keresés az interaktív riportban*

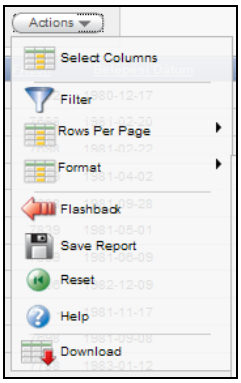
A különböző oszlopokban tud keresni és a megtaláltak minden adatát listázza.
Amikor beírjuk a kereső mezőbe azt, hogy `sales,` akkor megjelenik az összes kereskedő foglalkozású dolgozó.



3.9.ábra *Keresési beállítások illetve megszüntethetősége*

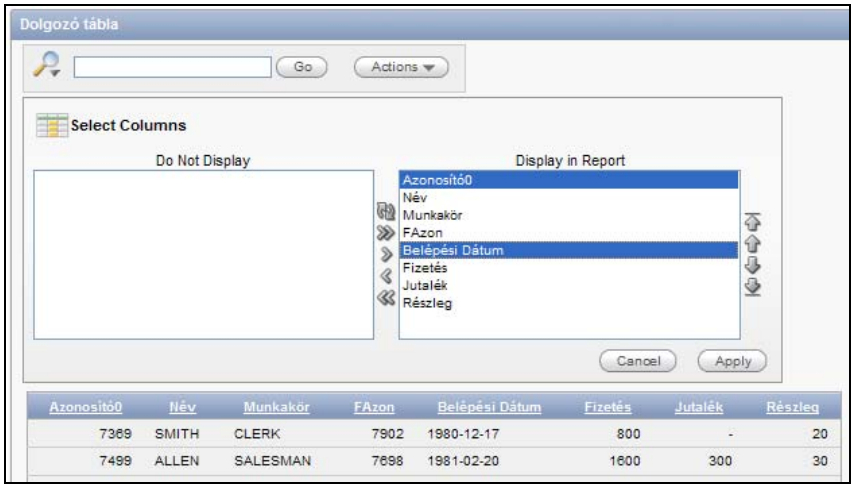
A szűrőt megelehet szüntetni a piros x -el (3.9. ábra). A keresés szövegre és számokra is használható.

- Oszlop Kiválasztása



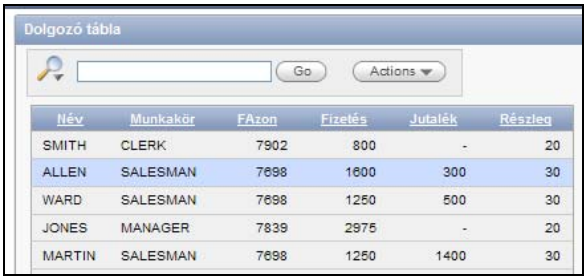
3.10.ábra A tevékenységek az interaktív riportban

Kattintsunk az oszlop kiválasztásra. Kiválaszthatjuk, hogy mely oszlopokat szertnénk megjeleníteni, illetve nem megjeleníteni (3.11. ábra)



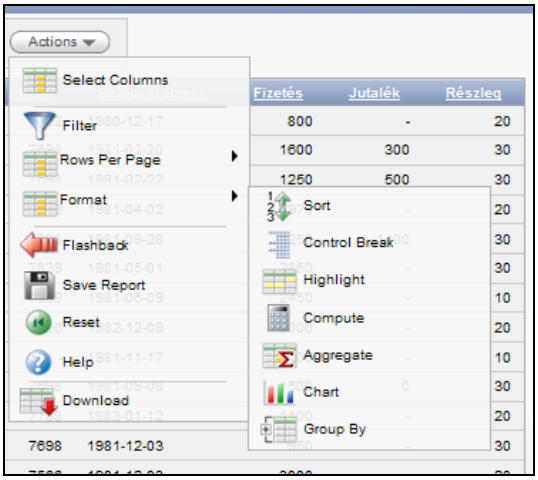
3.11.ábra *A megjenenítendő oszlopok kiválasztása*

Használjuk a balra nyilat. majd `Apply`, és a listában a kiválasztott oszlopok nem jelennek meg (3.12. ábra).



3.12.ábra *Sikeres riport*

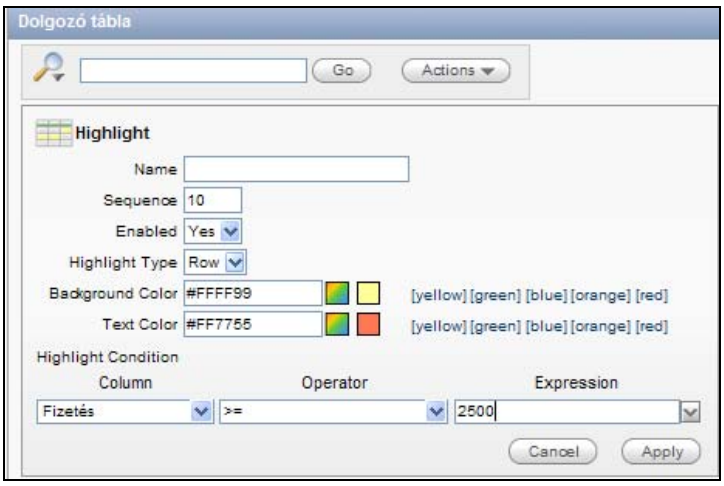
- Beállíthatjuk a megjelenő sorok számát (`Rows Per Page`)
- Beállíthatjuk a riport formátumát (3.13. ábra)



3.13. ábra *Formátum beállítás*

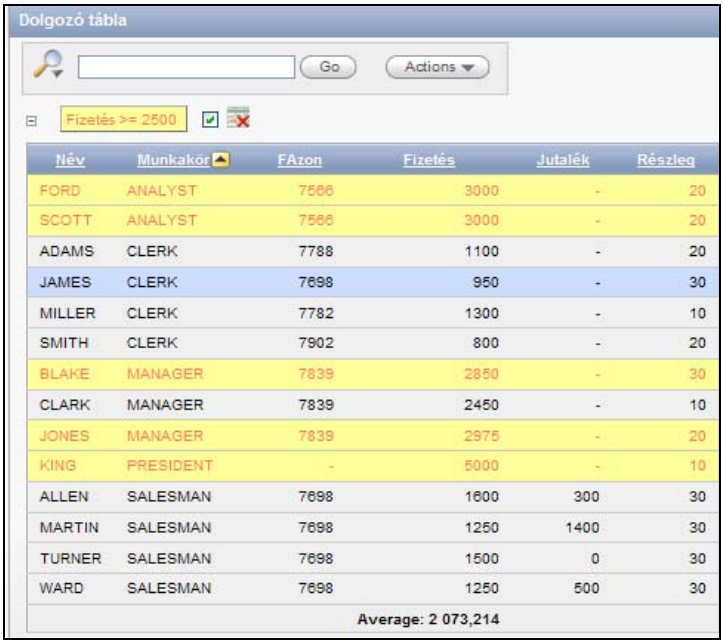
- Rendezés a kiválasztott oszlop pl. munkakakör szerint.
- `Control Break` Töréspontot hoz létre a választott oszlop szerint. Ez kivonja a megjelenő riportot az interaktív riportból és mint master rekordot jeleníti meg. lásd majd alább.
- Kiemel (`HighLight`)

  Emeljük ki azokat akiknek nagyobb vagy egyenlő a fizetésük 2000 nél, a háttér legyen sárga,és a betűszín piros (3.14. ábra)



3.14.ábra *Kiemelés és szín beállítása*

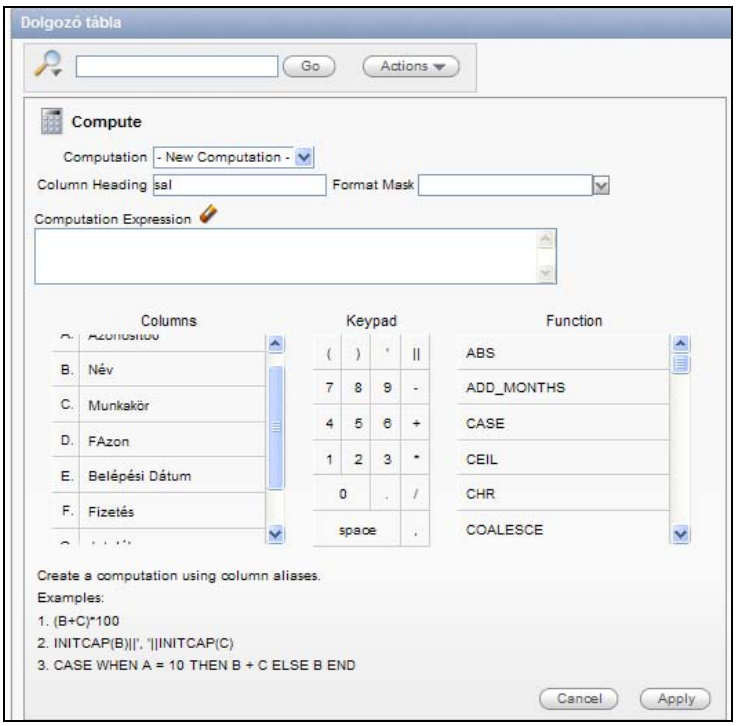
Apply



Név	Munkakör	FAzon	Fizetés	Jutalék	Részleg
FORD	ANALYST	7566	3000	-	20
SCOTT	ANALYST	7566	3000	-	20
ADAMS	CLERK	7788	1100	-	20
JAMES	CLERK	7698	950	-	30
MILLER	CLERK	7782	1300	-	10
SMITH	CLERK	7902	800	-	20
BLAKE	MANAGER	7839	2850	-	30
CLARK	MANAGER	7839	2450	-	10
JONES	MANAGER	7839	2975	-	20
KING	PRESIDENT	-	5000	-	10
ALLEN	SALESMAN	7698	1600	300	30
MARTIN	SALESMAN	7698	1250	1400	30
TURNER	SALESMAN	7698	1500	0	30
WARD	SALESMAN	7698	1250	500	30

3.15. ábra *Háttér sárga, betűszín piros*

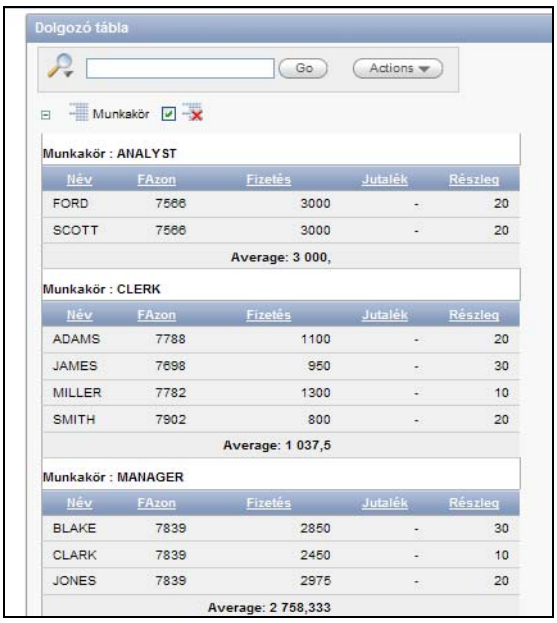
- Számítás (`Compute`): Ez a számítás matematikai számítás, vagy a létező oszlopokra alkalmazhatunk standard függvényeket (3.16. ábra).



3.16.ábra *Számítási lehetőségek*

- Függvény

Bállítottuk a `Control Break`-re a munkakört és ennek átlagfizetésére voltunk kiváncsiak (3.17. ábra)



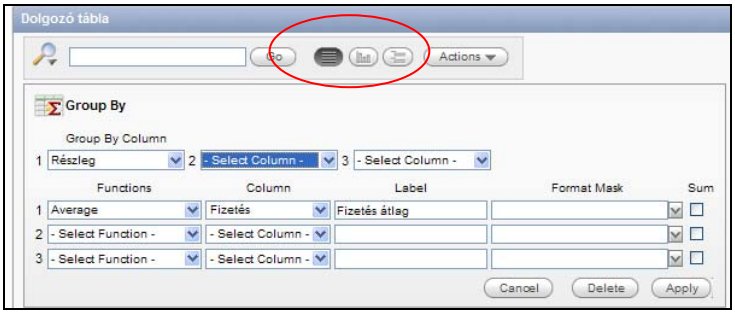
3.17.ábra *Átlag függvény*

- Grafikon (`Chart`)

3.18.ábra *Grafikon készítése az interaktív riporttal.*

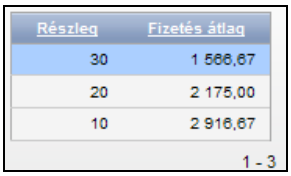
- csoportosítás (`Group By`)

A felső sor új ikonjai
`View Report`, `View Chart` valamint View `Group By`.



3.19. ábra *Csoportosítás*

Állítsuk be a formátum `Maskot` is, `Apply`

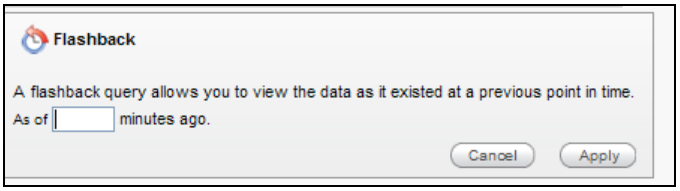


Részleg	Fizetés átlag
30	1 566,67
20	2 175,00
10	2 916,67

1 - 3

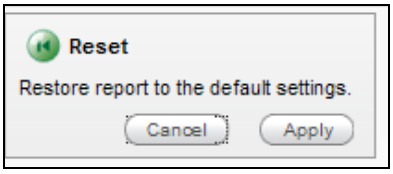
3.20.ábra *Az átlag Formátum maszkkal*

- Visszapillantás `(Flashback)`



3.21. ábra *Visszapillantás*

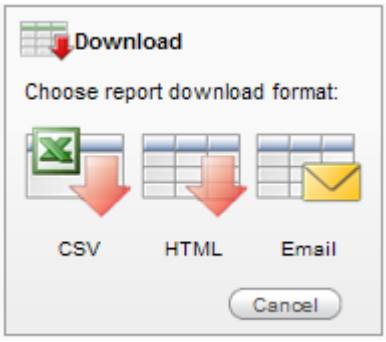
- Riport mentése (`Saved Report`)
- Alaphelyzetbe állítás (`Reset`)(3.22.ábra)



3.22.ábra *Visszaállítás*

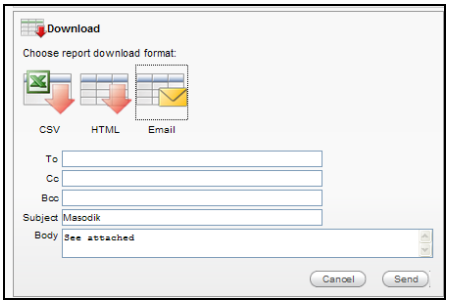
Az előző beállítást állítja vissza.

- Help
- Letöltés (`DownLoad`)



3.23.ábra *Letöltés*

Háttértárolóra tölti a riportot `csv` vagy `html` formátumban, illetve elküldi levélben (3.23. ábra.



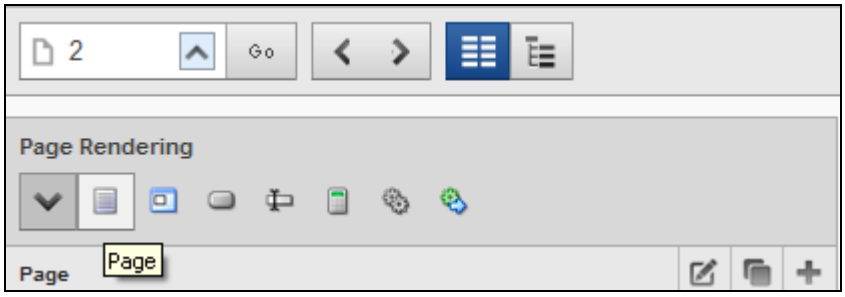
3.24.ábra Elküldés e-mailben

Lap törlése

Ha véletlenül valamelyik lapot elrontottuk akkor az első lap kivételével a következő módon törölhetjük. (Az első lapot ne töröljük símán, mert a Login lap az első lapra van beállítva, és akkor az alkalmazásunk nem indul el.

Ha például a második lapot szeretnén törölni, akkor menjünk az `Edit Page 2`-re

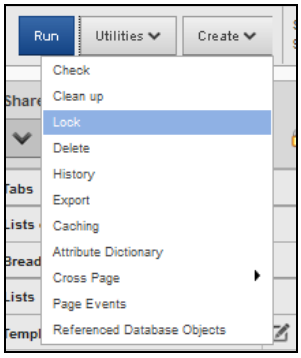
Itt a `Page Rendering` soron kattintsunk az alábbi `Page` ikonra (3.25. ábra).



3.25.ábra *Lap szekció*

Ekkor csak a Lap rész jelenik meg.

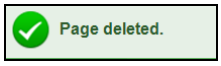
Kattintsunk a `Run` gomb mellett a `Utilities` gombra, és itt válasszuk a `Delete` menüpontot.(3.27. ábra)



3.26. ábra A Lap törlése funkció

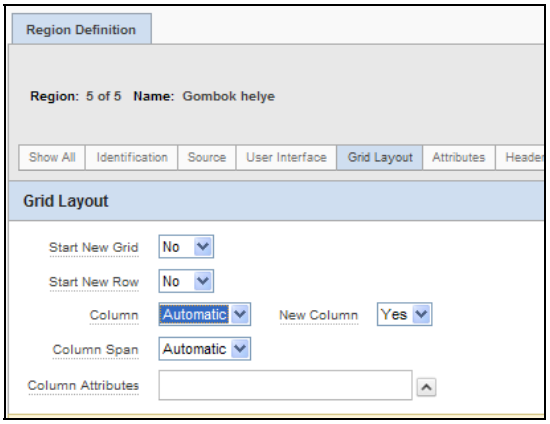
`Next`,

Kattintsunk a törlés gombra majd a következő ablakban erősítsük meg a törlést, azaz kattintsunk a

gombra. A lapot kitöröltük. 

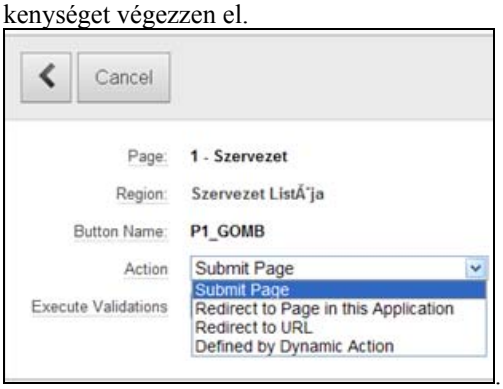
Navigáló gomb létrehozása

2. Ezekután menjünk vissza az első lapra és állítsuk be a létrehozandó gomb paramétereit a `Page Definition` ablakban.
   Hozzunk létre e lapon egy új tartományt (`region`). Legyen a típusa `HTML`, a neve pedig `Gombok helye` (3.27. ábra). legyen ez középen a `telephelyi átlag` riport alatt. Ezért a `Grid Layout` ablakban a `Column (oszlop)` mező értékét állítsuk `2`-re. Amikor az oszlopszámot beírjuk az automatic helyére , a new column mező eltűnik.



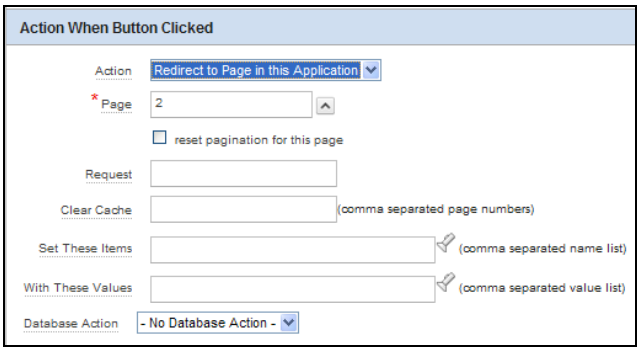
3.27. ábra. *A megjelenítési oszlop beállítása*

3. Hozzunk létre egy nem-Elem típusú gombot a `Gombok` tartományban.
   A gomb legyen `HTML Button` stílusú, és válasszunk `Create a button in a region position` pozíciójú (tehát nem-Elem típusú) gombot, majd a következő lapon adjuk meg a nevét, (Name) cimkéjét (Label), a stílusát (Button Style), esetleg a gomb mintáját (Button Template) és típusát Button Type). Next, Next.
   A megjelenő Action When Clicked lapon beállítjuk, hogy a gomb milyen típusú tevékenységet végezzen el.



3.28.ábra

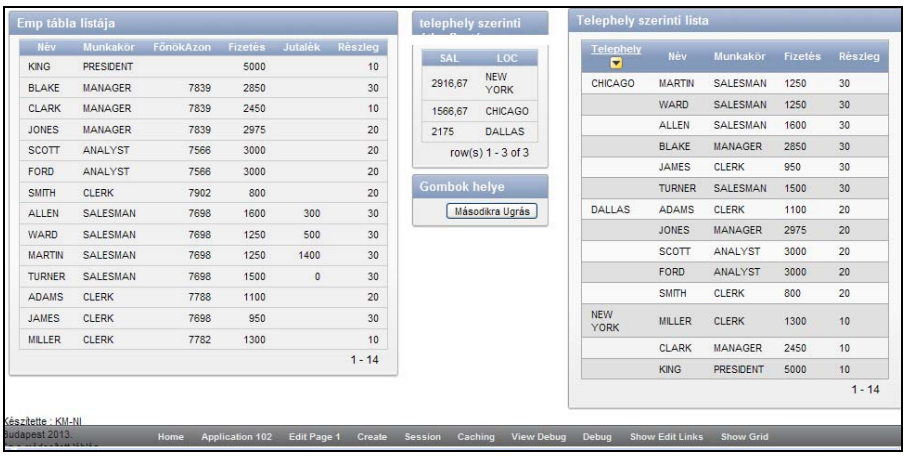
   Állítsuk be a `Redirect to Page in this Application` típusú tevékenységet (azaz legyen a ugorjon vissza az alkalmazás valamelyik lapjára.). Végezzen-e ellenőrzést vagy sem. `Next.` Ekkor még több kitöltendő szövegmező jelenik meg.
   -Melyik lapra ugorjon (`Page`)
   - Azonosítót megadhatunk (`Request`) megadható
   -Törölhető a lap (`Clear Cache`)
   - Beállítható valamilyen elem és valamilyen értékkel
   `Create Button`) a 3.29. ábra szerint.



3.29.ábra. *Lapra ugrás*

`Next`, beállítható feltétel a gomb megjelenéséhez, majd hozzuk létre a gombot. (`Create Button`)

A gomb valóban középre került a Telephelyi átlagfizetés riportja alá (3.30. ábra).



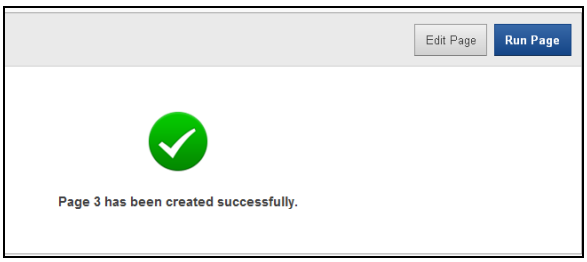
3.30. ábra. *Futási eredmény*

4. Menjünk át a második lapra és a korábbiak alapján helyezzünk el egy `Vissza` feliratú gombot a "dolgozo tábla" riport tartományában, melynek segítségével az előző lap lesz ismét aktív. A létrehozáshoz segítségként megjegyezzük, hogy e gomb nem-Elem és Átirányító típusú. Az elkészült lap a 3.31. ábrán látható:

3.31. ábra. *A második lap*

Fül utólagos létrehozása és megjelenítése

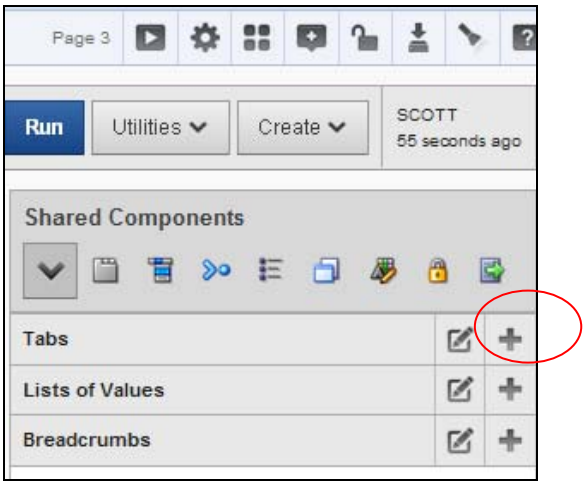
Az eddigiekben létrehozott alkalmazás lapjait lássuk el a közvetlen lapra navigálást elősegítő eszközzel (Navigációs Objektummal), vagyis Füllel (`Tab`). Nem mindig cél az összes lap közvetlen elérésének biztosítása. Az alábbiakban létrehozzuk az alkalmazás harmadik lapját, és Fület csak erre és az első lapra fogunk tenni. Ekkor a második lap változatlanul csak az első lapról lesz elérhető egy gomb segítségével (aminek például funkcionális, vagy biztonsági okai lehetnek).
Hozzunk létre tehát egy harmadik lapot, mely Üres (`Blank`) típusú. Futtassuk le az alkalmazásunkat.



3.32.ábra *Sikeres lap létrehozás*

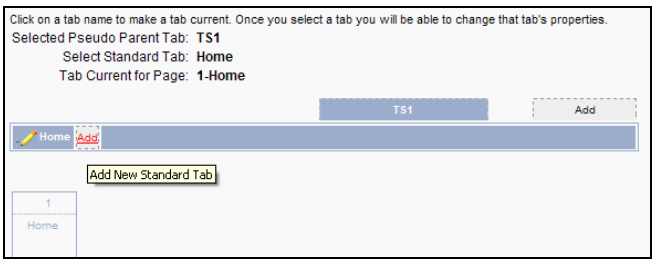
Látható, hogy futtatáskor a harmadik lap egyenlőre nem elérhető. Az elérhetősége érdekében tehát tegyünk rá egy Fület. Az alábbiakban ennek lépéseit mutatjuk be.

1. Lépjünk be a lap szerkesztésbe (`Edit Page 3`), majd kattintsunk a Megosztott Komponensek (`Shared Components`) rész Fül-Szerkesztés (`Tabs`) szekciójában a `Létrehozás` (plusz jelü) ikonra (3.33. ábra.)



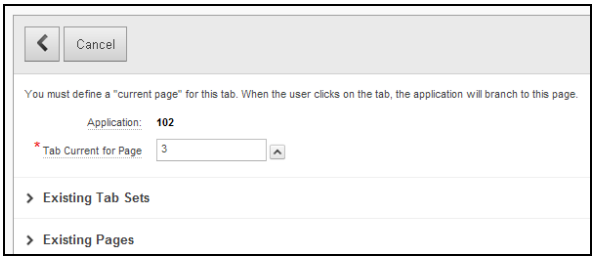
3.33. ábra. *Fül létrehozása*

A Fül-szerkesztő ablak az alkalmazás Füleinek grafikus reprezentációját mutatja. Az Alapértelmezett Fül Készlet (`Standard Tab Set` = `TS1`) arra használható, hogy a felhasználókat meghatározott laphoz kapcsolja. A Szülő Fül az alkalmazáson belül a navigáció egy másik szintjét adja a felhasználó számára. A felső sorban elhelyezkedő `Add` gombra kattintva szülő Fül Készletet hozunk létre, az alsó sorban levő `Add` gombra kattintva új Fület hozunk létre az `TS1`-ben. (3.34. ábra)



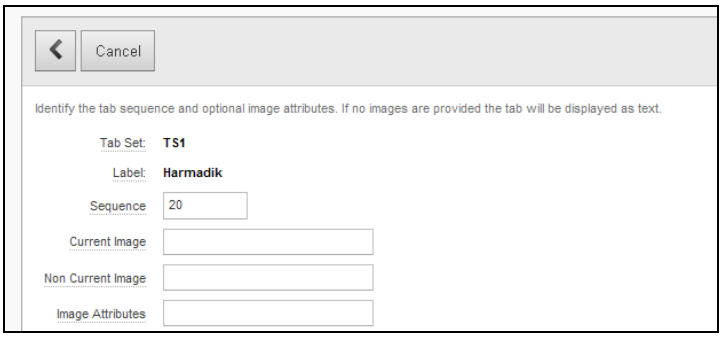
3.34.ábra Fül, fülkészlet létrehozása

2. Hozzunk létre (`Create`) egy új Fület a meglévő Standard Tab készletben a `TS1`-ben, kattintsunk az alsó sorban az `Add` gombra, megadjuk a Fül cimkéjét (vagyis a megjelenő nevét). Legyen ez a név `Szűrt Lista`., majd lépjünk tovább (`Next`) **.** Ezután a `Tab Current for Page` mezőben adjuk meg (válasszuk ki), hogy melyik lapra kerüljön a létrehozandó Fül (3.35. ábra), és lépjünk megint tovább.



3.35. ábra. *A fül aktuális lapja*

3. A következő lépés a megjelenő sorrend (`Sequence`) és kép (`Image`) megadása. Mivel ez lesz a második Fül fogadjuk el a felajánlott 20-as sorrendet. Képünk most nincs, lépjünk tovább (`Next`). (3.36. ábra)



3.36.ábra Fül sorrend beállítása

4. Megadhatjuk a Fül megjelenítési feltételeit (például létezik-e már az a tábla amelyet megjeleníteni, illetve szűrni szeretnénk ezen a lapon, (3.37. ábra) stb.).

3.37. ábra. *A fül megjelenítési feltétele*

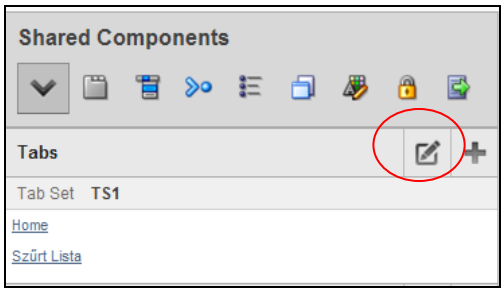
A Fül létezésének forrása :

```
Select table_name FROM user_tables
  WHERE UPPER(table_name) = UPPER ('Dolgozo')
```

5. A feltételek beállítása után (`Next`) megjelenő összesítő ablakban megerősítjük, létrehozzuk a Fület (`Create Tab`). Ekkor tehát a Fül Készlet és a harmadik oldali Fül létrejött. Nézzük meg a Lap-Definíciós (`Page Definition`) ablakban a Megosztott Komponensek (`Shared Components`) rész Fül-Szerkesztés (`Tabs`) szekciót. Megjelent a létrehozott Fül Készlet (`Oldalak`) és az első valamint a harmadik oldali Fül (`Szűrt Lista`) neve (3.38.ábra)

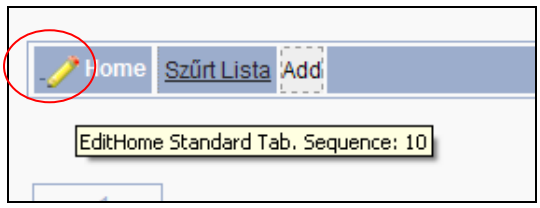
3.38. ábra. *A létrejött fülkészlet és fül*

6 .Lefuttatva megjelenik a létrehozott és a Standard fül is, amelynek változtassuk meg a nevét.Kattintsunk a `Edit Page 3` menüpontra a Fejlesztői Eszköztáron. A Harmadik oldal szerkesztői lapjánkattintsunk a Tabs szerkesztésére.



3.39.ábra Fül szerkesztése

Kattintsunk a megjelenő ablakban a `Home Linkre`, majd a szerkesztésre .



3.29. ábra. *Fül szerkesztése*

A neveket felajánlja, de megváltozutathatjuk. A fül cimkéje (`Tab Label`) legyen Riportok, majd `Apply Changes`.

7. A Fül-sorrend beállítása, ha szükséges.
   A fülek megjelenítési sorrendjének (`sequence`) beállításához a Lap-Definíciós (`Page Definition`) ablak Megosztott Komponensek (`Shared Components`) részének Fül-Szerkesztés (`Tabs`) szekciójában a kiválasztott Fül paramétereit be kell állítani a Fül nevének hiperhivatkozására való kattintással. Az így megjelenő Fül-Szerkesztő (`Edit Standard Tab`) ablakban `Name` szekciójában a `Sequence` paramétert kell beállítani úgy, hogy a helyes sorrendet kapjuk. Ehhez természetesen előbb mindkét Fülre vonatkozó értéket meg kell tekintenünk.

   Láthatjuk, hogy mindkét Fül megjelent, és helyes sorrendben. Ekkor az alkalmazásunk tehát a következőképpen működik: Az első lapról elérjük a `Másodikra Ugrás` Gomb segítségével a második lapot, a `Szűrt Lista` Fül segítségével pedig a harmadik lapot. A második lapról a `Vissza` Gomb segítségével érjük el az első lapot, a harmadik lapról pedig a `Riportok` Fül segítségével az első lapot. A kapcsolat így teljes a létrehozott három lap között.

8. Végül a jelenleg még üres harmadik lapon jelenítsük meg, a feladatkiírás értelmében a felhasználó által megadott (vagy kiválasztott) munkakör szerinti szűrt listát. Ez tehát egy futás közben definiált, azaz úgynevezett *Dinamikus Lista* lesz. Elkészítését az alábbiakban mutatjuk be. Előbb azonban ismerkedjünk meg az elemekkel (`Items`)

4. fejezet. Elemek (Items)

Az APEX-ben az Elemek (`Items`) a felhasználó és az alkalmazás közötti információátadásra szolgáló kommunikációs (adatbeviteli, adatmegjelenítő) eszközök. Ezek a (html) Lapok Tartományaiban helyezkednek el. Egyebek mellett ezek szolgálnak a felhasználó azonosításra, az alkalmazás munkafolyamataihoz szükséges adatok bevitelére, listák megjelenítésére, stb. Ennek megfelelően lehetnek `text` mezők, `jelszó` mezők, `combobox`, `checkbox` és így tovább.

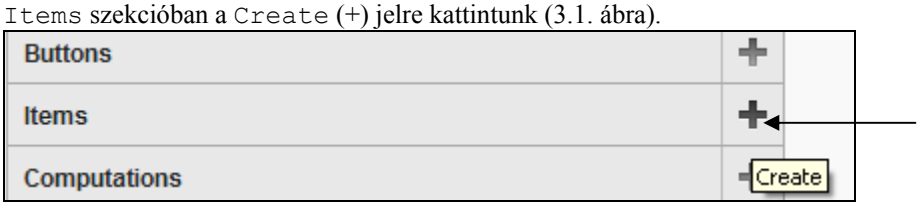
Az Elemeknek két típusa van:

- `Page items`: Lap-Elemek. Ezek a Lapokon találhatók, és valamilyen adat beírására, kiválasztására, megjelenítésére szolgálnak. A továbbiakban ezekkel foglalkozunk.
- `Application items`: Alkalmazás-Elemek. Ezek nincsenek laphoz társítva, az alkalmazás egészére vonatkozó tulajdonságok beállítására szolgáló eszközök, a felhasználói felületen nem láthatóak. Úgy használjuk azokat, mint globális változók.

Elemek típusai

Az alábbiakban áttekintjük a Lap-Elemek (továbbiakban egyszerűen csak Elemek) főbb tipusait, és használatukat.

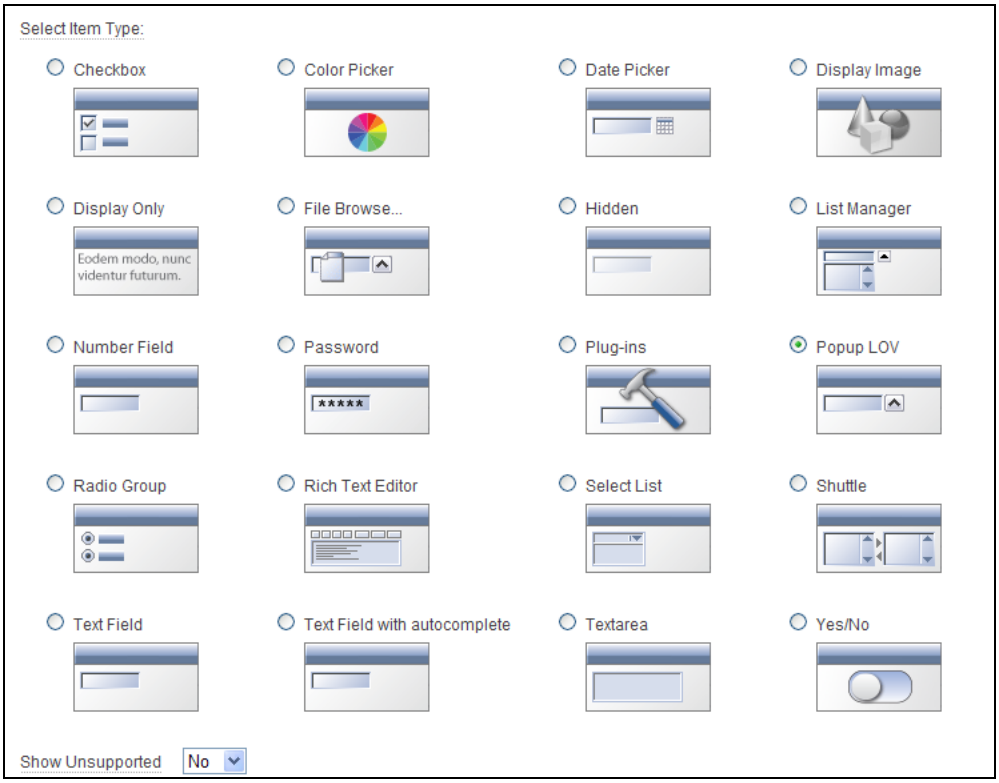
Egy Elem létrehozását oly módon kezdeményezhetjük, hogy a létrehozandó alkalmazás általunk kiválasztott lapjának Lap-Definíciós (`Page Definition`) ablakában oldalán az `Items` szekcióban a `Create` (+) jelre kattintunk (3.1. ábra).



4.1. ábra. *Az Elemek szekció*

Megjegyezzük, hogy az Elemek a használatuk és az alapvető tulajdonságaik szerint lehetnek úgynevezett `Combobox` és `nem-Combobox` típusúak. A `Combobox` típusú elemek esetén valamilyen előre megadott listából választhatunk ki egy vagy több megfelelő értéket, míg a nem Combobox típusúak egyszerű szövegbevitelre, illetve -megjelenítésre szolgálnak. A `Combobox` típusú elemek például a `Check Box`, a `List Manager`, a `Popup List`, a `Radio`, a `Select List`, míg `nem-Combobox` típusuak a `Text`, a `TextArea`, a `Password`, a `File Browse`, stb.

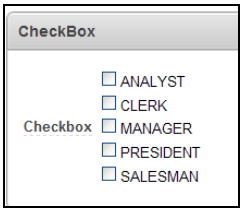
Válasszuk ki a megfelelő Elemet a megjelenő ablakban. A 4.2. ábrán az Application Express 4.2.1 változatában szereplő Elemeket láthatjuk.



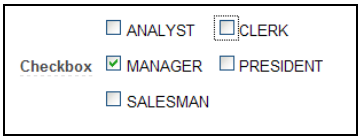
4.2. ábra. *Az Elemek típusai*

Az Elemek a következők:

- Kijelölő elem `Check Box`
  Listából való kiválasztást tesz lehetővé. A visszaadott érték egy karaktersorozat, melyben többszörös kijelölés esetén a kiválasztott értékek kettősponttal választódnak el.
  Többszörös kijelölés feldolgozása például a következőképpen történhet.
  Listázzuk a Dolgozo tábla kiválasztott munkakörű dolgozóit, és legyen a `Check Box` Elem neve `P1_VALASZT`. Ekkor a 4.3. ábra szerinti alkalmazásban a Mehet gombra való kattintással megjelennek a szükséges sorok. A kiválasztó utasítás az alábbi (ahol a paraméteres hivatkozásban az Elem neve előtt kettőspont áll):



vagy



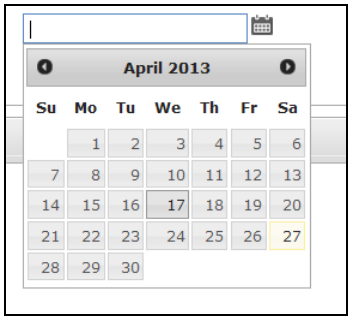
4.3. ábra Check Box Kiválasztó elem

A CheckBox által kijelölt munkakörök alkalmazottait példáiul a következő lekérdezés jeleníti meg.

```
select ALKALMAZOTT.ENAME as ENAME,
         ALKALMAZOTT.JOB as JOB,
         ALKALMAZOTT.SAL as SAL,
         ALKALMAZOTT.DEPTNO as DEPTNO
   from ALKALMAZOTT ALKALMAZOTT
    WHERE INSTR(:P1_VALASZT, "ALKALMAZOTT"."JOB") > 0
```

- `Data Picker` vagy `Data Picker`(Classic) (DátumVálasztó)

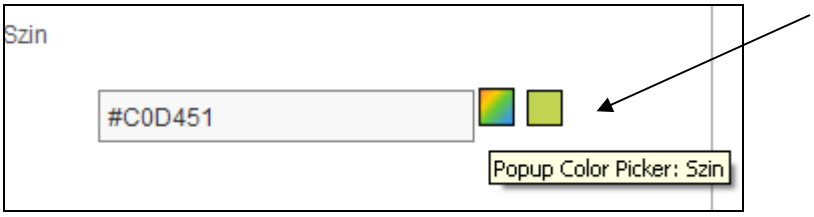
  Használata esetén egy naptár ikon jelenik meg a text mező mellett ezzel segítve a dátumbeírást. Az ikonra kattintva egy naptárt tartalmazó ablak ugrik elő, melynek a megfelelő napjára kattintva az adott formátumnak megfelelően kerül a dátum a szövegmezőbe(4.4.ábra)



4.4.ábra. DátumVálasztó

- `Color Picker (Szín választó)`

  Hasonlóan működik, mint bármelyik `popup` kiválasztás (4.5. ábra). A jobboldali kis körre kell kattintani, és akkor megjelenik a kiválasztott szín a második négyzetben.



4.5. ábra *Szin kiválasztás*

- `Display Image` (Kép megjelenítő)

  URL-ben vagy másik lapon, vagy BLOB típusú oszlopban tárolt képek megjelenítését teszi lehetővé.

- `Display Only (Csak Megjelenítő)`

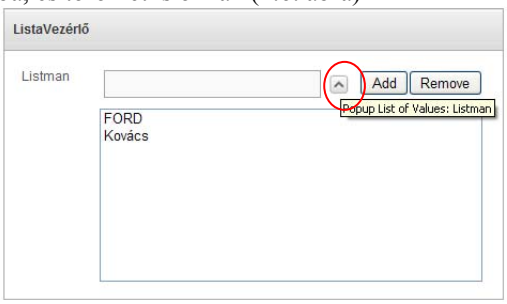
  Ez egy olyan szövegmező, mely csak olvasható megjelenítést tesz lehetővé.

- `File Browse (Fájl kereső)`

  Ez a szövegmező egy `Tallózás...` feliratú gombbal együtt jelenik meg. Segít-

ségével a felhasználó megkeresheti a fájlt a lokális fájl rendszerében, és betöltheti azt.

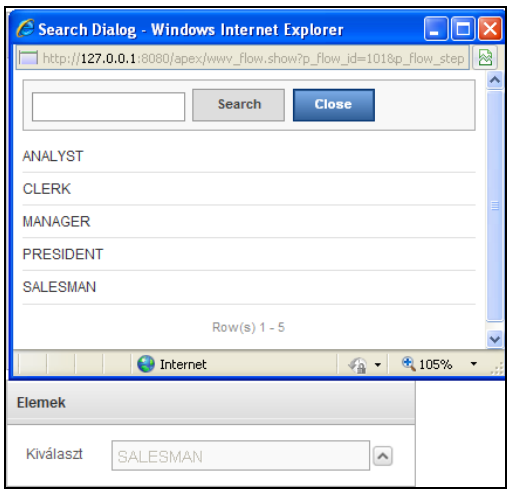
- `Hidden (rejtett)`
  Ez a szövegmező nem jelenik meg. Tulajdonképpen szövegváltozóként használható például adattárolás, vagy a lapok közötti adatcsere céljaira.
- `List Manager (ListaVezérlő)`
  Használatával a felhasználó egy segédmező segítségével bevihet adatot a listába, és törölhet is onnan (4.6. ábra)



4.6. ábra. *ListaVezérlő megjelenése*

A kis (piros karikával jelölt) ikonra kattintva egy felugró (`popup`) ablakban megjelenik a forrásként megadott SQL lekérdezés, vagy statikusan megadott értékek listája.
Az ebben kiválasztott elem bekerül a vízszintes szövegmezőbe, majd az `Add` gomra kattintva a függőleges lista (`Listavezérlő`) elemei közé. Felvehetünk azonban bármilyen más listaelemet is (pl. Kovács), amelyet mi írtunk be a vízszintes mezőbe, és szintén az `Add` gombra bekerül a függőleges kiválasztott elemei közé. A példa azt mutatja, hogy a popup ablakból kiválasztottuk a FORD nevű dolgozót, és beírtuk a Kovács nevű dolgozót is.

- `Number Field (Szám mező)`
  Egy szám mezőt jelenít meg. Beállítható a lehetséges szám tartomány a legkisebb (`MIN`) és legnagyobb (`MAX`) érték beállításával. Ez az Elem automatikusan vizsgálja, hogy a bevitt érték szám-e, és benne van-e az adott tartományban. Más különösebb ellenőrzés ekkor nem szükséges.
- `Password (Jelszó)`
  Ennek a szövegmező elemnek a segítségével rejtett módon lehet szöveget bevinni. A beírt karakterek helyett egy előre megadott karakter (például *) jelenik meg.
- `Popup List of Values (Felugró ablakból kiválasztás)`
  E szövegmező mellett egy felefelé mutató nyíl (az ábrán bekarikázott) jelenik meg. Erre kattintva a felugró (popup) ablakban a felhasználó kiválaszthatja a megfelelő értéket.

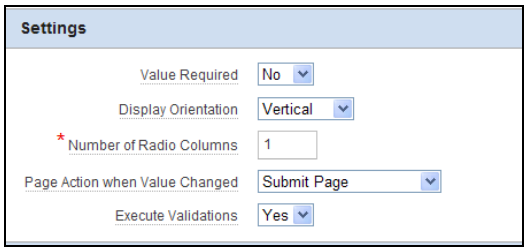


4.7. ábra. *Felugró ablak*

A fenti példában a felugró listába (4.7.ábra) a lehetséges foglalkozások nevei kerültek be.A kijelölt bekerül a szövegmezőbe, csak olvasható módon.

- `Radio (Rádiógombok)`

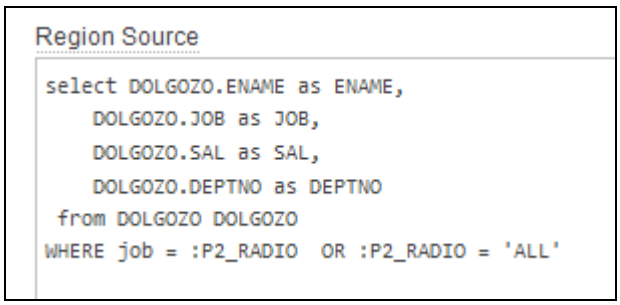
  Egy olyan listát jeleníthetünk meg segítségével, melynek elemei közül csak egy választható ki, de egy mindenképpen (alternatív elemek listája)(4.8. ábra).



4.8. ábra ádiógombok beállítása

A `Submit Page` választással a kijelölt elemre való kattintás hatására a kijelölt művelet azonnal végrehajtásra kerül

(Listázzuk a rádiógomb által kiválasztott munkakör alkalmazottait.)
A riport forrása (4.9. ábra) :



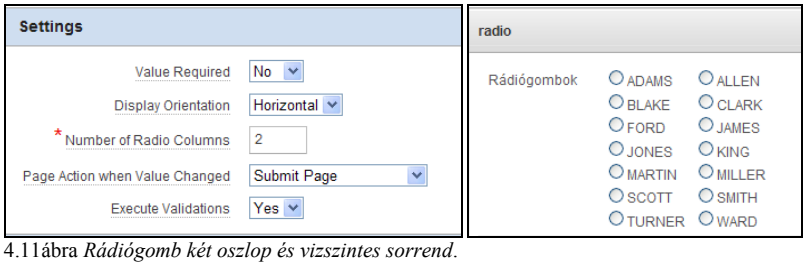
4.9. ábra Rádiógomb beállítása, ha a a null érték a LOV-ban 'ALL'

ahogy azt alább a List of Values-nél beállítottuk (4.10. ábra).

Megjelenése

4.10. ábra. *(Rádiógombok)*

A fenti példában a `CLERK` kijelölésének hatására azonnal megjelenik a riport. Amennyiben túl sok a megjelenítendő kiválasztható elemek száma, akkor használhatunk több oszlopot is (4.11. ábra).

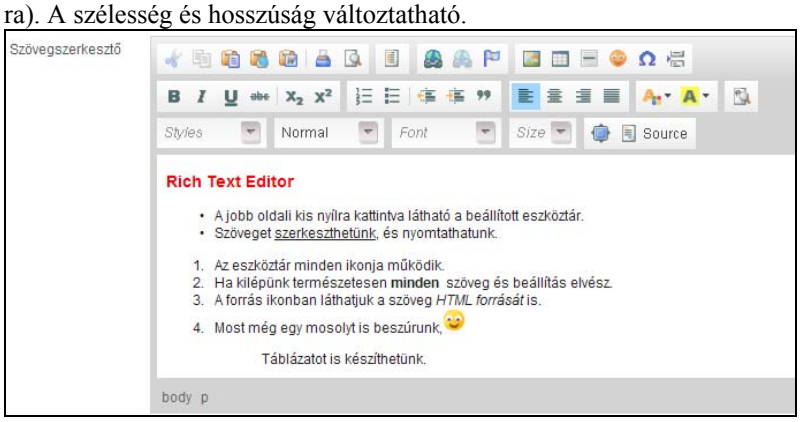


4.11ábra *Rádiógomb két oszlop és vizszintes sorrend.*

- `Rich Text Editor` (Szövegszerkesztő)

  Egy HTML területet jelenít meg. Nincs megadható maximális mérete (4.12. áb-

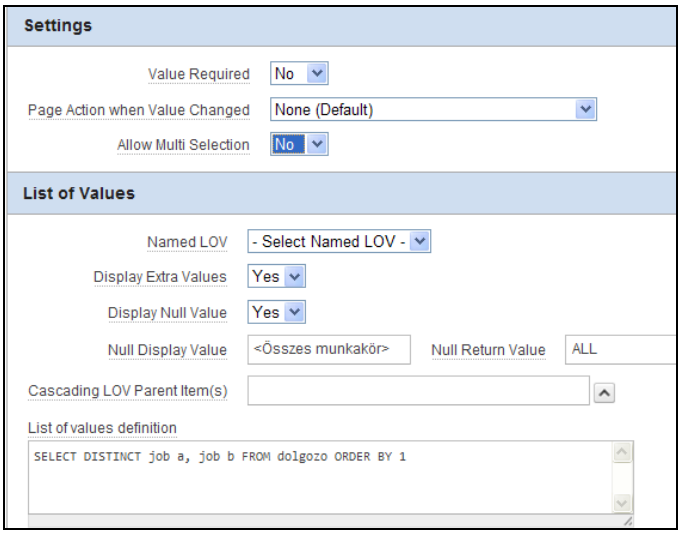
ra). A szélesség és hosszúság változtatható.



4.12.ábra A Szövegszerkesztő Elem

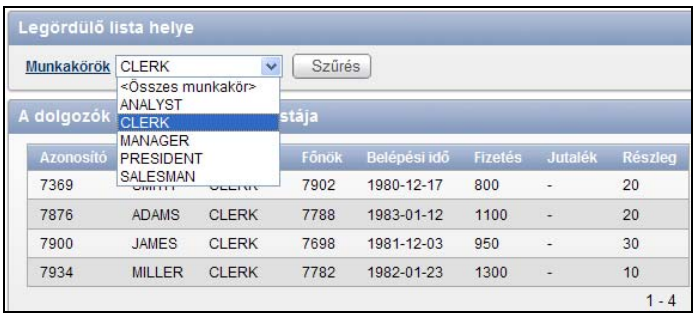
Kijelölés után kimásolható.

- `Select List` (KiválasztóLista)
  Egy legördülő listát jelenít meg.
  Beállításainál ügyeljünk a lap tevékenységére a megjelenő NULL értékekre, a visszatérő Null értékekre. Null értékre.(4.13. ábra)



4.13.ábra LOV beállítása

A fenti beállítások esetén a megjelenés:

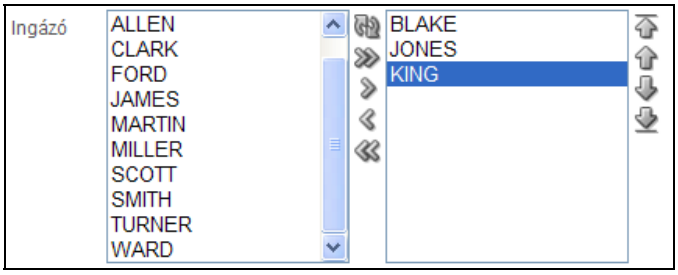


4.14. ábra. *KiválasztóLista*

A fenti példában a nyílra kattintva megjelennek a munkakörök, melyek közül a kiválasztott lesz a kimenő érték, és a `Szűrés` gombra való kattintás hatására aktíválható egy másik esemény ezzel a paraméterrel (4.14. ábra).
Ez az elem is használható azonnali végrehajtást kezdeményező módban (`Submit Page`) tevékenység esetén.

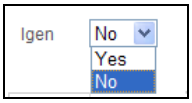
Az elem többszörös kiválasztást is megenged. Ezt a `Settings` ablak Allow `MultiSelection` menüpontban állíthajuk be amikor `Yes`-re állítjuk. Ebben az esetben a kiválasztott elemek sorozatát (:) kettősponttal választja el.

- `Shuttle` (Ingázó)
  Megjelenik az Elem, mint egy többszörös kiválasztású lista, amelyik 2 listát tartalmazó dobozból áll. A bal lista megjeleníti az értékek forrását. A felhasználó a nyilak segítségével mozgathatja a kijelölt elemeket balról jobbra, azaz a forrás dobozból a cél területre és vissza (4.15. ábra)



4.15.ábra Ingázó

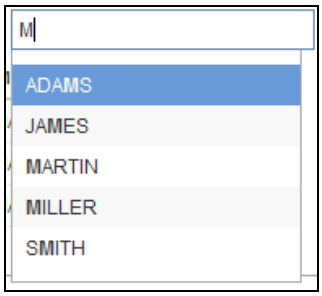
- `Yes/No` (Igen/Nem)
  Az elem megjelenik a felhasználói felületen, és mint egy kiválasztó elem (4.16. ábra).



4.16.ábra Kétállású kiválasztás

- `Text Field with autocomplete`
  Ez a szövegmező megjeleníti a lehetséges értékek listáját, amikor a felhasználó elkezd gépelni. Az értékekek egy `LOV` SQL utasítás alapján jelennek meg. Tehát a for-

rás pl:
```
SELECT ename FROM dolgozo ORDER BY 1
```
Ha leütjük ekkor az `M` betűt, akkor megjelenik a következő amiből kiválaszthatjuk a nekünk megfelelő paramétert (4.17.ábra).



4.17.ábra Automatikus segítés

- `Text Area (SzövegDoboz)`

  Megjelenít egy szövegdobozt a karakterek számának korlátozása nélkül. A szövegdoboznak nincs meghatározva hosszúsága. A hosszúság és szélesség a Height és Width mezőkben szerkeszthető. Megadható a maximális szélesség is.
- `Text (Szövegmező)`

  Megjelenít egy szövegmezőt, amely legfeljebb 30 000 karaktert tartalmazhat.
- `Stop and Start Table`

  Segítségével egy html tábla bezárása, vagy egy új megnyitása kezdeményezhető.

Hivatkozás Elemekre

Az Elemekre az alkalmazásban a következőképpen hivatkozhatunk (mely hivatkozás az Elem tartalmának beágyazását eredményezi):
SQL, PL/SQL esetén: a környezeti változóra való hivatkozás módján (`:Elemnév`),
Statikus szövegben: `&Elemnév.` (a végén ponttal!).

Új Elem létrehozása

Attól függően, hogy a fentiekben bemutatott első lépésben milyen Elemet választunk ki, más és más tulajdonságokat (attribútumokat) lehet beállítani.

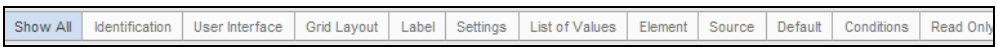
A rendszer által felajánlott `PN_X`, névben `N` az aktuális lap azonosítója (például a 2. lap esetén ez a `P2`, ezt célszerű elfogadni), az `X` helyébe pedig írjunk be valamilyen, az Elem funkcióját tükröző azonosító nevet.

A különböző típusú Elemek létrehozásának lépései hasonlóak. Nem minden tulajdonságot kell minden Elem-tipusnál beállítani, de vannak olyan adatok (például az elhelyezkedés tartománya, a megjelenítendő értékek forrása, stb.), melyeket minden esetben meg kell adni.

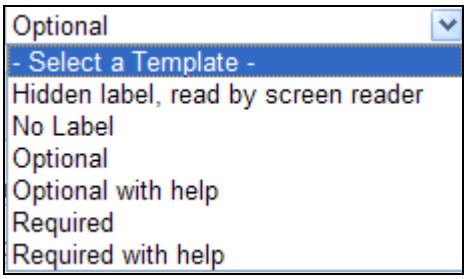
A kiválasztott Elem tulajdonságai utólag is szerkeszthetők. Az egyes tulajdonságok mind a létrehozás során, mind az utólagos beállításnál csoportosítva jelennek meg. Az alábbiakban az utólagos módosításnál (az `Edit Page Item` ablakban) szereplő fontosabb tulajdonság-csoportokat (4.18. ábra) mutatjuk be. Megjegyezzük, hogy ezek nem mindegyike értelmezhető minden Elemnél!

4.18. ábra. *Az Elemek tulajdonság csoportjai*

Az egyes tulajdonságcsoportok neve és általános leírása:



- `Show All` — Minden tulajdonság legyen látható
- `Identification` — Azonosítás
  - `Name` — a név és a megjelenés típusa
  - `Display as` — milyen típusú Elemként jelenjen meg
- `User Interface` — Felhasználói Felület
  - `Region` — Az elem melyik tartományban helyezkedjen el
  - `Temlate` — sablon választható (`Required`: csillagot tesz a cimkéhez, ezt kötelező beírni kérdés és adatbevitel esetén



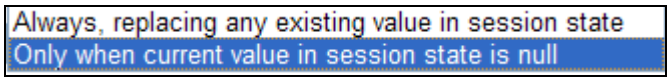
4.19.ábra *Sablon választás*

- `Grid Layout` — rács elrendezés
  - `Start New Grid` — új rácsonton kezdődjön-e vagy sem
  - `Start New Row` — új sorban kezdődjön
  - `Column` — a választott tartomány hányadik oszlopban legyen
  - `Column span` — az oszlopok távolsága
- `Label` — a címke neve, elhelyezkedése, és jellege
- `Settings` — környezet
  - `Value Required` — Amennyiben a választás Yes, és a LapElem látható, akkor az APEX automatikus NOT NULL érték elllenőrzést végez, amikor a lapot feldolgozza.
- `Page Action when Value Chaned`: A lap tevékenysége, amikor az érték megváltozik
  - `None` — Nincs
  - `Submit Page` — Lap végrehajtása

- Redirect to Page — Visszatér az alkalmazás meghatározott lapjára
- Redirect to URL — URL címre ugrik

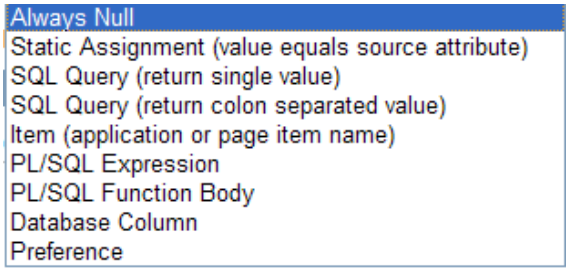
- LOV (List of Values) — a megjelenő és a visszaadott értéklista forrása a **Combobox típusú** Elemeknél
- Element — pozícionálás, tulajdonság megváltoztatása
- Source — a forrás megadása, és az értékmegjelenítés módja a **nem Combobox** típusú Elemeknél

- Source Used — Forrás használata (4.20. ábra) :
Kétféle lehet:
1:Mindig,
2: Csak ha a Session állapota nulla



4.20. ábra *A használt forrás*

- Source Type — forrás típusa a 4.21. ábrán látható.



4.21.ábra A forrás típusa

- Default — alapértelmezett érték és típusa
- Conditions — a megjelenítés feltételének jellege és megadása. Választhatunk számtalan lehetőség közül. Például: (4.22. ábra)

```
Exists (SQL query returns at least one row)
NOT Exists (SQL query returns no rows)
SQL Expression
PL/SQL Expression
PL/SQL Function Body Returning a Boolean
Request = Expression 1
Request != Expression 1
Request Is Contained within Expression 1
Request Is NOT Contained within Expression 1
Value of Item / Column in Expression 1 = Expression 2
Value of Item / Column in Expression 1 != Expression 2
Value of Item / Column in Expression 1 Is NULL
Value of Item / Column in Expression 1 Is NOT NULL
Value of Item / Column in Expression 1 = Zero
Value of Item / Column in Expression 1 != Zero
```

4.22.ábra

- `Read Only` a csak olvasható megjelenítés jellege és feltétele
- `Security` engedélyezés, hitelesítés jellege
- `Help` a Súgó szöveg megadása

Elemek külalakjának megváltoztatása

- Az elemek betűvastagságának megváltoztatása:
  a `HTML Form Element Attributes` mezőbe beírjuk:
  `class="fielddatabold"` szöveget.
- Statikus szöveg megjelenítése esetén a Forrás (`Source`)ablakba a tartomány forrása (`Region Source`) szövegablakba a megfelelő HTML kódú szöveget írjuk be, természetesen java szkrip is lehetséges ill XML forrás (4.23.ábra)

  Méretváltoztatás:
   Első sor.

  Soremelés

   Második sor.

  Színváltoztatás:
  Harmadik sor!

  Szöveg fekete, ha az egérrel fölé megyünk piros lesz
  <tr>
  <td onmouseover="this.style.color = 'red'" onmouseout="this.style.color = 'black'"> Negyedik sor </td>
  </tr>

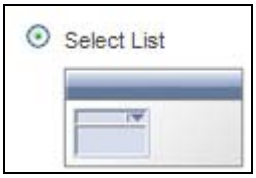
4.23. ábra *Formázott megjelenített szöveg*

Paraméterezett (dinamikus) riport készítése

A saját alkalmazásunk harmadik lapjának elkészítése. Üres lapunk már van.

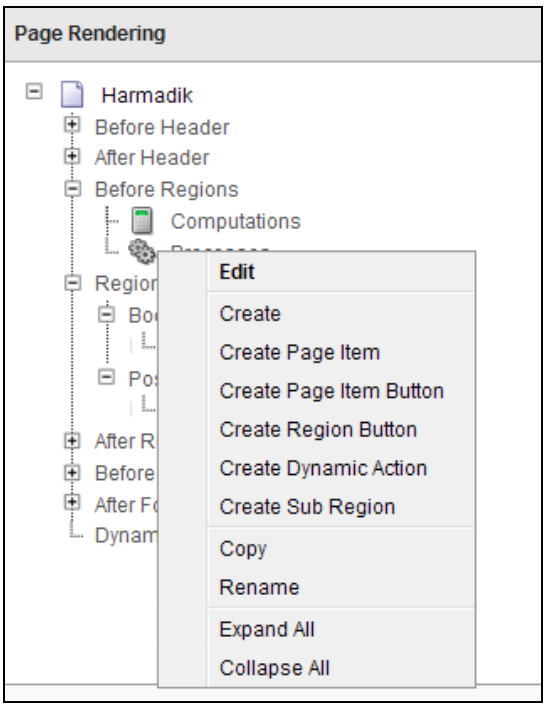
Dinamikus listázásról beszélünk, ha a listázó utasítás valamely paramétere az alkalmazás egy futásközbeni értéke, vagy egy, a felhasználó által megadott paraméter. Az alábbiakban bemutatjuk az ilyen listák létrehozásának, és egyben a környezeti változók használatának módját, lépéseit. A felhasználói értékkijelölés számára egy legördülő listát hozunk létre, az ebben kiválasztott érték jóváhagyására pedig egy gombot.

1. Hozzunk létre először egy `html` tartományt (`Region`) a lista és a gomb elhelyezésére. Amennyiben ezt Fa szerkezetben tettük meg, akkor utána már erre a regióra állva és a jobb gombra kattintva tudjuk a további Elemeket létrehozni. (4.24. ábra)
2. Hozzuk létre a legördülő listát. Ennek érdekében az `Items` szekcióban a `Create` (plusz jelű) ikonra kattintva elindítjuk az Elem létrehozó varázslót. Itt válasszuk a `Select List` elemet.

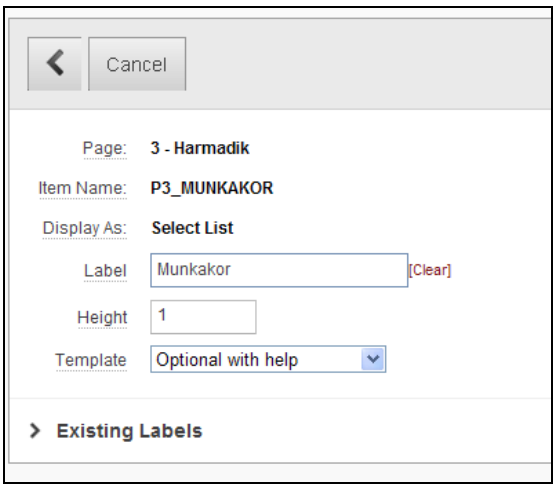


4.25. ábra Kiválasztó lista választás

Lépjünk tovább (`NEXT`), és adjunk neki nevet, amely legyen: `P3_Munkakor`.



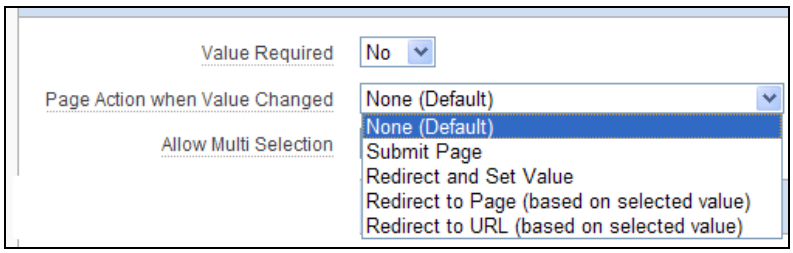
4.24.ábra Fa szerkezetben létrehozás



4.25.ábra

Ugyanitt válasszuk ki a megfelelő (például az előbb létrehozott) tartományt (a szekvencia értéket hagyhatjuk változatlanul), lépjünk tovább (`NEXT`), adjuk meg a cimkét (`Label =munkakörök`) (4.25. ábra) és `Next`.

A következő ablakban `Value Required` maradjon `No`, és a `Page Action` when `Value Changed` maradjon `None` (`Default`) (4.26. ábra). majd `Next`.

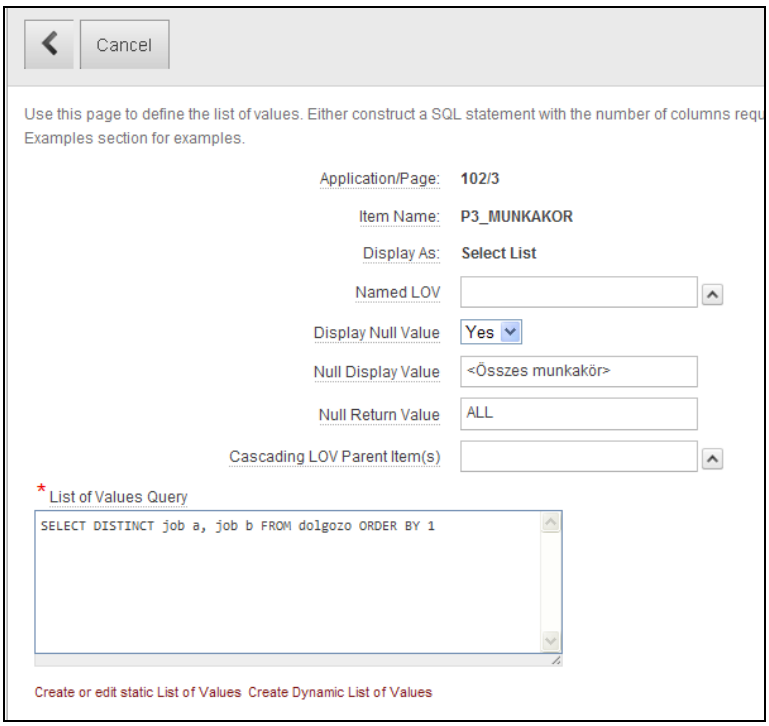


4.26. ábra tevékenység választás

Mivel ez `Combobox` típusú, ezért adatforrásként egy `SQL Select` utasítást kell megadni a `List of Values Query` szövegmezőben (4.27. ábra). A utasítás szelekciós részében az első paraméter a *megjelenő*, a második a *visszatérő* érték attribútuma. (Az utasítás szintaktikája a `List of Values Examples` ikonra való kattintáskor válik láthatóvá.)

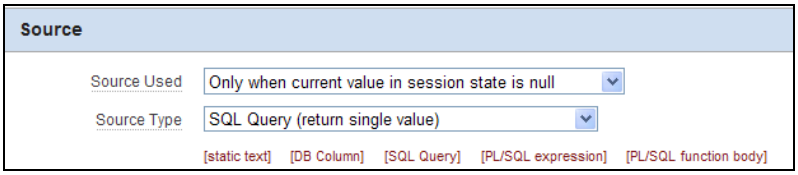
Lehetőség van a megadott szelekciós utasítás listájában nem szereplő, olyan speciális elem megjelenítésére, melynek kiválasztása esetén valamilyen más listát aktivizálunk. Ha erre igényt tartunk, akkor a `Display Null Value` mezőben a `Yes` értéket válasszuk. Ekkor a `Null Display Value` szövegmezőben megadhatjuk a speciális elemhez tartozó megjelenítendő szöveget, a `Null Return Value` szövegmezőben pedig a visszatérő értéket. Esetünkben a megjelenítendő szöveg `>Összes munkakör<` legyen, a visszatérő érték pedig `ALL`. Amikor majd feldolgozzuk a `P3_MUNKAKOR` lista elem által visszaadott értéket, akkor egy olyan `Select` utasítást fogunk aktivizálni, mely nem tartalmaz munkakör szerinti szűrést.

Ha az alábbi ábrán a `Display Null Value` mezőben a `Yes` értéket választottuk, de a `Null Display Value` szövegmezőt üresen hagyjuk, akkor a megjelenő szöveg egy `%` jel lesz, és ha a `Null Value` szövegmezőt hagyjuk üresen, akkor a visszatérő érték a `%null%` karaktersorozat lesz. Természetesen a visszatérő érték akkor is lehet `NULL` érték, ha nem kezeltük le, vagyis a a `Display Null Option` mezőben a `No` értéket választottuk. `Next`.



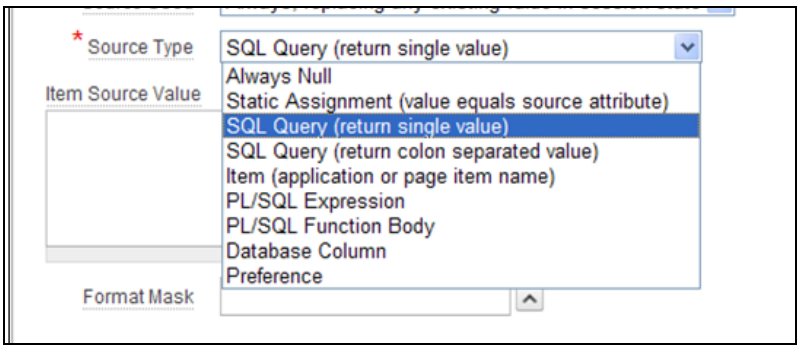
4.27.. ábra. *A legördülő lista megjelenő értékeinek SQL utasítása*

Itt állítható be, hogy mikor szeretnénk használni a legörsülő listát (`Source Used`).Ezt `LOV` típusu elemeknél figyelmen kívül kell hagyni.



4.28. ábra A forrás típusa

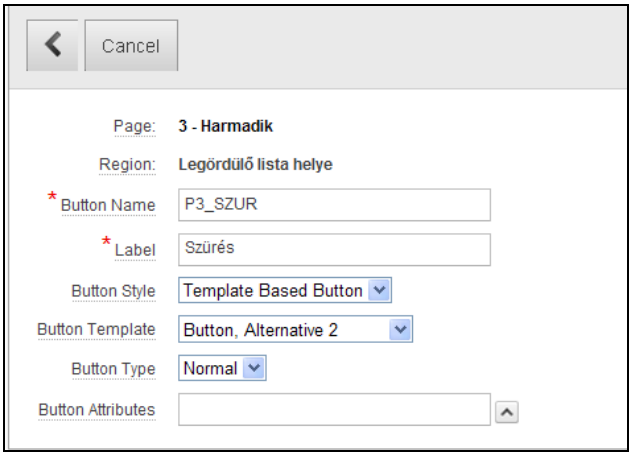
Beállítható még az Elem forrásának típusa, azaz az SQL utasítás, különböző PL/SQL utasítások illetve Statikus értékek. A kiválasztott beállításnak megfelelő utasítást írjuk be a forrás mezőbe (4.29. ábra)



4.29. ábra *nem-Combobox típusú elemek választható forrása*

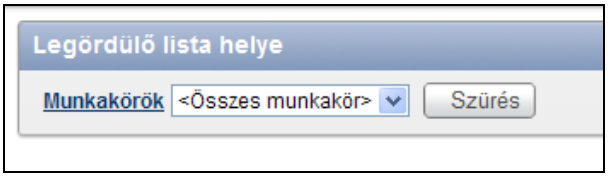
Fejezzük be (`Create Item`).
Mást most nem kell beállítani.

3. Következő lépésben hozzuk létre a kijelölést érvényesítő gombot. Ennek érdekében most a `Buttons` szekció varázslóját indítsuk el (plusz jelü ikonnal). Tegyünk fel egy `Szűrés` feliratú Elem típusú gombot (a `Request` értéke legyen `mehet`), melyre kattintva fog megjelenni a szűrt vagy a teljes lista (4.30. ábra).



4.30. ábra. *A szűrés gomb beállításai*

Legyen a gomb mintája `Alternative2`, majd hozzuk létre a gombot (`Create Button`).
Mivel elkészítettük a harmadik lapon a legördülő listát és a kijelölést érvényesítő gombot, ezért most jelenítsük meg ("futtassuk le") ezt a lapot (4.4.31 ábra).



4.31. ábra. *A legördülő lista és a gomb*

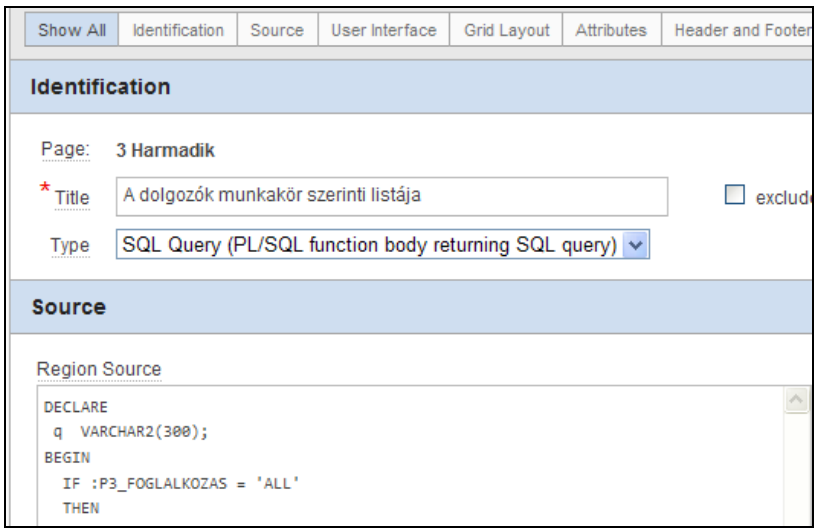
4. Az alábbiakban bemutatjuk a felhasználó által kijelölt szűrésnek megfelelő riport létrehozását.
   Szerkesszük a harmadik oldalt (`Page Definition`). Hozzunk létre egy riport tartományt (`Report Region`, `SQL Report`). Adjuk meg a fejlécet (`Title`), amely legyen `A dolgozók foglalkozás szerinti listája`, mely egyben az elkészítendő riport neve is a Tartomány-Definiáló részben (`Regions`). Ezután írjuk meg a lekérdezést.

   4.A. Készítsük el a lekérdezést a `Query Builder` segítségével. Ezzel az eszközzel csak egyszerű (Where) szűréseket tudunk megadni, és arra sincs lehetőségünk, hogy paraméteres hivatkozást használjunk (amire például a fenti legördülő listára való hivatkozásnál volna szükség), arra azonban alkalmas, hogy a szelekciós listára és a táblára való hivatkozás szintaktikailag helyes alakját automatikusan előállítsuk. A `Query Builder` környezetében lévő `Run` gomb segítségével ellenőrizhetjük a létrehozott lekérdezés helyességét. Ezután a szintén itt található `Return` gomb segítségével visszatértünk a riportot létrehozó lapra, majd az `Enter SQL Query...` szövegmezőben módosítjuk (ezúttal létrehozzuk) a lekérdezés Where feltétel részét. Ennek során adjuk meg a legördülő listára való hivatkozást, másrészt azt, hogy a felhasználó `Összes munkakör` választásakor a gomb elemtől kapott `ALL` visszatérő érték esetén a listázó utasítás valóban megjelenítse az összes (tehát különböző foglalkozású) dolgozót. (A fentiekben létrehozott legördülő lista visszatérő értékére `:P3_MUNKAKOR` módon hivatkozhatunk.) A listázó utasítás végleges alakja esetünkben (ahol vastagon jelöltük az általunk az utasításhoz utólag hozzáírt részeket):

```
     DOLGOZO.JOB as JOB,
       DOLGOZO.MGR as MGR,
       DOLGOZO.HIREDATE as HIREDATE,
       DOLGOZO.SAL as SAL,
       DOLGOZO.COMM as COMM,
       DOLGOZO.DEPTNO as DEPTNO
   from DOLGOZO DOLGOZO
  WHERE DOLGOZO.JOB = :P3_MUNKAKOR or (:P3_MUNKAKOR = 'ALL')
```

   `Next`
   A továbbiakban még kiválaszthatjuk a riport megjelenési alkját. (`Report Template`), azaz válasszuk most a `tempalte 2: Alternative .... Create Region.`
   Jelenítsük meg az aktuális (harmadik) lapot. Láthatjuk, hogy a felhasználói kiválasztásra valóban helyesen működik a lekérdezés.

   4.B. Készítsük el a lekérdezést közvetlenül az `Region Source` szövegmezőben, de ezúttal ne SQL-utasítást, hanem egy, visszatérő értékként SQL-utasítást adó PL/SQL függvényt (pontosabban annak a törzsét) írjuk meg (4.32.ábra). ezt állítsuk is be az `Identification` ablakrész típusánál.



4.32. ábra. *A riport lekérdezésének PL/SQL utasítása*

```
DECLARE
 q  VARCHAR2(300);
BEGIN
  IF :P3_MUNKAKOR = 'ALL'
  THEN
    q:='SELECT * FROM dolgozo';
  ELSE
    q:='SELECT * FROM dolgozo WHERE job = ';
    q:=q||':P3_MUNKAKOR';
  END IF;
  RETURN q;
END;
```

Mivel az oszlop fejléc (`Columns Headings`) megállapításához már szükséges egy belső értelmező (vagyis az nem származtatható közvetlenül a beírt kódból), ezért a értékeként a `Generic Columns` választás szükséges. Kattintsunk a Riport oszlopainak szerkesztésénél a Headings Type Column Name rádiógombra..és akkor az eredeti oszlopfejlécek jelennek meg.

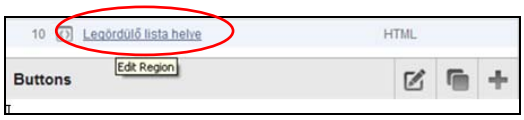
Ügyeljünk az utasításban használt aposztróf karakter használatára, mivel a wordben alapértelmezett jel nem megfelelő.

Az eddigiek alapján az az alkalmazás harmadik oldala megjelenik, de még nem szép, nem megfelelő a felhasználói felület. Egyrészt a lap közepén kezdődik a riport , másrészt a fejlécek szövege nem magyar. Módosítsuk ezt is.

Tartományok helyzetének változtatása

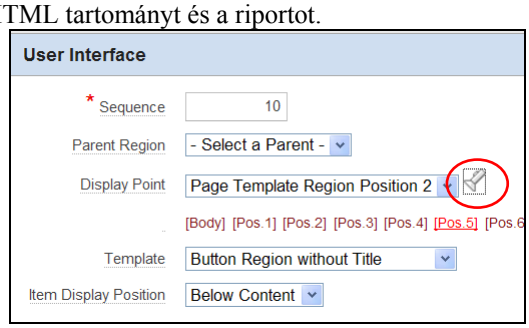
Menjünk a harmadik oldal Lapszerkesztésébe és először a HTML régiót, majd a Riport régiót helyezzük a képernyő másik pontjára.

Menjünk a 3. lap Lapszerkesztő felületére (`Edit Page 3`). Kattintsunk Tartomány szekcióban a HTML régió szerkesztésére, azaz a `Legördülő lista helye` linkre (4.33.ábra).

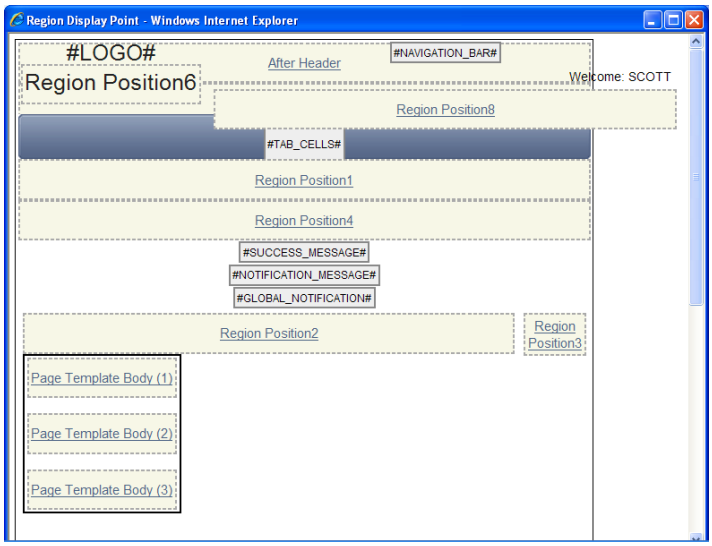


4.33. Tartomány szerkesztése

Kattintsunk a `User Interface` fülre, vagy gördüljünk le ebbe az ablakrészbe.
Itt kattintsunk az elemlámpára és nézzük meg hogyan tehetnénk egymás mellé, vagy alá a HTML tartományt és a riportot.

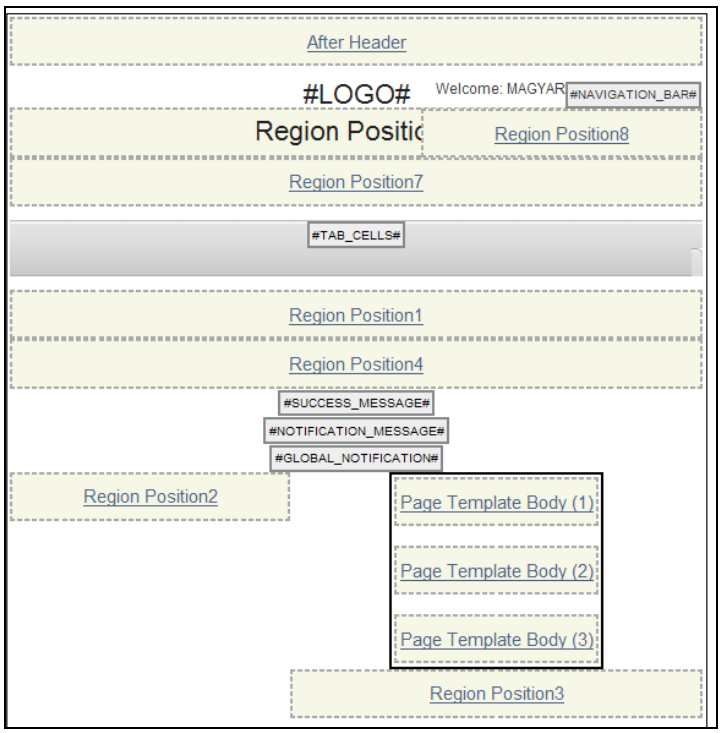


4.34.ábra *Megjelenési pont helyzetei*



4.35.ábra *Megjelenési pont helyzetei egyik mintánál*

Válasszuk a `Region Position2-t` a HTML tartomány számára.
vagy másik Mintánál (4.36. ábra)



4.36.ábra *Megjelenési pont helyzetei másik mintánál*

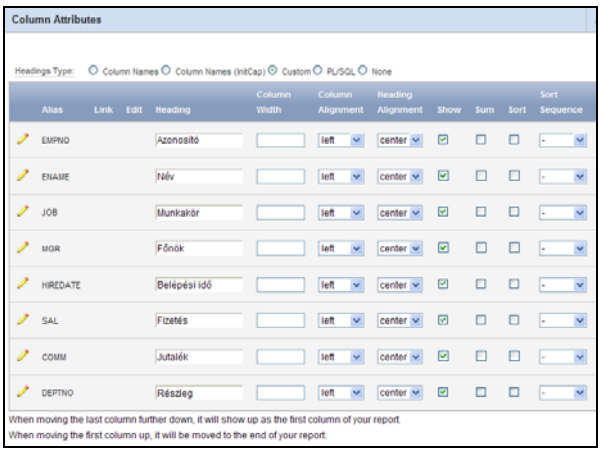
Nézzük meg a `Szűrő lista` riportjának felhasználói felületét.Szerkesszük meg tehát ezt is. Ezt ne változtassuk meg.

Futtassuk le a harmadik lapot.Most már a helyzetük megfelelő (`User Interface`), de még nem magyar a fejléc. A már ismertett módon írjuk át magyarra.

(`Edit Page 3-` `Szűrő lista` link- `Report Attributes` fül- oszlopnevek átírása és `Apply Changes`)

Riport fejlécének módosítása

A korábban elkészült riport fejlécének módosítása érdekében Szerkesszük a riportot. Kattintsunk a Lap-Definíciós (`Page Definition`) ablak Tartomány-Definiáló részében (`Regions`) `A dolgozók foglalkozás szerinti listája` nevű riport hiperhivatkozására, majd a `Report Attributes` fülre.



4.37. ábra. *Riport fejlécének módosítása*

A megjelenő ablakban az egyedi fejléc-megadást válasszuk (a `Heading Type` értéke legyen `Custom`), ilymódon átírhatjuk a fejlécben megjelenő neveket. Formázhatjuk is a megjelenést a `Layout and Pagination` szekcióban, és rendezhetjük a `Sorting` szekcióban. A riport-megjelenítési hibák esetére hibaüzeneteket küldhetünk a `Message` szekcióban, és beállíthatjuk kiválasztott oszlop egymásalatti soraiban azonos értékek megjelenítésének elkerülése érdekében az ismétlésmentes megjelenést a `Break Formatting` szekcióban.

A fenti beállításokat követően (`Apply Changes`), a riport megjelenése az alábbi lesz (4.38.ábra):

4.38. ábra. *Magyar fejléccel ellátott riport*

Most már csak egy apró hibánk van. Első indításkor a harmadik lapon a riport helyén a `no data found` hibaüzenet jelenik meg. A lap betöltésekor, még nem kattintottunk a Szűrés gombra, tehát ebben az értelemben a hibaüzenet jogos.

"no data found" hibaüzenet kiküszöbölése.

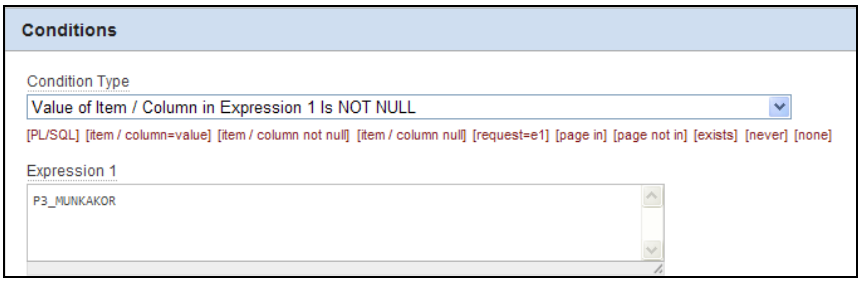
Az alapfeladat; milyen módon lehet elérni azt, hogy egy riport addig ne jelenjen meg, amíg a hozzárendelt listát aktivizáló tevékenységet (jellemzően gombra való kattintást) legalább egyszer el nem végeztük. Ez egy olyan feltétel-beállítás, amelyikről egy lap első meg-

jelenítésekor kell gondoskodni, egyébként a riport helyén a `no data found` hibaüzenet jelenik meg.

Kattintsunk a Lap-Definíciós (`Page Definition`) ablak a Tartomány-Definiáló (`Regions`) szekciójában a `A dolgozók munkakör szerinti listája` nevű riport hiperhivatkozására. A megjelenő ablak a Megjelenítési Feltétel (`Conditional Display`) szekciójában előbb állítsuk be a Feltétel Típusát (`Condition Type`).

Mivel a `Szűrés` gomb lenyomásához szeretnénk a riport megjelenítését hozzárendelni, ezért feltételként e gomb azonosító (`REQUEST`) értékét adjuk meg a feltételben. Az ehhez tartozó Feltétel Típus tehát a `Value of Item/ Column in Expression1 is NOT NULL`, és a Kifejezéshez (`Expression 1`) pedig írjuk be a `P3_MUNKAKOR` legördülő lista nevét (4.39. ábra).

Alkalmazzuk a módosításokat (`Apply Changes`). Ekkor a programunk már valóban hibátlanul működik. A riport helyén most már sem az alkalmazás indításakor, sem a harmadik lap későbbi megjelenítésekor nem jelenik meg hibaüzenet.

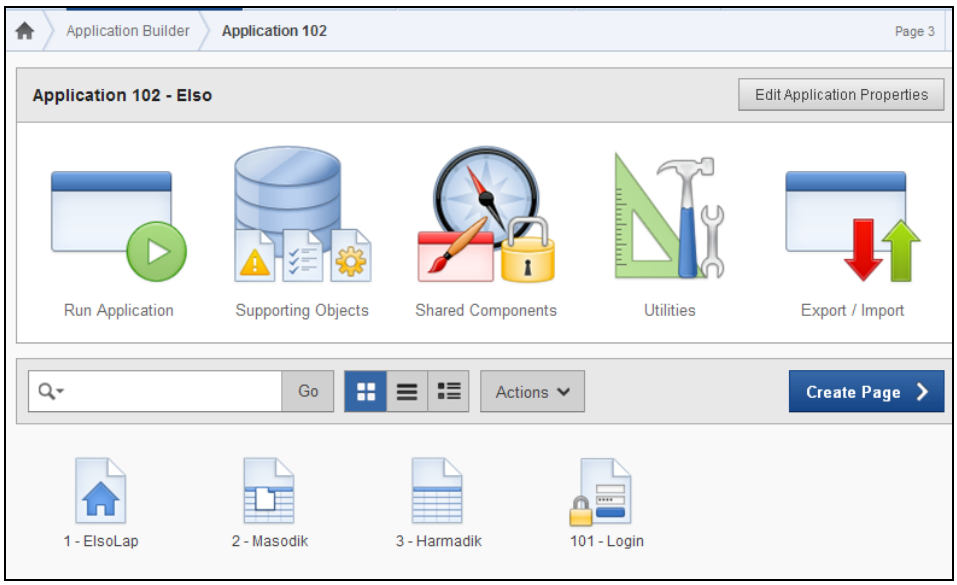


4.39. ábra. *Riport megjelenítési feltételének beállítása*

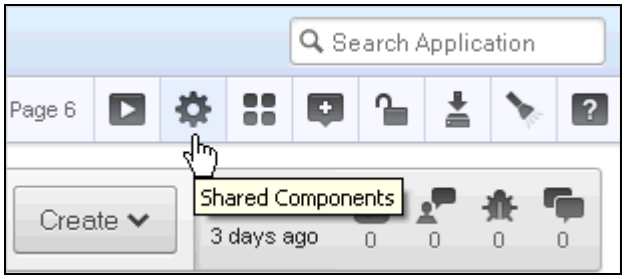
Felületi Minta megváltoztatása

Alkalmazásunk tehát elkészült, de utólag rájöhetünk, hogy más Felületi Mintával (`Theme`) szeretnénk azt létrehozni (szin, forma, elhelyezkedés, stb). Ez is könnyen megoldható.

Menjünk vissza az alkalmazásunkhoz, azaz a futtatáskor látható Fejlesztői Eszközsoron `Application 102` (az alkalmazás azonosító száma), és válasszuk a `Shared Components` ikont (középső)



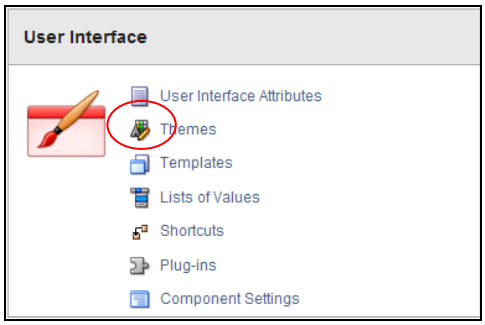
4.40. A megosztott komponens elérése



4.41.megosztott komponens elérése

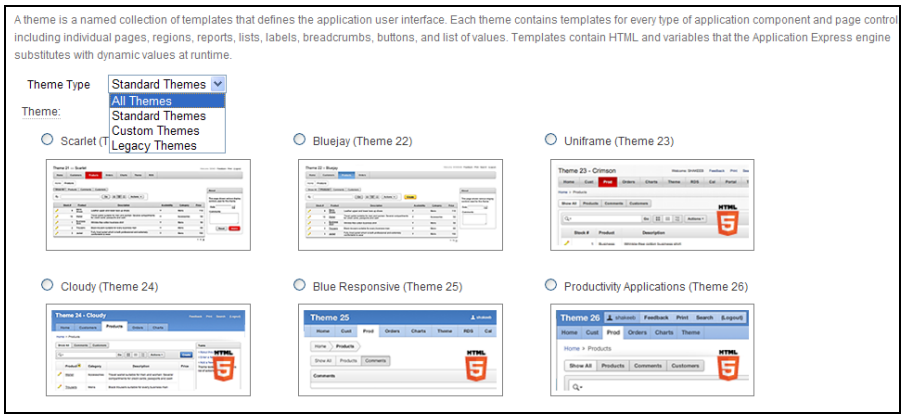
Másik lehjetőség válasszuk a fogaskereket (4.41. ábra). természetesen attól függően, hogy hol vagyunk.

Itt válasszuk ki a jobboldali második ablakot a Felhasználói Felület (`User Interface`) (4.42. ábra) és ezen belül a Felületi Minta(`Themes`) ikont (4.43. ábra).



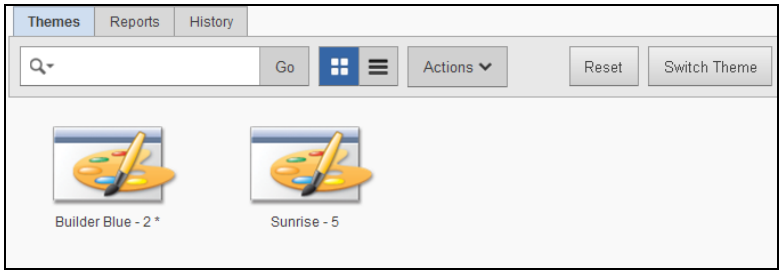
4.42. ábra *A felhasználói felület*

A megjelenő ablakban, mivel még nincs másik témánk az alkalmazásunkhoz kattintsunk a Létrehozás (`Create`) gombra . Itt válasszuk a Raktárból (`From the Repository`) rádiógombot és `Next`. Itt válasszuk az Asztal Felületet (`Desktop`) , majd `Next`. A megjelenő ablakban választhatunk kedvünkre a felületi minták típusaiból, és ezon belül számtalan előre definiált felületi mintából.



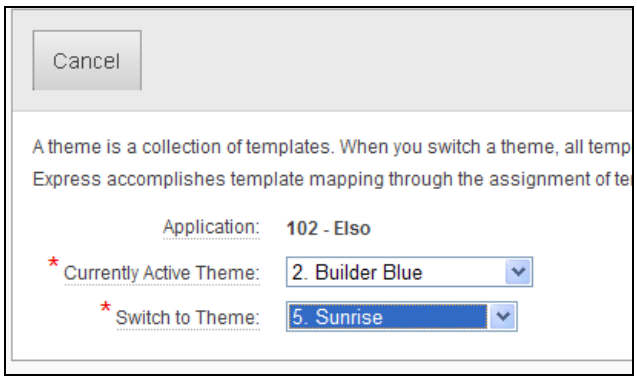
4.43.ábra *Lehetséges minták*

Válasszuk az örökölt (`Legacy`) témák közül a `Sunrise` (`Theme 5`) rádiógombot, majd `Next` (4.44.ábra) , itt ellenőrizhetjük, majd `Create` gomb.



4.44. ábra *Új minta választása*

Most már van másik témánk is, és erre át lehet váltani. Kattintsunk az Átváltás (`Switch Theme`) gombra. A megjelenő ablakban válasszuk ki az aktív és a átváltó Feületi MIntát, ellenőrizzük a kompatibilitást, erősítsük meg kattintsunk az Átváltás (`Switch Teme`) gombra.



4.45. *Minta cserélése*

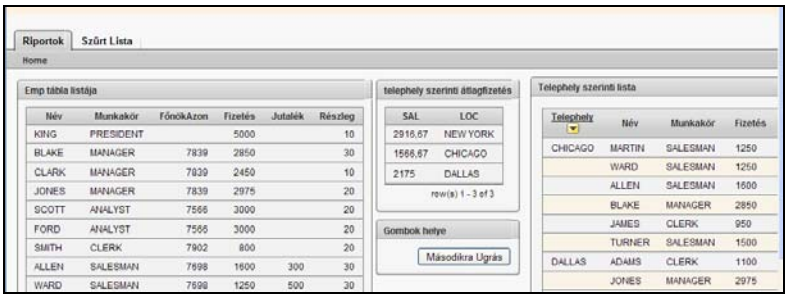
Futtassuk le az alkalmazásunkat (4.46.ábra)

Másik mód, ahogy ide eljuthatunk:

Mivel a Felületi Minta az alkalmazás megosztott komponens, ezért bármelyik lap szerkesztő ablakából kiindulva elvégezhetjük a módosítást.

Kattintsunk az alkalmazásunk egyik lapjának Lap-Definíciós (`Page Definition`) ablakában a Megosztott Komponensek (`Shared Components`) rész Felületi Minta (`Theme`) szekciójában a `Létrehozás` (plusz jelü) ikonra.

Az alkalmazás újraindítása, vagy bármelyik lap megjelenítése esetén most már az új minta fog megjelenni.

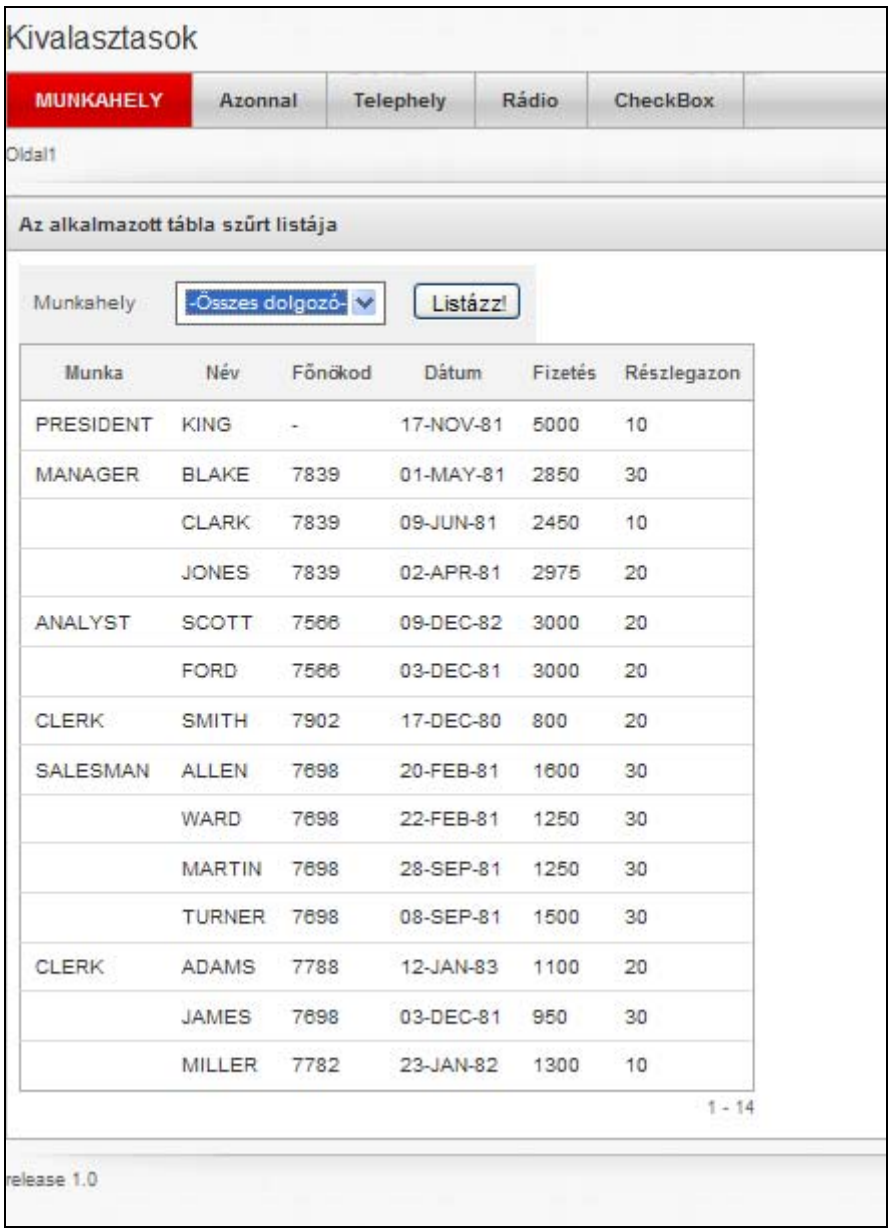


4.46. ábra. *Az alkalmazásunk megváltoztatott felületi mintával*

Példa:

Készítsünk alkalmazást, amelynek lapjain különböző elemekkel alkalmazzuk a dinamikus listát kiválasztásra.

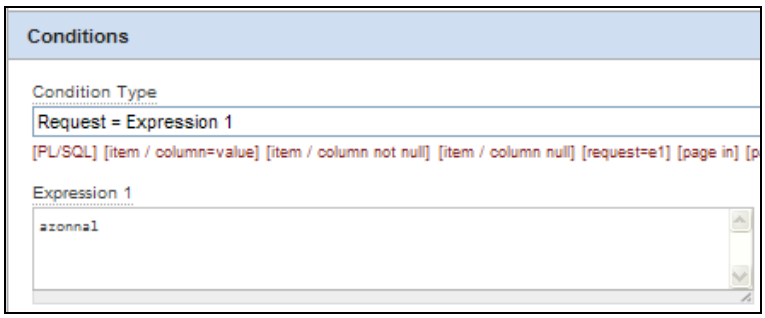
035_KivalasztasiModszerek (03).sql.



4.47.ábra Az alkalmazás első lapja

1.lap Munkakör szerint választjuk ki az alkalmazottakat.

2. lap Munkakör szerint választjuk ki az alkalmazottakat, de a kiválaszto elemre kattintva ez a kiválasztás azonnal megtörténik. feltételt is adtunk, hogy első esetben ne jelenjen meg hibaüzenet. A feltétel a következő lehet (4.48. ábra).

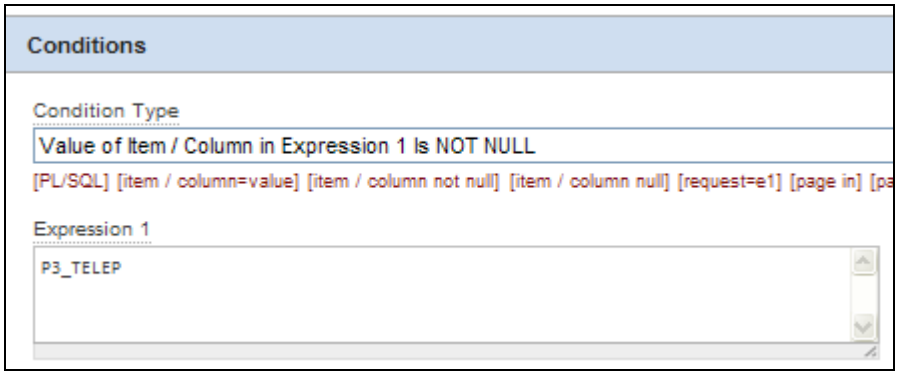


4.48.ábra Feltétel

Az azonnal a gomb requestje.
A lekérdezés SQL query beírással,

```
select * from alkalmazott
where job = :P2_MUNKA or :P2_MUNKA='ALL'
```

3.lap Telephely szerint listázzuk ki az alkalmazottaka. Ez tehát két táblás lekérdezés
Itt a feltétel az alábbi:



4.49.ábra Telephely feltétele

3.lap Az alkalmazottak kiválasztása rádiógombbal és azonnali feldolgozással (submit)
4.lap Az alkalmazottak kiválasztása Check Boxxal és gomb segítségével.

Tábla megjelenítése szövegdobozban

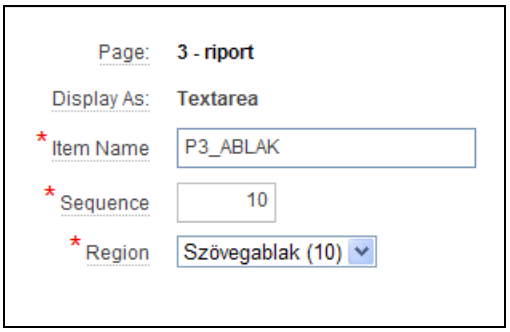
Jelenítsük meg egy szövegdobozban (`textarea`) a dolgozók neveit, fizetésüket és részleg-azonosítójukat ez utóbbi szerint rendezve.

Megoldás

Hozzunk létre egy új `Blank` típusú lapot.

Helyezzünk az oldalra egy szövegdobozt (`textarea`). Ennek azonosítója legyen `P3_Ablak`.

Továbbá töltsük ki a címke (`Label`) (4.50. ábra), `Element`, és forrás (`Source`) (4.51. ábra), ablakokat a következőképpen:



4.50. ábra. *A címke beállítása*

Állítsuk be a szövegdoboz méreteit.

Töltsük ki a forrást (`Source`):

A Forrás használata (Source Used): `Only when current value....`

A Forrás Típusa: `PL/SQL Function Body`

A munkamenet állapotának fenntartása (`Maintain session...`): `Per User`

Ezután írjuk meg a PL/SQL függvényt:

```
DECLARE
  Sor  VARCHAR2(60);
BEGIN
-- fejléc
  :P3_Ablak := 'DolgozóNév'||'  Fizetés'||'   RAzonosító' ||
CHR(13)||'==========================='|| CHR(13) ;
-- Rejtett (implicit) kurzorral menjünk végig a táblán
   FOR TablaNev IN (SELECT ename, sal, deptno FROM dolgozo Order by 3)
  LOOP
    sor := RPAD(TablaNev.ename,10)||RPAD(' ',7) ||TablaNev.sal
||LPAD(TablaNev.deptno,8)|| CHR(13) ;
    :P3_Ablak := :P3_Ablak || sor;
  END LOOP;
END;
```

Source

Source Used: Only when current value in session state is null

Source Type: PL/SQL Function Body

[static text] [DB Column] [SQL Query] [PL/SQL expression] [PL/SQL function body]

Maintain session state: Per session

Source value or expression

```
DECLARE
  Sor  VARCHAR2(60);
BEGIN
-- fejléc
  :P3_Ablak := 'DolgozóNév'||'  Fizetés'||'   RAzonosító' || CHR(13)
||'==========================='|| CHR(13) ;
-- Rejtett (implicit) kurzorral menjünk végig a táblán
   FOR TablaNev IN (SELECT ename, sal, deptno FROM dolgozo Order by 3)
  LOOP
    sor := RPAD(TablaNev.ename,10)||RPAD(' ',7) ||TablaNev.sal ||LPAD(TablaNev.deptno,8)|| CHR
(13) ;
    :P3_Ablak := :P3_Ablak || sor;
  END LOOP;
END;
```

4.51. ábra. *A forrás megírása*

Végül hozzuk létre az elemet kattintsunk a létrehoz (`Create`) gombra. Futtassuk le:

Szövegablak

Ablak

```
DolgozóNév Fizetés  RAzonosító
===========================
CLARK       2450    10
MILLER      1300    10
KING       5000    10
FORD        3000    20
SCOTT       3000    20
JONES       2975    20
SMITH      800    20
ADAMS       1100    20
WARD        1250    30
MARTIN      1250    30
TURNER       1500    30
JAMES      950    30
ALLEN       1600    30
BLAKE       2850    30
```

4.52. ábra. *Szövegablakban megjelenő lista*

Mint láthatjuk a 4.52. ábrán, a dolgozó tábla megfelelő adatait listázza, lapozás helyett.

5. fejezet. Folyamatok létrehozása

Elágazás (Branches)

Az Elágazás ,(`Branch`) egy olyan utasítás, amely kijelöli, hogy a vezérlés melyik lapra, eljárásra vagy URL-címre adódjon át.

Egy új Elágazást (`Branch`) a szekciójának varázslójával lehet létrehozni. Ennek során egyrészt meg kell adni azt, hogy az Elágazás az alkalmazás működésének mely pontján aktivizálható (`Branch Point`), valamint az Elágazás típusát (`Branch Type`). A létrehozott Elágazás típusa meghatározza a `Branch` típusát is.

Aktivizálási Pont és Tevékenység

Ha az alkalmazásban alapértelmezett Fülre kattintunk, akkor az APEX motor megállapítja a `session state`-et, végrehajtja a számításokat és átkapcsol a cél Lapra. Nem futtat le semmilyen folyamatot, vagy explicit meghatározott elágazást. Abban az esetben , amikor a Lap végrehajtódik anélkül, hogy Fülre kattintottunk volna, az APEX motor egyértelműen meghatározza azt az elágazást, amely közvetlenül a felhasználótól érkezett.

Az Elágazás (`Branch`) feltétele:

Mint más vezérlőnek az Elágazásnak is adhatunk feltételt. A feltétel megadása esetén, ki kell választanunk a feltétel típusa listából egy megfelelő feltételt, és az Expression mezőbe be kell írni a szöveget, amely megvalósítja a változáshoz a feltétel típusát.

Az aktivizálási pont (`Branch Point`) lehet:

- `On Submit: Before Computation` beállítás esetén a megjelenítési továbblépés bármilyen Számítási Folyamat végrehajtása (például a bevitt adat ellenőrzése) előtt történik. Jellemzően ezt használjuk olyan gomboknál, amelyeknél nincs szükség semmilyen feldolgozási folyamatra (például a `Cancel` gomb).
- `On Submit: Before Validation` beállítás esetén, ha az adatellenőrzés hibát állapít meg, akkor az adatbevitel érvényesítése helyett a megjelenítés egy hibaüzenettel visszatér az adatbevitelhez.

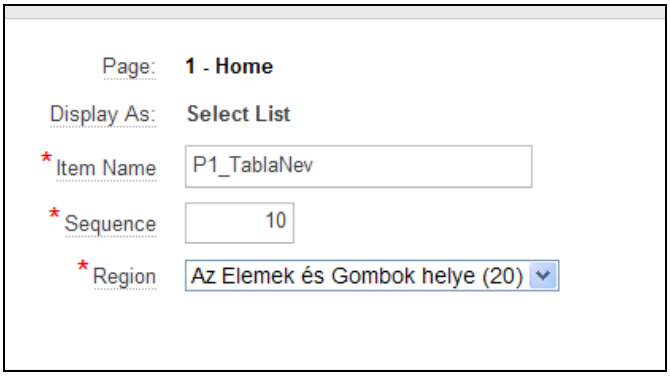
- `On Submit: Before Processing` beállítás esetén a megjelenítési továbblépés az adatellenőrzés és a tényleges adatbevitel után történik, de az alkalmazás további folyamatai előtt.
- `On Submit: After Processing` beállítás esetén a megjelenítési továbblépés az alkalmazás teljes aktuális feldolgozási folyamata után történik.
- `On Load: Before Header` beállítás esetén az alkalmazási logika szerint következő lap betöltése helyett egy másik (alkalmazásbeli, vagy külső, URL-címével adott) lapot jelöl ki megjelenítésre.

Az Elágazás típusától függően kiválasztható a következő: hozzáadandó tevékenység tulajdonságaira vonatkozó információ:

- Az a Lapszám, amelyre elágazni kívánunk
- PL/SQL eljárás, amely végül is előkészíti az elágazást a cél Lapra
- URL cím

Listából kiválasztott tábla megjelenítése (példa)

Készítsünk egy alkalmazást, mely egyetlen lapból áll, és ennek baloldalán álljon a magyar-felhasználó tábláinak legördíthető listája (`Select List`), valamint a `Megjelenítés` és a `Bezárás` gombok, a jobboldalán pedig a felhasználó által kijelölt tábla listája.



5.1. ábra. *A Lista létrehozásának fázisa*

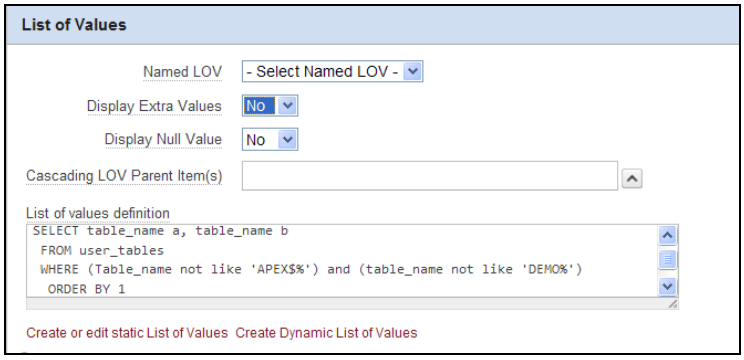
A megvalósítás menete a következő:

1. Első lépésként hozzunk létre egy új alkalmazást `Blank` típusú lappal.
2. Ezután hozzunk létre a lapon egy új html tartományt, és tegyünk erre egy legördülő listát, melynek neve `P1_TablaNev`. (5.1. ábra). Next.
   Adjunk cimkét a listának. Next., Next.
   A kiválasztó lista tartalmazza a felhasználó tulajdonában lévő táblák neveit ábécé szerint rendezve. A listaértékeket a
   Amennyiben adminisztártor userként vagyunk benn, ha síma fejlesztőként akkor APEX$% helyett ADMIN% kell.

```
SELECT table_name a, table_name b
```

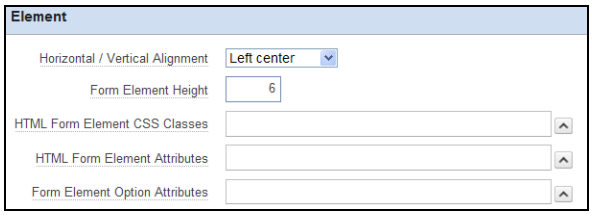
```
FROM user_tables
WHERE (Table_name not like 'APEX$%') and (table_name not like 'DEMO%')
 ORDER BY 1
```

utasítással generáljuk. majd Next.(5.2. ábra) és `Create Item`



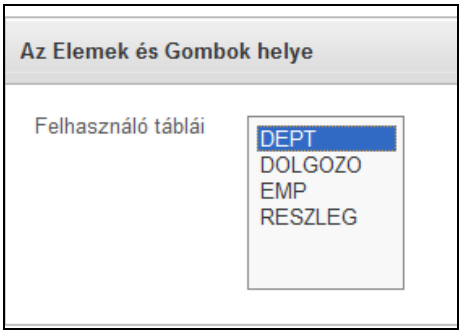
5.2. ábra. *A lista létrehozásának következő fázisa: a táblák neveinek SQL lekérdezése*

A legördülő elem magasságát változtassuk nagyobbra(5.3. ábra) (`Form Element Height` = `6`)



5.3. ábra *Az elem további beállításai (igazítás, magasság)*

Futtassuk le :

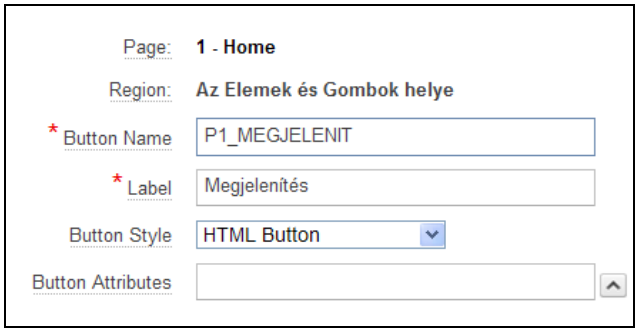


5.4. ábra *A megjelent legördülő elem*

Láthatjuk, ha kevesebb a megjelenő lista elemeinek száma, mint a magasság, akkor nincs gördülés (5.4.ábra). Amennyiben több tábát jelenítünk meg, akkor újra megjelenik a legördíthetőség.

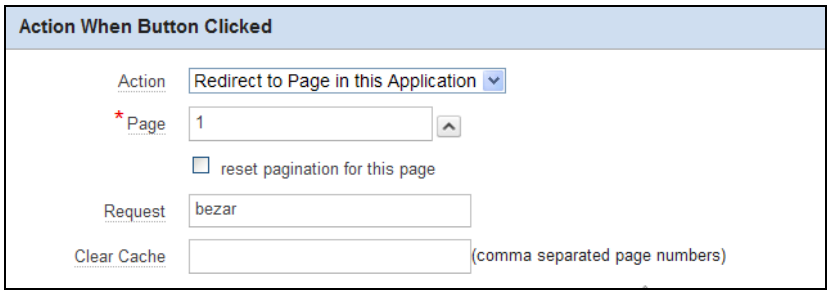
3. Tegyünk a kiválasztó lista alá két Elem-típusú gombot (5.5. ábra). Az egyik jelenítse meg a kiválasztott tábla listáját (a címkéje legyen `Megjelenítés`), a másik pedig tüntesse

el e listát (ennek a címkéje pedig legyen `Bezárás`), a stílusuk pedig legyen `HTML Button`.



5.5. ábra *A megjelenítés gomb*

Az első neve legyen `P1_MEGJELENIT`, Action legyen `Submit` a második gomb neve legyen `P1_BEZAR`, azonosítója pedig `bezar`.
`Action`: legyen Irányítás az alkalmazás egy lapjára (`Redirect to Page in this application`)



5.6. ábra A gomb tevékenységének kiválasztása

Futtassuk le:

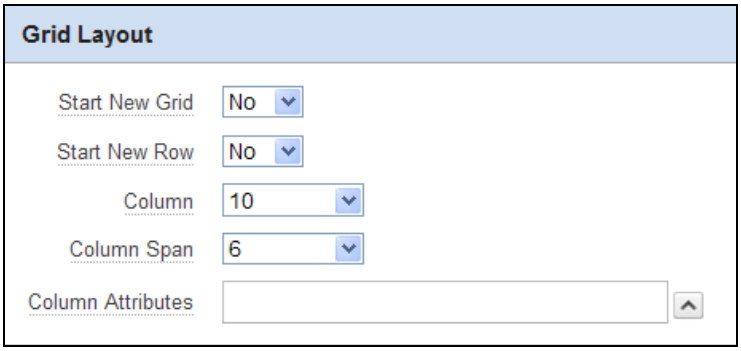
5.7. ábra A felhasználó táblái a legördülő listában

4. Ezután hozzuk létre a kiválasztott tábla listáját megjelenítő riport tartományt, a legördülő lista mellett. Nevezzünk el ezt html-riportnak, mivel a lap betöltésekor (`Submit`) a felhasznált böngésző program (például Internet Explorer) html-értelmezőjének kezdeményezésére dolgozza fel az adatbázis-szerver.

Forrása (`Source`) legyen az alábbi, SQL lekérdezéssel visszatérő PL/SQL függvénytörzs (dinamikus táblamegjelenítés):

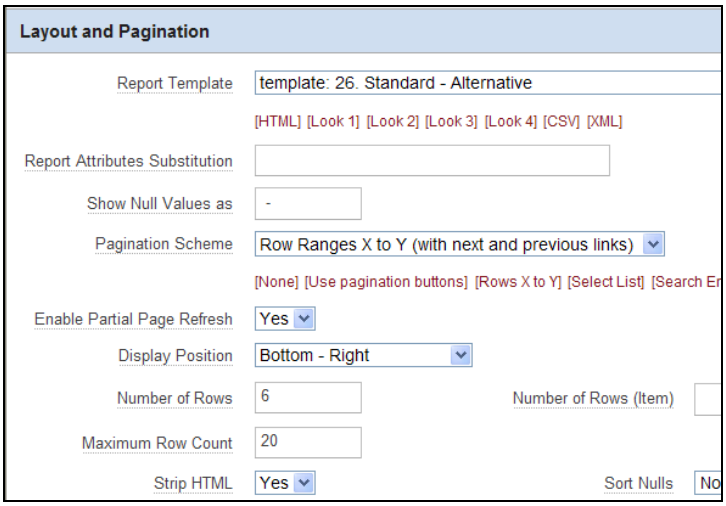
```
begin
  return 'SELECT * FROM ' || :P1_TablaNev;
end;
```

Mivel a leírt PL/SQL utasítás szintaktikai elemzéséből nem lehet közvetlenül megállapítani a lista oszlopneveit, ezért a futási időben generált oszlopnév beállítást (a `Columns Headings` opcióhoz a `Use Generic Columns Names` rádiógombot) válasszuk.
Ne kezdődjön új rácsvonalon, ne kezdődjön új sorban amint azt alább láthatjuk(5.8. ábra)



5.8. ábra *A gomb elhelyezése a másik mellett*

Utólag tudjuk beállítani a riport tulajdonságainál



5.9. ábra *Megjelenés*

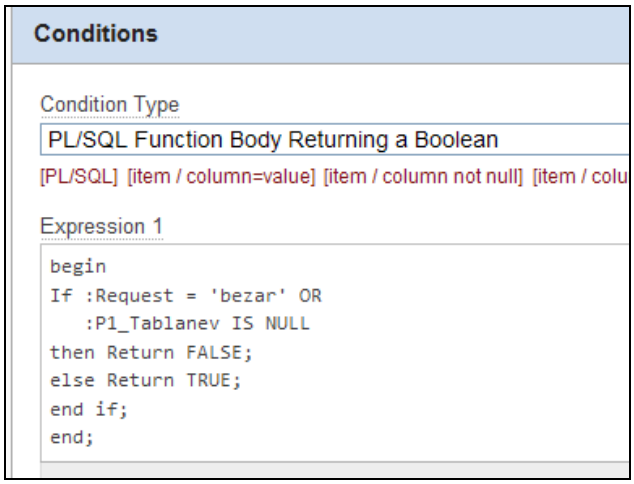
Az aktuális ablakban a megjelenített sorok számaként (`Number of Rows`) adjuk meg a 6 értéket. (Ilymódon lehetőségünk lesz a 14 soros listában lapozni!) (5.9. ábra)

5. Mivel számunkra a két gombhoz két különböző tevékenység tartozik, így szükségünk van két különböző tevékenységre. Egyik a tábla listázása, amikor kijelölünk egy táblanevet a

legördülő listában, a másik, hogy ne jelenjen meg a tábla. harmadik lapozás esetén csak az történjen. Az Lap első megjelenésekor, tehát még nincs kijelölés, akkor sem jelenjen meg hibaüzenet.
A megoldásunk azonban még nem tökéletes, ugyanis egyrészt az alkalmazásba való belépéskor az alábbi hibaüzenetet kapjuk:

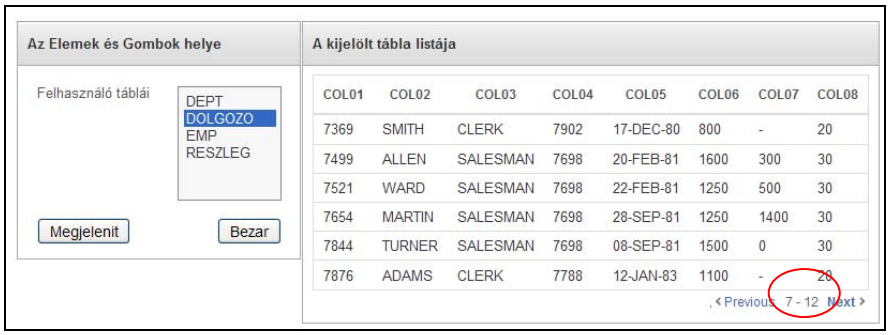
```
failed to parse SQL query:
ORA-00903: invalid table name
```

A másik probléma, hogy nem tudunk lapozni a tábla listában, ugyanis a bekarikázott lapozó ikonra kattintva a Bezárás gombhoz hasonlóan eltünik a lista
Mindkét a problémának a megoldása az, hogy a riport megjelenítését a táblakiválasztás elvégzése, valamint a globális `request` változó megfelelő gombhasználati értékének feltételéhez kötjük. Megjegyezzük, hogy az Apex környezetben globálisan létező `request` változóra a PL/SQL-környezetben környezeti változóként (tehát kettősponttal kiegészített módon) kell hivatkozni.
A fentieket úgy tudjuk megvalósítani, hogy `A kiválasztott tábla listája` nevű riport Szerkesztő Ablakának (`Edit Region`) a Megjelenítési Feltétel szekciójában (`Conditional Display`) a Feltétel Tipusaként (`Condition Type`) a `PL/SQL Function Body Returning a Boolean` elemet választjuk ki, az `Expression 1` mezőbe pedig az alábbi PL/SQL függvény törzset írjuk:
A feltétel egy PL/SQL függvény, melynek visszatérési értéke igaz vagy hamis.
A request a gomb azonosítója.
Tehát ha lenyomtuk a `Bezár` gombot, vagy `:P1_Tablanev` kiválasztott listelem értéke Nulla, akkor Hamis értékkel térjen vissza, azaz ne jelenjen meg semmi a a kiválasztott tábla helyén, amennyiben nem így van, akkor a visszatérő érték legyen igaz, és a kiválasztott tábla listája jelenjen meg.
Ezzel befejeztük az alkalmazásunk lapjának létrehozását, működik rajta a táblakiválasztás és dinamikus táblamegjelenítés is.



5.10. ábra Feltétel

Most már az elkészült alkalmazásunk hibátlanul működik!

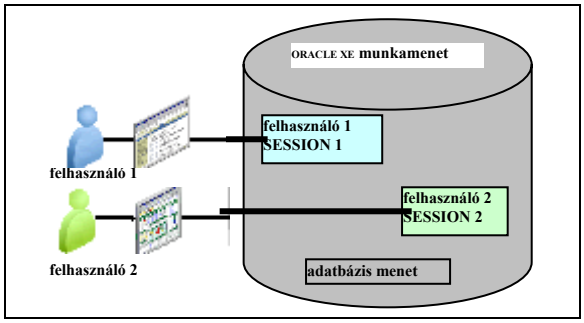


5.11. ábra. *A kiválasztott tábla listája*

Munkamenet (Session State)

A **munkamenet** (`session`) a számítógép-hálózatoknál két számítógép közötti kommunikáció, mely során az egyik (vagy mindkét) gép átmenetileg adatokat tárol a másikról; ennek segítségével egy állapotmentes protokollon keresztül is lehet állapotokat megőrizve kommunikálni.

Amikor a felhasználó belép az Oracle Application Express rendszerbe, akkor megkezd egy munkafolyamatot – ezt nevezzük *munkamenetnek* –, és ez mindaddig tart, amíg a munkát be nem fejezi, azaz ki nem lép. Mivel az OracleApplication Express egy webes eszköz, így a felhasználó a bizalmas adataival az interneten keresztül kommunikál az Oracle Application Express szerverrel. Ezt a folyamatot tehát fokozott védelemmel kell ellátni, és a munkamenet felügyeletének keretében az Oracle APEX erről a védelemről is gondoskodik. A munkamenet elve 5.12. ábrán látható.



5.12. ábra. *Munkamenet vázlata*

Az Oracle Application Express maga alakítja ki, építi fel a munkamenetet, és minden munkamenet egy egyedi azonosító számot kap. Például amikor egy felhasználó elindít egy alkalmazást, kap egy azonosítót. Ez az azonosító szám segíti a ki- és bejelentkezést. Minden bejelentkezéskor, az autentikációnál új azonosítót kap, de minden lap küldése és az adatpostázás is megváltoztatja a munkamenet azonosítóját. Ez a változtatás rugalmasságot és titkosságot biztosít a felhasználónak.

A felhasználó azonosítása két eszközzel történik; az adatbázisban tárolt Munkameneti Rekorddal (Session Record), és a felhasználó gépén tárolt Munkameneti Adatfájllal (Session Cookie).

A Munkameneti Rekord a munka teljes időtartama alatt az adatbázisban található, és még a munka befejezte után is egy ideig. (Így az ügyfél a kilépés után még újabb autentikáció nélkül visszaléphet az alkalmazásába.) A Munkameneti Adatfájl tartalmazza a felhasználó korábbi munkáinak legfontosabb adatait, gyorsítva ezzel az újbóli belépés utáni munkát.

A Munkamenet tárolja a hivatkozható adatokat.

Igy különböző hivtkozásokban , melyeket a session őriz

`SQL` : ElememMax 30 karakter hosszú lehet
: "Elemem"speciális, vagy unicode karakterek esetén, ekkor az e lem nevét Dupla idézőjelbe kell elrejteni

PL/SQL V(ELEM)PL/SQL szintaxisban használjuk az elemre való hivatkozás esetén a `V` függvényt.

PL/SQL `NV(ELEM)` PL/SQL szintaxisban használjuk számokat tartalmazó elem esetén az `NV` függvényt.

StaticText `&ELEM`Statikus szöveg esetén használjuk a konvencionális `&Item_Név` hivatkozást.

Speciális karakterek esetén

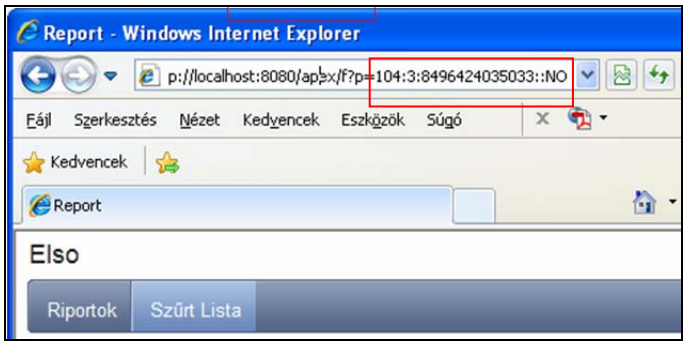
&"ELEM" A hivatkozás speciális karakter, unikode esetén Dupla idézőjel (" ")közé kell rejtenünk

Amikor a felhasználó Elküld feldolgozásra egy lapot (submit) , akkor az Application Express motor automatikusan eltárolja az elemek értékét a session state-ben.

Például: Van egy két lapos alkalmazásunk.Az első lap tartalmazza a formon a felhasználó által beírt egy telefonszámot. Ezt a elemet elnevezte a felhasználó `P2_telefon`-nak. A második lapon meg szertnénk jeleníteni ezt a telefonszámot. Amikor a lap végrehajtásra kerül az Application Express foglyul ejti ezt az általunk beírt telefonszámot, és eltárolja. A felhasználó által beírt telefonszám olvasható a session state által, amely az elemre hivatkozva társul a lap mezejéhez és megjeleníti azt.

A Munkamenet törölhető. A fejlesztés közben azt találhatjuk, hogy a lapon specifikált elemek, az alkalmazás minden lapja, vagy az aktuális felhasználói munkamenet törölhető (clear).Egy Cached érték törlése után értéke null lesz.

A munkamenettel kapcsolatban számos információ magában az alkalmazás URL-Címében található(5.13. ábra).



5.13. ábra. *A munkamenet információi a címben*

Az URL-cím felépítése:

```
http://LocalHost:Port/APEX/HomeLink
```

ahol a `http://LocalHost:Port/APEX` egy IP-cím, mely azonosítja az adatbázis-szervert, a `HomeLink` pedig magát az alkalmazást. A `HomeLink` felépítése:

```
f?p=App:Page:Session
```

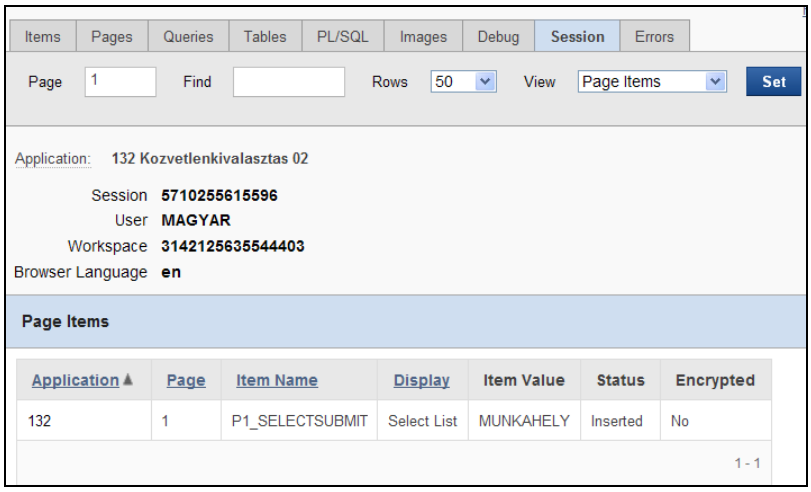
ahol a kettőspontok (`:`) az egyes szintaktikai elmeket választják el, és például a

```
f?p= 104:3:8496424035033::NO:::
```

esetben `104` jelöli az alkalmazás (`App`) azonosítóját, `3` a lapszámot (`Page`), és a munkamenet (`Session`) azonosítóját pedig a `8496424035033` számsorozat.

A munkamenet azonosítóját még további öt paraméter követheti (melyek hiánya esetén is az elválasztó kettőspontok szerepelnek). Ezek a `Request`, a `Debug`, a `ClearCache`, az `itemNames`, az `itemValues` és a `PrinterFriendly`, ahol

`Request`	a lapváltási eseményt kiváltó gomb azonosítója (`Button Request`),
`Debug`	`NO` vagy `YES` a hibakövetési információ megjelenítési mód (`NO` esetén nem jelenik meg hibakövetési információ),
`ClearCache`	megadja hogy melyik oldal gyorsmemóriáját törli,
`itemNames`	az olyan elemnevek vesszővel elválasztott listája, amelyek beállítják a munkamenet állapotát,
`itemValues`	az előzőekben megadott elemnevekhez tartozó értékek vesszővel elválasztott listája,
`PrinterFriendly`	az adott lap nyomtatásának engedélyezése (`YES`, vagy `NO`).



5.14. ábra. *A Session állapota*

Ha a fejlesztő eszköztáron a `Session` gombra kattintunk, minden lényeges információt megtalálhatunk a lapon.(5.14. ábra)

Felül található egy menüsor, amely igen érdekes dolgokra világít rá. megőrzi a hibákat, stb...

Kiszámító Folyamatok (Computations)

A Kiszámító Folyamat (`Computation`) egy számítási eljárást rendel hozzá egy (és csak egy) Elemhez (`Item`). Gyakran használjuk oly módon, hogy egy Elem típusú gombbal aktivizálunk egy Kiszámító Folyamatot, melynek eredménye például egy szövegmezőben megjelenő érték.

Létrehozhatunk Lapszintű Kiszámító Folyamatot, mely a lap megjelenéséhez kapcsolódik (`Page Rendering`), és Alkalmazásszintű Kiszámító Folyamatot, amely a lap feldolgozása során (az alkalmazás logikájához kapcsolódóan) hajtódik végre (`Page Processing`).

Az Alkalmazás szintű Kiszámító Folyamat egy logokai egység egy egyszerű lap vagy alkalmazás-szintű elem értékét állítja be, és ezek a beállított pontok az alkalmazás több pontján futnak keresztül.

Mina Lapszintű mind az Alkalmazás-szintű kiszámító Folyamat alapiulhat SQL, lekérdezésen, PL/SQL programon, elem- vagy statikus értéken.

Ez utóbbi eset úgy állítható be, hogy a Kiszámító Folyamat létrehozásakor, vagy szerkesztésekor a `Computational Point` mezőt `After Submit` értékűre választjuk. Ekkor a Kiszámító Folyamat a Lap-Definíciós (`Page Definition`) ablak Folyamat-Definíciós csoportjában (`Page Processing`) jelenik meg. Minden egyéb értékválasztás esetén a Kiszámító Folyamat lapszintű lesz, aminek megfelelően a Lap-Megjelenítési (`Page Rendering`) csoportban jelenik meg.

Alkalmazászintű Kiszámító Folyamat

Az Alkalmazásszintű Kiszámító Folyamat általános használata, hogy az utolsó látható lap értékét tárolja. Egy elem értékének eltárolásával a visszatérő gombhoz adható, és azokat a felhasználó átirányíthatja a Kiszámító Folyamat által zsákmányolt lapszámra.

Ez a folyamat igen jól dolgozik , abban az esetben amikor az engedélyezett felhsználónak szüksége van egy hibás lap utáni visszatérésre.

Példa

Az utolsó lap számát tároljuk el egy elemben, ennek neve legyen `UTOLSO_LAP`

A hozzá tartozó Lapszintű Kiszámító Folyamat programtörzse pedig:

```
BEGIN
  :UTOLSO_LAP := NVL(:AKTUALIS_LAP, :APP_PAGE_ID);
  :AKTUALI_LAP := :APP_PAGE_ID;
  :RETURN :UTOLSO_LAP;
END;
```

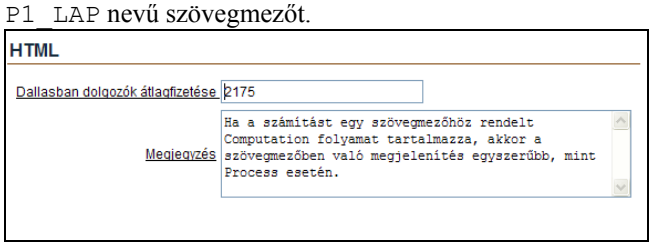
Létrehozása a Megosztott (Shared)komponensben történik.

Lapszintű Kiszámító Folyamat létrehozása

Példa

Számítsuk ki a `dolgozo` tábla részlegenkénti átlagát, és írjuk ki egy szövegmezőbe (5.15.ábra).

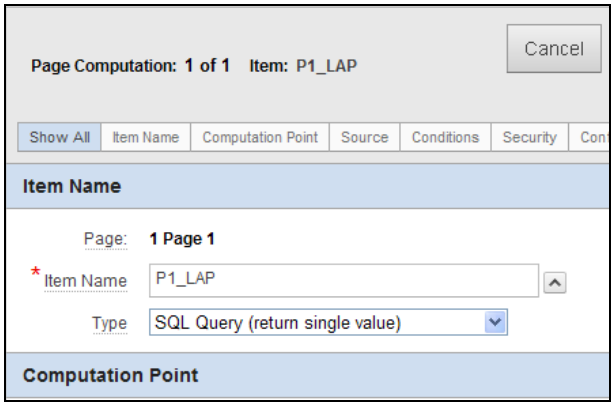
1. Vegyünk fel egy lapot és egy `Dallasban dolgozók átlagfizetése` feliratú `P1_LAP` nevű szövegmezőt.



5.15.ábra. *A megjelenő felület*

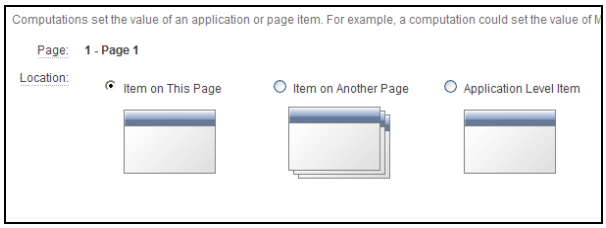
2. Kiszámító Folyamat létrehozása

Kattintsunk a Kiszámító Folyamat (`Computation`) létrehozó ikonjára (5.16.ábra).



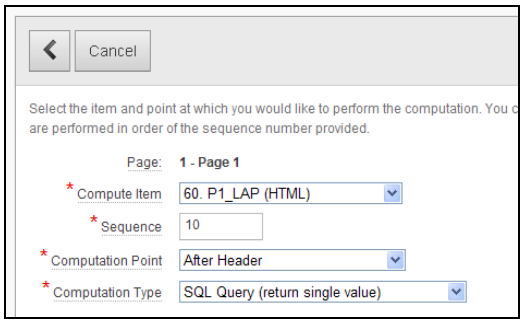
5.16. ábra. *Kiszámító folyamat létrehozása*

A varázsló első ablakában válasszuk ki, hogy a Kiszámító Folyamat eredményét megjelenítő szövegmező az aktuális lapon legyen.(5.17. ábra)



5.17. ábra A Kiszámító Folyamat helye

A következő ablakban rendeljük a Kiszámító Folyamathoz (`Compute Item`) az eredményét megjelenítő szövegmező nevét ,a Kiszámító Folyamat végrehajtási pontját (`Computation Point`), végül a Kiszámító Folyamat (`Computation Type`) típusát (5.18.ábra)



5.18. ábra *Az elem, belépési pont stb kitöltése*

A végrehajtási pontok (`Computation Point`) az alábbiak lehetnek:

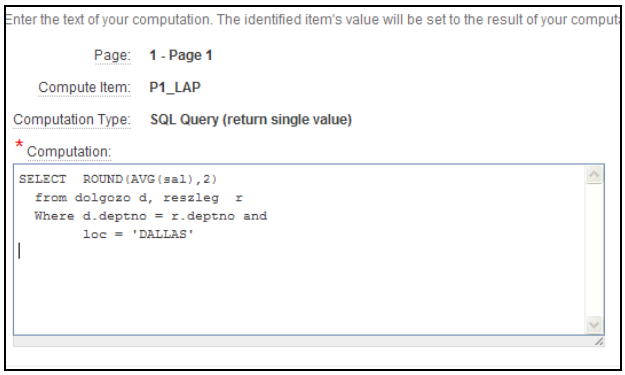
`Before Header`: A Kiszámító Folyamat végrehajtódik a lap megjelenítése előtt.

`After Header`	A közvetlenül a fejléc megjelenése után.
`Before Region(s)`	A Kiszámító Folyamat végrehajtódik mielőtt a tartomány megjelenik
`After Region(s)`	A Kiszámító Folyamat végrehajtódik miután a tartomány megjelenik
`Before Footer`	A Kiszámító Folyamat végrehajtódik a tartomány megjelenése után, de a lábléc megjelenése előtt
`After Footer`	A Kiszámító Folyamat végrehajtódik a tartomány után, de a lábléc megjelenése után.
`After Submit`	A Kiszámító Folyamat végrehajtódik a lap feldolgozása után.

A Kiszámító Folyamat (`Computation Type`) típusai:

`Static Assignment:`	Konstans érték hozzárendelése
`PL/SQL Function Body:`	PL/SQL függvény törzs
`SQL Query(return single value):`	SQL lekérdezés (egyszerű érték tér vissza)
`SQL Query(return colon separeted value)`	SQL lekérdezés (visszatér vesszővel elválasztott értékkel
`SQL Expression:`	SQL kifejezés
`PL/SQL Expression:`	PL/SQL kifejezés
`Item Value:`	Elem visszatérési értéke

A továbblépés után írjuk meg a lekérdezést a Kiszámítás (`Computation`) szövegmezőben, mely a `List` címkéjű (és `P1_List` nevű) legördülő mező visszatérési értékére (a kiválasztott részleg azonosítójára) hivatkozik: (5.19. ábra)



5.19. ábra. *A Számítás SQL utasítása*

3. A `Next` továbblépés után megadhatnánk valamilyen megjelenítési feltételt (mi most nem adunk meg). Ezekután hozzuk létre a Kiszámító Folyamatot (`Create Computation`), majd futtassuk le az elkészült alkalmazást (ábra).

   Megjegyezzük, hogy azok a folyamatok, melyeknél a végrehajtási pont a lap-feldolgozás utáni (`After Submit`), a Lap-szerkesztő (`Page`) ablak középső oszlopában (`Page Processing`) jelennek meg a `Computations` szekcióban.

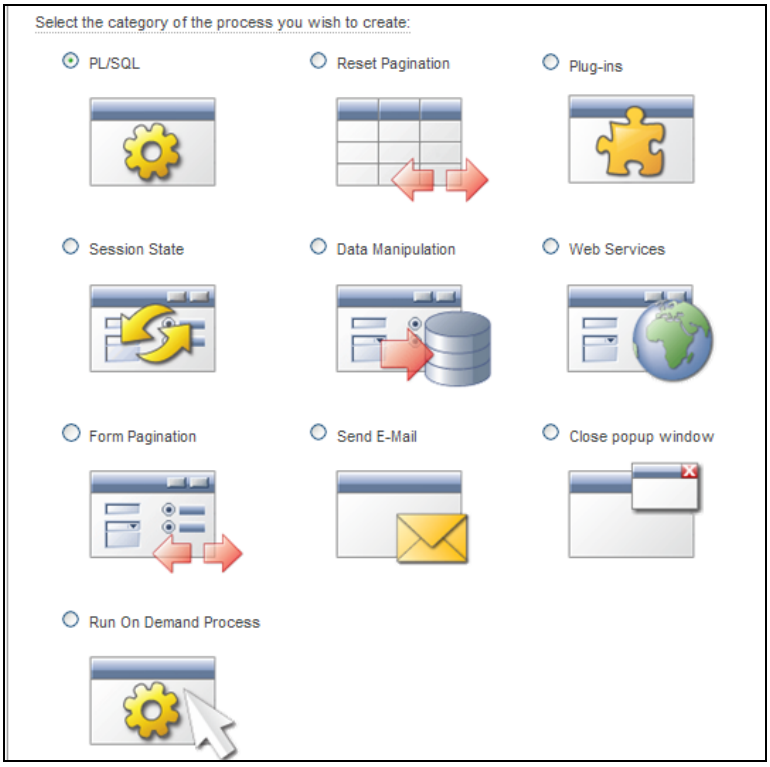
Általános Folyamatok (Processes)

Az Általános Folyamat (vagy röviden Folyamatok) egy olyan PL/SQL nyelven írt eljárás, mely valamilyen eseményre aktivizálódik. Ilyen esemény lehet egy gombra való kattintás, egy kiválasztás, egy lap betöltése, vagy átirányítása, egy munkafolyamat visszaállítása, vagy egy táblabeli sor automatikus feldolgozása.

Az Általános Folyamatok is kétfélék lehetnek, Lapszintűek (`Page Rendering`), melyek a lap megjelenéséhez kapcsolódnak, és Alkalmazásszintűek (`Page Processing`), melyek a már megjelent lapon egy esemény bekövetkezésére végzik a lap feldolgozását az alkalmazás logikájának megfelelően.

Folyamatokat tehát létrehozhatunk mind a Lap-Megjelenítési csoportban (`Page Rendering`), mind a Folyamat-Szerkesztő csoportban (`Page Processing`).valamint a Megosztott (`Shared`) komponensek eszköz segítségével Az, hogy végül is hol fogjuk látni, a létrehozási paraméterek beállításától függ. A létrehozást a `Process` szekció `Create` (+ jelű) ikonnal kell elindítani. A létrehozás során először a folyamat kategória típusát kell kiválasztanunk.

A Folyamatok a következő típusúak lehetnek (5.20. ábra):



5.20. ábra Választható Folyamatok

`PL/SQL`

Egy általunk létrehozott PL/SQL programot lefuttat. Ezt a folyamat típust használjuk, amikor egy PL/SQL blokkból indítunk egy folyamatot, vagy egy API eljárást.

`Reset Pagination`

Visszaállítja a riport tartományban az eredményhalmazt. Az APEX motor nyomon követi a felhasználó tevékenységeit a riporton , és ez a kategória visszaállítja a kezdeti eredményhalmazt.

`Plug ins`

Beépülő modul: megengedi a fejlesztést deklarativ módon kiterjesztett APEX által elérhető modulokkal. Ez az opció akkor érhető el, ha plug-in típusú folyamat létezik az alkalmazásban

`Session State`

A munkamenet elem értékét inicializálja (nullára állítja). E folyamat választása az alkalmazási munkamenetek vagy elemek gyorsítótárát (cache) törli, valamint törli a felhasználói beállításokat.

`Data Manipulation`

A varázsló által gyakran használt adatmódosítási folyamatok:

- Létrehozás (`Create`) érdekében automatikus sor lehívás (`Automatic Row Fetch`), és automatikus sor folyamat (`Automatic Row Process`).
- A táblázatos formában (`tabular form`) több sor módosítása (`Multi Row Update`) és több sor törlése (`Multi Row Delete`)
- Új sor hozzákapcsolása a táblázatos formához (`Add Row to Tabular Form`)

`Close Popup Window`

Ennek végrehajtása során bezárja a felugró (popup) ablakot és a hívó ablakot frissíti.

`Form Pagination`

Lapozást valósít meg a master-detail form és a detail rekordok között.

`Send E-Mail`

Deklarált interface-t szolgáltat `APEX_MAIL.SEND_MAIL` számára. Ez a folyamat típus támogatja a csatolást.

`Web Service`

A lapon folyamatként valósít meg egy web szervert.

`Run On Demand Process`

Létrehoz egy alkalmazás szintű folyamatot, amely csak akkor hajtódik végre, amikor egy speciális lapot hív. Ha ezt Lapszintű Folyamatnál hozzuk létre, akkor egy referenciát adhatunk meg egy már létező Alkalmazásszintű Folyamathoz.

Figyeljünk arra, hogy a PL/SQL programot lezáró `end`; után üres sor ne legyen !!

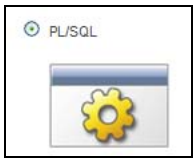
PÉLDA (ÁLTALÁNOS FOLYAMATOKRA)

Készítsük el az előző feladat ellenőrzését Folyamat (`Process`) segítségével (azaz jelenítsük meg az adott részleg dolgozóinak átlagfiztését).

1. Hozzunk létre egy új oldalt másolás segítségével.

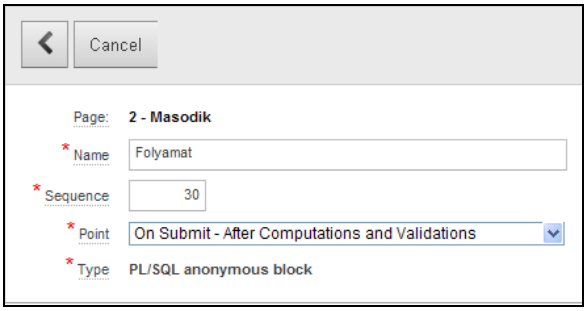
   Lépjünk be a meglévő l. lap LapSzerkesztő ablakába, és a Lap-szerkesztő soron kattintsunk `Create Page` gombra.
2. Vegyünk fel különböző szövegmezőket a lapra, így például két szövegmezőt (`Text-Field`) amely egyike legyen csak olvasható (`Read Only`) beállítása (`Always`), és egy szövegdobozt (`TextArea`).
3. Hozzuk létre a folyamatot varázsló segítségével egy folyamatot. Kattintsunk a Lapszerkszés (`Page Definition`) ablak, Lap-Folyamatok szekció Folyamatok (`Pro-`

`cesses`) szekciójában a Létrehozás (`Create`) ikonjára. Ebben az ablakban válasszuk a PL/SQL folyamatot (5.13. ábra), és lépjünk tovább (`Next`).



5.21. ábra. *PL/SQL választása*

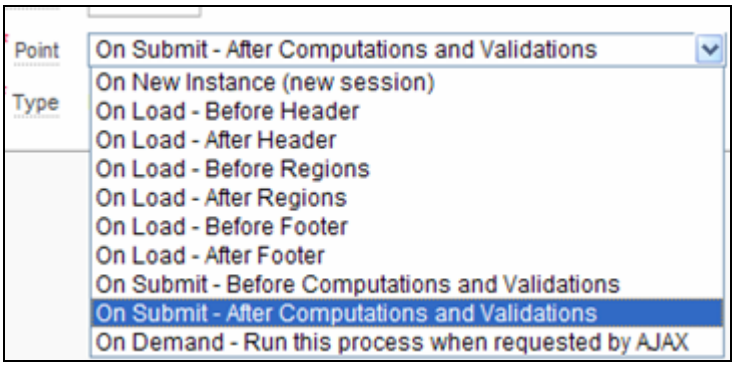
A megjelenő ablakban (5.22. ábra) adjuk meg az Folyamat nevét (`Name`), és a végrehajtási pontját (`Point`), amely legyen feldolgozás utáni (`On Submit-After Computations`...), mivel egy gomb lenyomására írjuk ki az átlagot.



5.22. ábra. *Oldali Folyamat paramétereinek beállítása*

Megjegyzés

A folyamat belépési pontja (`Point`) a következő lehet:



5.23. ábra A folyamat belépési pontjai

`On New Instance.(newsession)`
: Egy új objektum esetén a hitelesítés után hajtja végre a Folyamatot

`On Load - Before Header`
: Betöltéskor a fejléc előtt hajtja végre a Folyamatot

`On Load - After Header`
: Betöltéskor a fejléc után hajtja végre a Folyamatot

`On Load - Before Regions`
: Betöltéskor a tartomány megjelenése előtt hajtja végre a Folyamatot

`On Load - After Regions`
: Betöltéskor a tartomány megjelenése után hajtja végre a Folyamatot

`On Load - Before Footer`
: Betöltéskor a lábléc megjelenése előtt hajtja végre a Folyamatot

`On Load - After Footer`
: Betöltéskor a lábléc megjelenése előtt hajtja végre a Folyamatot

`On Submit - Before Computations and Validations`
: Végrehajtáskor, de a kiszámító folyamat és értékellenőrzés előtt hajtja végre a Folyamatot

`On Submit - After Computations and Validations`
: Végrehajtáskor , de a kiszámító folyamat és értékellenőrzés után hajtja végre a Folyamatot

`Deprecated - Before Showing page items`
: helytelenített, az oldali elemek megjelenése előtt hajtja végre a Folyamatot

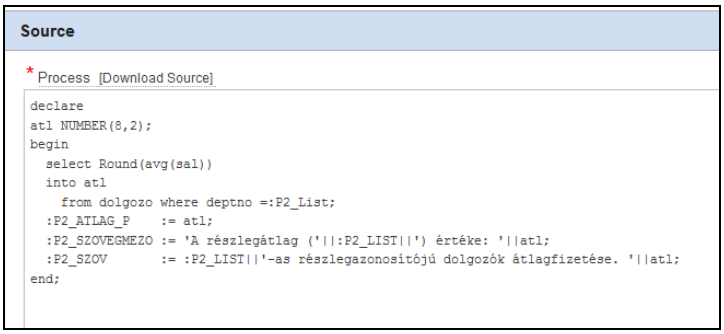
`On Demand - Run this process when requested by AJAX`
: Ez a folyamat akkor fut le , ha az AJAX kéri

Továbblépés után, írjuk meg a PL/SQL programot, majd menjünk tovább (`Next`).

A dolgozók átlagát az alkalmazás második lapján többféle kiírási módban is bemutatjuk, a kiszámítás egy folyamat nevű processben történik (5.24. ábra).

Kiírási módok:

- `P2_ATLAG_P`-Szövegmezőben -`Text Field`
- `P2_szovegmezo`Szövegdoboz - `Text Area`)
- `P2_szov`- csak megjelentetés (`Display Only`)

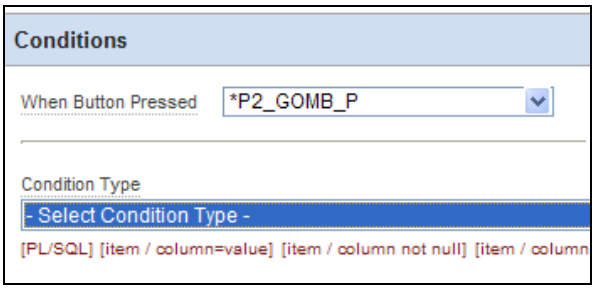


5.24. ábra. *A Folyamat PL/SQL programja*

Most már írjunk sikeres és sikertelen végrehajtás esetén hibaüzenet is (5.16. ábra). Ezek után haladjunk tovább (`Next`).

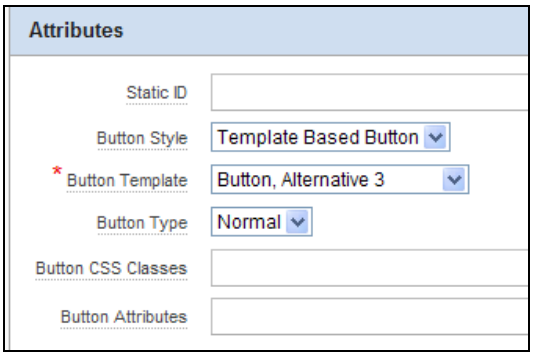
5.25. *Folyamat hibaüzenetei*

Az utolsó ablakban adjuk meg a folyamat végrehajtási feltételét, azaz válasszuk ki melyik gomb lenyomására indul el a folyamat.(5.26. ábra).



5.26. ábra *Feltétel*

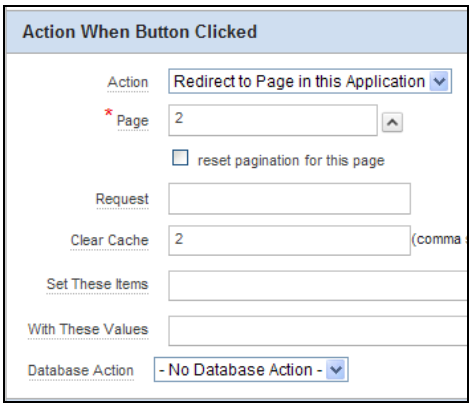
4. Teljesség kedvéért tegyünk a lapra egy `Törli a szövegmezőket` címkéjű nem Elem típusú gombot, amely kitörli az összes szövegmező tartalmát, ha egy másik részleg dolgozóinak átlagát szeretnénk megjeleníteni.
   A létrehozott gombot tulajdonságait adjuk meg



5.27.ábra Törlő gomb tulajdonságai

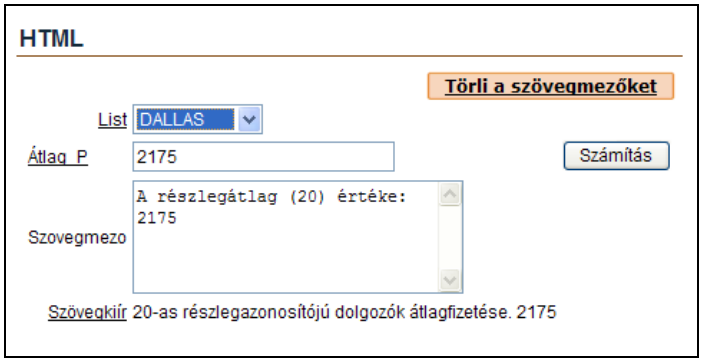
A gomb stílusa: a létező minták közül legyen `Template Based Button`
A gomb mintája `Alternative 3`

Amikor erre a gombra kattintunk akkor térjen vissza a másokid lapra azért, hogy minden szövegmezőt (amibe írtunk) töröljön (5.28. ábra).



5.28. ábra. *Törlő gomb tevékenysége*

Futtassuk le a lapot (5.29.ábra)



5.29.. ábra. *A futtatási lap*

Láthatjuk, hogy a folyamat esetében az eredményt általánosan, több helyen is felhasználhatjuk, kiírásunk egyértelműbb.

COMPUTATION ÉS PROCESS PÉLDA

Készítsen Apex alkalmazást, egy füleket NEM tartalmazó oldallal és kivételkezeléssel a következőképpen:

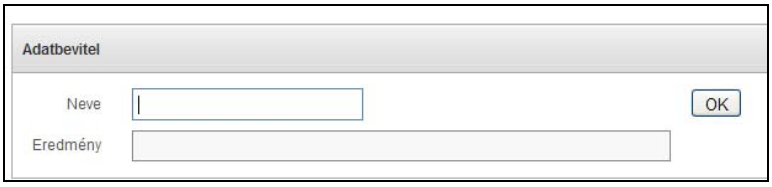
Hozzon létre az `emp` táblából egy `dolgozo` nevű táblát. Ebben a felhasználó által megadott nevű dolgozó fizetését emelje meg 500 USD-ral, ha van nullánál nagyobb jutaléka, egyébként pedig 200 USD-ral. Végül a beavatkozás eredményét jelenítse meg egymás mellett; Név, Régi fizetés, Új fizetés szerkezetben.

A fenti feladatot oly módon oldja meg, hogy

1.) a dolgozó nevét egy szövegmezőbe kérje be,

2.) a feldolgozást egy OK feliratú gombbal lehessen indítani,

3.) egy Általános Folyamattal (`process`) végezze el a beírt nevű dolgozó kiválasztását, és azonosítójának feljegyzését egy rejtett mezőbe tárolja,
4.) egy Kiszámító Folyamattal (`computation`) végezze el a fizetésemelés meghatározását az azonosítójával kijelölt dolgozóra vonatkozóan, és az eredményt egy szövegmezőben jelenítse meg,
5.) hibás adatbevitel esetén az eredménymezőbe írassa ki, hogy "A bevitt adat hibás!".



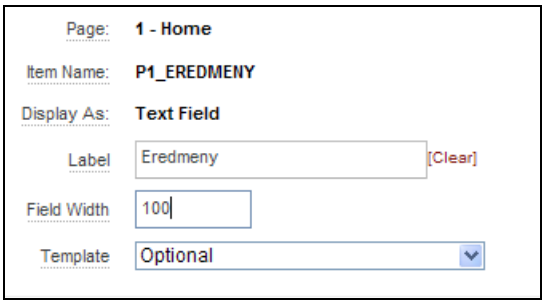
5.30. ábra *A Kalkula program felületet*

Megjegyzés

E példa bemutatja, hogy mit is jelent Általános Folyamat esetén az
`On Submit - Before Computations and Validations`
folyamat belépési pont. Ebben az esetben ugyanis ha létezik Kiszámító Folyamat, akkor az Általános Folyamat előbb hajtódik végre, mint a Kiszámító Folyamat.

Megoldás

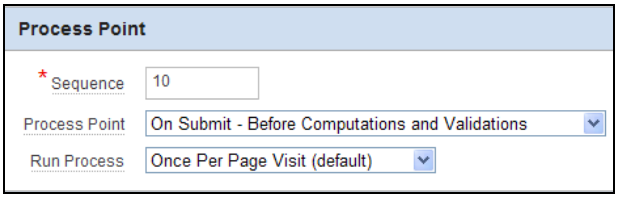
- Hozzunk létre egy új alkalmazást, egy füleket nem tartalmazó oldallal. Ez már ismert.
- Az oldalon legyen egy HTML tartomány, amelyen majd elhelyezzük az elemeket és gombot, amely tartomány neve `Adatbevitel`.
- Vegyünk fel egy szövegmezőt, amelybe a dolgozó nevét írja be majd a felhasználó. Ennek neve legyen `P1_NEVE`. Ennek a dolgozónak a fizetését szeretnénk megemelni, az adott feltételek alapján.
- Vegyünk fel egy rejtett szövegmezőt, neve legyen `P1_ADAT`: Ez a mező veszi fel a Folyamat (`process`) végrehajtása után a keresett dolgozó azonosítóját, és tárolja el, tulajdonképpen egy globális változó lesz.
- Vegyünk fel egy Elem típusú gombot, neve `P1_OK`, a felirata `OK`, azonosítója is `OK`.
- Vegyünk fel egy hosszabb szövegmezőt, melynek neve legyen `P1_EREDMENY`. Ide írjuk majd be az előre megadott forma szerint a dolgozó nevét, régi és új fizetését 5.31. ábra) `Field Width :100`.



5.31. ábra *Mező hosszának beállítása*

- Hozzunk létre egy folyamatot (`process`). A processz adatai a következők (5.32. ábra):

  Belépési pont a `Before Computation and Validation` ,.
  4



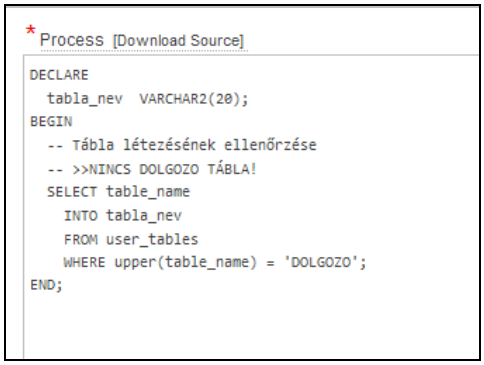
5.32. ábra. *Folyamat létrehozásának első lépése*

Neve: `Tablaellenorzse`,

Pontja : `On Submit - Before Computations and Validations`

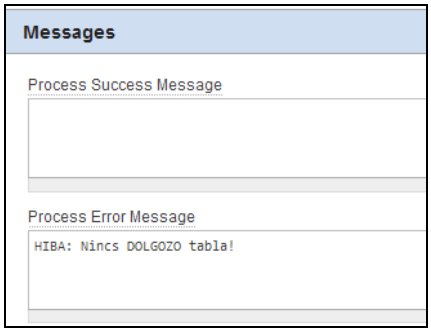
mivel előbb átvesszük a felhasználó által beírt nevet, ellenőrizzük a tábla meglétét, és csak ezután számoljuk ki az átlagot, amelyet majd elemhez rendelünk.

Írjuk meg a PL/SQL folyamatot (5.33. ábra).



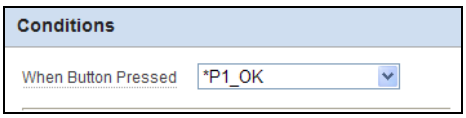
5.33. ábra. *A Folymat PL/SQL programja*

Ugyanitt írjuk meg a sikeres és sikertelen végrehajtás üzeneteit (5.34. ábra).



5.34.ábra *A megjelenő hibaüzenet*.

A feltételnél állítsuk be a gomb feltételt (5.35. ábra).



5.35 ábra. *Hivatkozás beállítása*

Kattintsunk a `Create Process` gomb.
Futtassuk le (5.36. ábra).

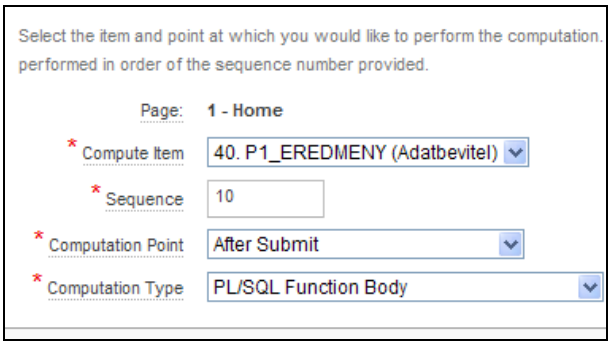
5.36. ábra. *Futtatás*

Ez a folyamat tehát működik, azaz megtalálja és eltárolja a keresett dolgozó azonosítóját. (Itt a PL/SQL folyamatban nem érvényesül az exception hibaüzenet kírása.)

Ezek után hozzuk létre a feltételeket és növelést tartalmazó Kiszámító Folyamatot.

- A Alkalmazásszintű Kiszámító Folyamat létrehozása varázsló segítségével.

  Válasszuk először a `Create`, majd az `Item on This Page` menüpontot, ezután pedig állítsuk be, hogy a Kiszámító Folyamat melyik elemhez tartozzon (ez legyen a `P1_EREDMENY`), majd a belépési pontját (`After Submit`), és a típusát (`PL/SQL Function Body`) (5.37. ábra).



5.37. ábra. *Kiszámító folyamat létrehozása*

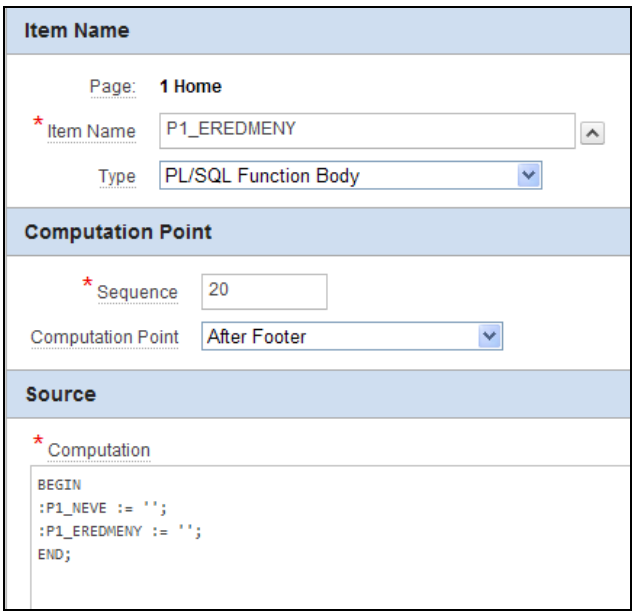
Kattintsunk a `Next` gombra, és írjuk meg a Kiszámító Folyamatot, amely a következő:

```
DECLARE
  Old_Sal      INTEGER;
  New_Sal      INTEGER;
BEGIN
 -- A bevitt adat létezésének ellenőrzése, rejtett elem feltöltése vele
```

```
  -- >>A bevitt adat hibás!
  SELECT empno
   INTO :P1_ADAT
   FROM dolgozo
   WHERE UPPER(ename) = UPPER(:P1_NEVE);

  -- Az eredeti fizetés meghatározása
  SELECT sal
    INTO Old_Sal
    FROM dolgozo
    WHERE empno = :P1_ADAT;

 -- Az új fizetés érték meghatározása
  :P1_Eredmeny := 'Név = '||Upper(:P1_Neve)||',  Régi fizetés =
'||Old_Sal;
  UPDATE dolgozo
    SET sal = sal +
              (SELECT CASE
                        WHEN (comm IS NOT NULL) AND (comm > 0)
                        THEN 500
                        ELSE 200
                      END
                FROM dolgozo
                WHERE empno = :P1_ADAT)
    WHERE empno = :P1_ADAT;

  -- Az új fizetés érték kiiratása
  SELECT sal
    INTO New_Sal
    FROM dolgozo
    WHERE empno = :P1_ADAT;
  :P1_Eredmeny := :P1_Eredmeny||',  Új fizetés = '||New_Sal;
  Return :P1_Eredmeny;

EXCEPTION
   WHEN no_data_found
    then
      :P1_EREDMENY := 'A bevitt adat hibás!';
      RETURN :P1_EREDMENY;
END;
```

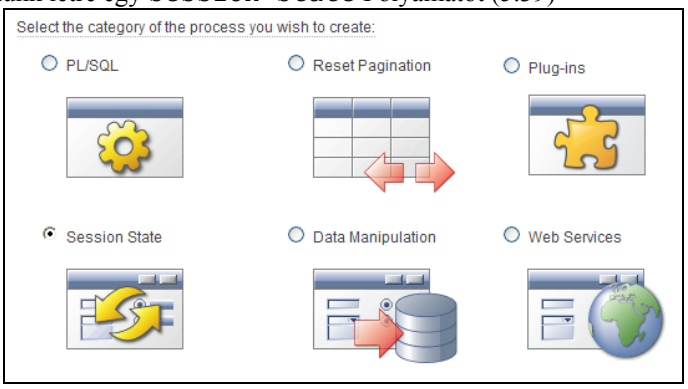
Kiszámító Folymat esetén nem létezik Beépített hibaüzenet, itt nekünk kellett a kivételkezelést megoldani.
Feltételt nem adunk meg.
Hozzuk létre (`Create`) a Kiszámító Folyamatot.
Inicializáljuk az eredmény mezőt.

1. Készítsünk egy lapszintű Kiszámító Folyamatot (`Computation`) (5.38. ábra)



5.38.ábra *A kiszámító folyamat*

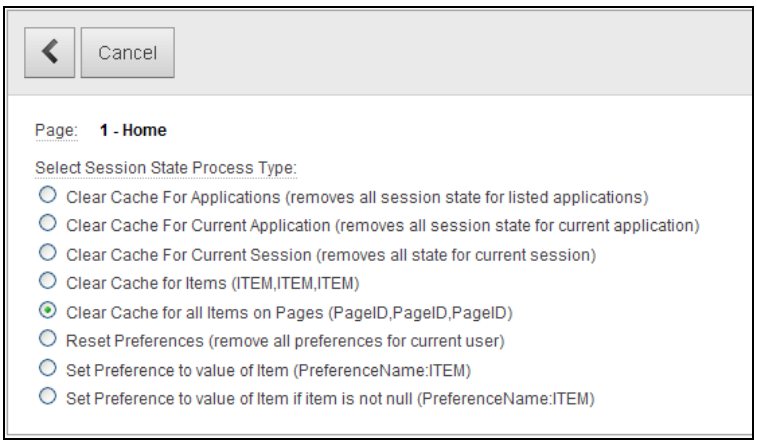
2. Hozzunk létre egy `Session State` Folyamatot (5.39)



5.39. ábra *Session State választás*

`Next`,

`Name: PageInit. Next`,

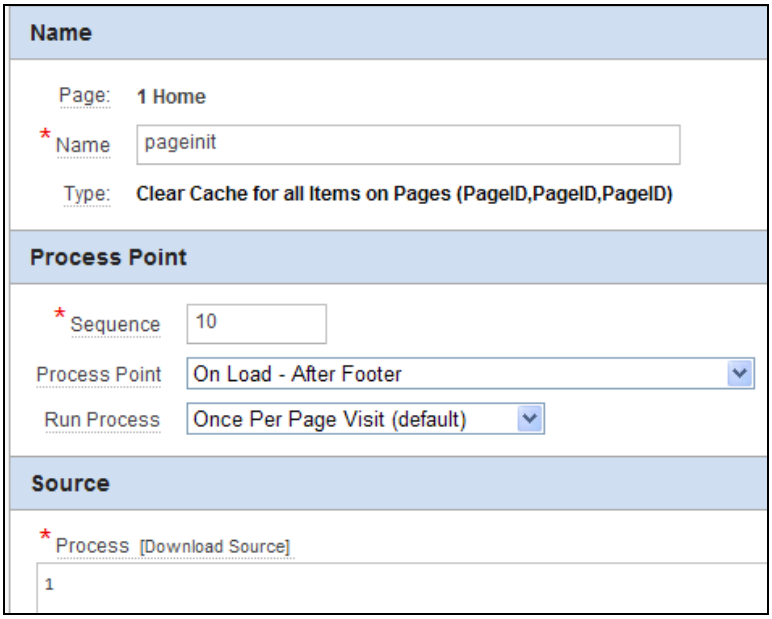


5.40.ábra lapszintű elemek törlése

Válasszuk a tipusának `Cear Cache for all items on pages(PageID,PageID,...)` rádiógombot.

`Next.`

A következő ablakot (5.41. ábra) töltsük ki



5.41. ábra KAz ablakok kitöltése

`Process Point`: legyen `On Load-After Footer` Next,

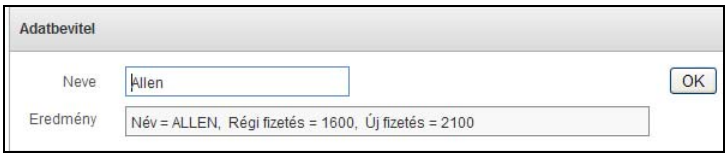
`Source: az 1.` lap

A többi ablakban nincs tennivalónk, tehát `Create Process`.

Végül indítsuk el az alkalmazást (5.42. ábra).

5.42. ábra. A Kiszámító folyamat eredménye

Ezek után írjuk be `Allen` nevű dolgozót, akinek fizetése `1600`, a jutaléka `300`, és mint fent láthatjuk, most az új fizetése `2100` lesz (5.43.ábra).



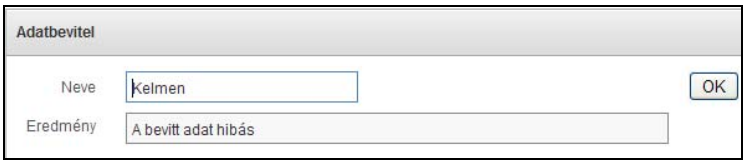
5.43 Ellenőrzés Allen dolgozóval

Következőként írjuk be a `Turner` nevű dolgozót t, akinek `0` a jutaléka (5.44. ábra):

5.44. ábra. *Akinek jutaléka 0 USD, annak új fizetése 200 USD-vel több*

Láthatjuk, hogy `Turner` fizetése csak 200 USD-ral emelkedett, tehát alkalmazásunk jól működik. Megjegyezzük, hogy mivel a feltétel `(comm IS NOT NULL) and (comm>0)` volt, így a `Null`-értékű és a nulla-értékű jutalékokra a program azonosan viselkedik.

Írjunk be nem létező nevet, akkor az eredménymezőben hibaüzenet fog megjelenni (5.45. ábra).



5.45. ábra. *Hibás név megadása esetén a hibaüzenet megjelenik*

6. fejezet. SQL utasítások futtatása alkalmazásból

Dinamikus SQL utasítás

A dinamikus SQL segítségével általános célú eljárásokat hozhatunk létre, valamint PL/SQL-ben DDL valamint bizonyos DML utasításokat hajthatunk végre. A dinamikus SQL utasítások fordítási idő helyett futási időben jönnek létre. Használhatjuk névtelen blokkban, tárolt eljárásokban, csomagokban. Az adatbázis az alapértelmezett adatbázistól különböző adatbázis-kezelést biztosító felülettől kapja: hivatkozási változót, mint paramétert. A hivatkozási változót a fejlesztési felülettől kapja, és esetleg oda kell az eredményt visszaadni. A gazdakörnyezeti változóra a dinamikus SQL ben kettősponttal hivatkozunk, a végrehajtó utasításban pedig sztring lehet. A rendszer elemzi a megadott SQL utasítást, az elemzés után előállítja az utasítás végrehajtható formáját, majd az utasítás lefut. A rendszer futás után megsemmisíti az utasítás alapján előállított végrehajtható alakot. Tehát egyszer használatos utasítás esetén célszerű így használni. A többszöri végrehajtás esetén célszerű a `PREPARE` utasítás használatával előkészíteni az `EXECUTE` utasítást.(ezt célszerű nem Oracle fejlesztői környezetben használni)

A dinamikus SQL utasítás beépíthető a programokba úgy, hogy karakteres sztringként tároljuk az utasításokat. A paraméterek pedig futási időben definiálódnak. Fontos megjegyezni, hogy a sztring valódi SQL utasítást vagy PL/SQL kódot tartalmaz.

Dinamikus SQL utasítás az `EXECUTE IMMEDIATE` utasítás.

Általános alakja:

```
EXECUTE IMMEDIATE dinamikus_SQL_sztring
[INTO definiált_változo1, definiált_változo2,...]
[USING [IN|OUT|IN OUT] paraméter1,paraméter2,...]
[RETURNING INTO | RETURN paraméter1,paraméter2,...];
```

Ahol :

- `INTO` záradék azokat az előre definiált változókat tartalmazza, amelyek a `SELECT` utasítás visszatérő adati. (Hasonló, mint a statikus `SELECT INTO` utasítás.)

- A USING záradék a hivatkozási paraméterek listáját tartalmazza, amely lehet 3 féle IN, OUT és IN OUT típusú. Alapértelmezés az IN típus.
- Végül pedig a RETURNING INTO vagy RETURN záradék. Amely az SQL vagy PL/SQL blokk végrehajtása utáni, hivatkozott paraméter eltárolt értékeit tartalmazza. Hasonló a USING záradékhoz.

Amennyiben interaktívan akarjuk használni a dinamikus SQL-t, akkor & -tel hivatkozunk a változóra.

Jobb megoldás, ha paraméterezetten használjuk. Ekkor a paraméterek sorszámát adjuk meg a SELECT utasításban, és a USING záradék után írjuk a paraméter listát, amelynek első eleme a SELECT utasítás hivatkozásában: 1 lesz.

Egyik nagy előnye a dinamikus SQL utasításnak, hogy hozzárendelt változókat használhatunk.

1. Példa

Töröljük ki a dolgozó táblából SCOTT dolgozót.

A nev változónak értékül adjuk a SCOTT értéket, majd ezt az értéket átadjuk paraméterként a DELETE utasításnak a USING utasitásrész segítségével.

```
SET SERVEROUTPUT ON
Declare
 sqlsztring   VARCHAR2(100);
 nev          VARCHAR2(10):='SCOTT';
BEGIN
  -- hozzuk létre a dolgozó táblát
  EXECUTE IMMEDIATE 'CREATE TABLE dolgozo AS SELECT * FROM emp';

  -- Töröljük ki a nev változóban megadott dolgozót
  sqlsztring:= 'DELETE dolgozo WHERE ename = :nev';
--  sqlsztring:= 'DELETE dolgozo where ename = :1';  ez is jó

  EXECUTE IMMEDIATE sqlsztring USING nev;

END;
/
SELECT * FROM dolgozo;
```

Futási eredmény

```
A PL/SQL eljárás sikeresen befejeződött.


EMPNO ENAME      JOB              MGR HIREDATE      SAL    COMM  DEPTNO
----- ---------- --------- ---------- ---------- ------ ------- -------
 7369 SMITH      CLERK           7902 1980-12-17    800              20
 7499 ALLEN      SALESMAN        7698 1981-02-20   1600     300      30
 7521 WARD       SALESMAN        7698 1981-02-22   1250     500      30
 7566 JONES      MANAGER         7839 1981-04-02   2975              20
 7654 MARTIN     SALESMAN        7698 1981-09-28   1250    1400      30
 7698 BLAKE      MANAGER         7839 1981-05-01   2850              30
 7782 CLARK      MANAGER         7839 1981-06-09   2450              10
 7839 KING       PRESIDENT            1981-11-17   5000              10
 7844 TURNER     SALESMAN        7698 1981-09-08   1500       0      30
 7876 ADAMS      CLERK           7788 1987-05-23   1100              20
 7900 JAMES      CLERK           7698 1981-12-03    950              30
```

```
 7902 FORD       ANALYST         7566 1981-12-03   3000              20
 7934 MILLER     CLERK           7782 1982-01-23   1300              10

13 sor kijelölve.
```

2. Példa

a.)

A felhasználó által megadott azonosítójú dolgozó nevét keressük. Oldjuk meg ezt interaktívan.

Megoldás

```
SET SERVEROUTPUT ON
ACCEPT azon PROMPT 'Kérem a keresendő név azonosítóját: '

DECLARE
  nev          VARCHAR2(20);
  azon         NUMBER(4);
  sqlsztring   VARCHAR2(100);
BEGIN
 sqlsztring := 'SELECT ename FROM emp WHERE empno = :1';

 EXECUTE IMMEDIATE sqlsztring
  INTO nev
  USING &azon;
DBMS_OUTPUT.PUT_LINE('A keresett név:  '||nev);
end;
/
Adja meg a(z) azon értékét: 7788
régi  10:   USING &azon;
új  10:   USING 7788;
A keresett név:  SCOTT

A PL/SQL eljárás sikeresen befejeződött.
```

Megjegyzés

Amennyiben az interaktívan megadott változó karakter típusú ügyeljünk a megfelelő aposztrófok használatára.

Pl. Ha a nevet adja meg a felhasználó akkor a

`EXECUTE IMMEDIATE sqlsztring INTO` ***lekérdezett, kiírando változók listája*** **`USING UPPER('&nev')`**

b.)

A felhasználó által megadott azonosítójú dolgozó rekordját szeretnénk látni.

Megoldás

```
PROMPT >>>> második  példa >>>>
ACCEPT azon PROMPT 'Kérem a keresendő rekord azonosítóját: '
DECLARE
  nev           Varchar2(20);
  azon          NUMBER(4);
  foglalkoz     emp.job%type;
  fizetes       emp.sal%TYPE;
  reszleg       emp.deptno%TYPE;
  sqlsztring    VARCHAR2(100);
BEGIN
  sqlsztring := 'SELECT ename, job, sal, deptno '||
```

```
                'FROM emp'                        ||
                'WHERE empno = :1';
  EXECUTE IMMEDIATE sqlsztring
  INTO nev, foglalkoz, fizetes, reszleg
  USING &azon;
  DBMS_OUTPUT.PUT_LINE (' A keresett ember rekordja: ');
  DBMS_OUTPUT.PUT_LINE ('*********************************');
  DBMS_OUTPUT.PUT_LINE (nev       ||'   '||
                        foglalkoz ||'   '||
                        fizetes   ||'   '||
                        reszleg);
END;
/
```

Futási eredmény

```
>>>> második  példa >>>>
Kérem a keresendő rekord azonosítóját: 7654
régi  14:  USING &azon;
új  14:  USING 7654;
A keresett ember rekordja:
********************************
MARTIN   SALESMAN   1250   30

A PL/SQL eljárás sikeresen befejeződött.
```

3. Példa

Tábla eldobás, létrehozás, `SELECT` lekérdezés dinamikus SQL utasítással (DDL).

Hozzunk létre egy `kereskedo` táblát az `emp` tábla `30`-as részlegének dolgozóiból, és írjuk ki a részleg dolgozói létszámát.

Megoldás

```
SET SERVEROUTPUT ON
Declare
  sqlsztring   VARCHAR2(100);
  v_letszam    NUMBER(2);
BEGIN
-- ha már van ilyen táblánk akkor hagyjuk így,
-- ha nincs akkor tegyük megjegyzésbe a következő két utasítást.
  sqlsztring:= 'DROP TABLE kereskedo';
  EXECUTE IMMEDIATE sqlsztring;
  sqlsztring:= 'CREATE TABLE kereskedo AS SELECT * FROM emp WHERE deptno
=30';

  EXECUTE IMMEDIATE sqlsztring;
  EXECUTE IMMEDIATE 'SELECT COUNT(*) FROM kereskedo'
  INTO v_letszam;
  DBMS_OUTPUT.PUT_LINE('A 30-as részleg létszáma: '||v_letszam);
END;
/
SELECT * FROM kereskedo
```

Futási eredmény:

```
A 30-as részleg létszáma: 6

A PL/SQL eljárás sikeresen befejeződött.

EMPNO ENAME      JOB           MGR HIREDATE        SAL    COMM  DEPTNO
----- ---------- --------- ------- ---------- -------- ------- -------
 7499 ALLEN      SALESMAN     7698 1981-02-20     1600     300      30
 7521 WARD       SALESMAN     7698 1981-02-22     1250     500      30
 7654 MARTIN     SALESMAN     7698 1981-09-28     1250    1400      30
 7698 BLAKE      MANAGER      7839 1981-05-01     2850              30
 7844 TURNER     SALESMAN     7698 1981-09-08     1500       0      30
 7900 JAMES      CLERK        7698 1981-12-03      950              30

6 sor kijelölve.
```

Megjegyzés

Hozzárendelt változó adatdefiníciós műveleteknél nem használható !

Módosítás

Az UPDATE utasítás PL/SQL esetén ugyanúgy működik mint az SQL UPDATE utasítás, annyi különbséggel hogy itt a WHERE utasításrészben megadható egy CURRENT OF opció, amely csak a megadott kurzor utoljára lehívott sorát módosítja. (Ilyenkor a kurzorban meg kell adni a FOR UPDATE utasításrészt. Kiegészülhet opcionálisan a RETURNING INTO utasításrésszel. A RETURNING utasításrészben felsoroljuk azokat az adatElemeket (oszlopneveket), amelyek értéke megváltozik az utasítás által módosított sorokban. Csak skalár és REF típusú értéket tudunk így betölteni. Az adatElem PL/SQL változó, vagy hozzárendelt változó lehet. Az INTO záradékban veszi fel ezt az értéket. Ilyenkor az EXECUTE IMMEDIATE utasításnál is használjuk a USING utasításrészben a RETURNING INTO változónév záradékot.

Az utasítás szintaktikája a következő :

```
UPDATE { minősített táblanév | allekérdezés}
  SET {oszlop = {kifejezés| allekérdezés} |DEFAULT },ez ismétlődhet...
  [WHERE {feltétel | CURRENT OF kurzor} ]
  [RETURNING ( oszlopnév1 [,oszlopnév2]...)
    INTO (változónév1 [,változónév2]...) ]
```

4. Példa

Emeljük meg a felhasználó által megadott azonosítójú dolgozó jutalékát 600 $ -ra.

Megoldás

```
SET SERVEROUTPUT ON
ACCEPT azon PROMPT 'Kérem a dolgozó azonosítószámát: '
DECLARE
  sqlsztring   VARCHAR2(100);
  v_azon       NUMBER(4);
  v_jut         NUMBER(4);
BEGIN
  v_azon := &azon;
  sqlsztring := 'UPDATE kereskedo SET comm = 600 WHERE empno = :v_azon '||
                'RETURNING comm INTO :v_jut ';
  EXECUTE IMMEDIATE sqlsztring USING v_azon RETURNING INTO v_jut;
  DBMS_OUTPUT.PUT_LINE ('Az ÚJ JUTALÉK: ' ||v_jut);
END;
/
SELECT * FROM kereskedo;
```

Futási eredmény

```
Kérem a dolgozó azonosítószámát: 7844
régi   6:   v_azon := &azon;
új   6:   v_azon := 7844;
Az ÚJ JUTALÉK: 600

A PL/SQL eljárás sikeresen befejeződött.

EMPNO ENAME      JOB           MGR HIREDATE      SAL   COMM   DEPTNO
----- ---------- --------- ------- ---------- ------ ------ --------
 7499 ALLEN      SALESMAN     7698 1981-02-20   1600    300       30
 7521 WARD       SALESMAN     7698 1981-02-22   1250    500       30
 7654 MARTIN     SALESMAN     7698 1981-09-28   1250   1400       30
 7698 BLAKE      MANAGER      7839 1981-05-01   2850              30
 7844 TURNER     SALESMAN     7698 1981-09-08   1500    600       30
 7900 JAMES      CLERK        7698 1981-12-03    950              30
6 sor kijelölve.
```

OPEN-FOR, FETCH, CLOSE utasítások használata

Dinamikus SQL használata több sort visszaadó SELECT utasítás esetén. E használat elve hasonló a statikus kurzorokéhoz.

Tehát először a kurzorváltozót kell deklarálni, aztán meg kell nyitni a kurzort a belépési pontnál úgy, hogy az első sort adja vissza az eredményhalmaz számára, majd lehívjuk a második sort és így tovább. Végül, ha minden sor részt vett a folyamatban bezárjuk a kurzort.

KURZORVÁLTOZÓK

Az Explicit kurzorhoz hasonlóan a kurzorváltozó is egy halmaz, amely a tábla valamelyik sorának feldolgozására alkalmas. Amíg a statikus kurzor egyetlen a fordításkor ismert SELECT utasítás eredményét tartalmazza, addig a kurzorváltozó dinamikus. Ez azt jelenti, hogy futási időben a kurzorváltozóhoz bármely típuskompatibilis kérdés kapcsolható. A kurzorváltozó használata megfelel a statikus kurzor használatának, tehát deklaráció, megnyitás, egy-egy sor betöltése, majd bezárás.
A kurzorváltozó referencia típusú változó, és mindig a hivatkozott sor címét tartalmazza.
A kurzorváltozó deklarálása két lépésben történik.
Első a kurzorváltozó típus létrehozása, majd ezután a kurzorváltozó deklarálása következik kurzorváltozó típussá.
Kurzorváltozó típus általános alakja :

```
TYPE név IS REF CURSOR [RETURN
{ táblanév | kurzor_név | kurzorváltozó_név|%ROWTYPE | rekordnév%TYPE |
   Rekordtípusnév | kurzorreferenciatipus_név}]
```

Beszélhetünk erős kurzorreferencia típusról, ekkor a RETURN utasítás szerepel (visszatérési értéke egy rekord_típus), vagy gyenge kurzorreferencia típusról, amikor nem szerepel a RETURN utasítás a típusdeklarációban.

Jelen esetben gyenge kurzorreferenciát használunk. A REF CURSOR kulcsszó azt eredményezi, hogy az új típus egy pointer lesz, ami a definiált típusra mutat.
Kurzor megnyitása

```
OPEN kurzorváltozó FOR dinamikus_SQL_sztring
[USING  param1, param2,…]
```

A *kurzorváltozó* egy gyenge REF CURSOR típus, míg a *dinamikus_SQL_sztring* egy olyan sztring, amely tartalmazza a több sort visszaadó lekérdezést.

Az utasításkód kiolvasása (FETCH)

```
FETCH kurzorváltozó
  INTO PL/SQL_válozó1, PL/SQL_változó2,...
EXIT WHEN kurzorváltozó%NOTFOUND;
```

A kurzor bezárása

```
CLOSE kurzorváltozó;
```

5. Példa

Írjuk ki a felhasználó által megadott foglalkozású dolgozók adatait.

*Megoldás**(a)***

```
PROMPT <<< adott foglalkozású dolgozók adatai >>>
```

```
DECLARE
  TYPE dolgozo_kurzor_tipus IS REF CURSOR;
  dolgozo_kurzor    dolgozo_kurzor_tipus;
  v_nev    VARCHAR2(20);
  v_azon   NUMBER(4);
  v_job    emp.job%TYPE:='CLERK';
  v_sal    emp.sal%type;
  v_deptno emp.deptno%Type;
BEGIN
  Open  dolgozo_kurzor FOR 'SELECT empno, ename , sal, deptno FROM emp'||
                           ' WHERE UPPER(job)= :1'
  USING v_job;
  LOOP
     FETCH dolgozo_kurzor  INTO v_azon,v_nev,  v_sal, v_deptno;
     EXIT WHEN dolgozo_kurzor%NOTFOUND;
     DBMS_OUTPUT.PUT_LINE(v_azon||'   '||v_nev||'   '||v_sal||'
'||v_deptno);

  END LOOP;
END;
/
```

Futási eredmény

```
<<< adott foglalkozású dolgozók adatai >>>
7369   SMITH   800   20
7876   ADAMS   1100   20
7900   JAMES   950   30
7934   MILLER   1300   10

A PL/SQL eljárás sikeresen befejeződött.
```

Megoldás ***(b)***

```
PROMPT <<< adott foglalkozású dolgozók adatai >>>
  TYPE dolgozo_kurzor_tipus IS REF CURSOR;
  dolgozo_kurzor    dolgozo_kurzor_tipus;
  v_nev    VARCHAR2(20);
  v_azon   NUMBER(4);
  v_job    emp.job%TYPE;
  v_sal    emp.sal%type;
  v_deptno emp.deptno%Type;
BEGIN
  Open  dolgozo_kurzor FOR 'SELECT empno, ename , sal, deptno FROM emp'||
                           ' WHERE UPPER(job)= :1'
  USING '&v_job';
  LOOP
     FETCH dolgozo_kurzor  INTO v_azon,v_nev,  v_sal, v_deptno;
     EXIT WHEN dolgozo_kurzor%NOTFOUND;
     DBMS_OUTPUT.PUT_LINE(v_azon||'   '||v_nev||'   '||v_sal||'
'||v_deptno);

  END LOOP;
END;
/
```

Futási eredmény

```
<<< adott foglalkozású dolgozók adatai >>>
Adja meg a(z) v_job értékét: CLERK
régi  12:   USING '&v_job';
új  12:   USING 'CLERK';
```

```
7369   SMITH   800   20
7876   ADAMS   1100   20
7900   JAMES   950   30
7934   MILLER   1300   10

A PL/SQL eljárás sikeresen befejeződött.
```

Megoldás (c)

```
PROMPT <<< adott foglalkozású dolgozók adatai >>>
ACCEPT fogl  PROMPT 'Kérem a foglalkozást: '
DECLARE
  TYPE dolgozo_kurzor_tipus IS REF CURSOR;
  dolgozo_kurzor    dolgozo_kurzor_tipus;
  v_nev    VARCHAR2(20);
  v_azon   NUMBER(4);
  v_job    emp.job%TYPE := '&fogl';
  v_sal    emp.sal%type;
  v_deptno emp.deptno%Type;
BEGIN
  Open  dolgozo_kurzor FOR 'SELECT empno, ename , sal, deptno FROM emp'||
                           ' WHERE UPPER(job)= :1'
  USING v_job;
  LOOP
     FETCH dolgozo_kurzor  INTO v_azon,v_nev,  v_sal, v_deptno;
     EXIT WHEN dolgozo_kurzor%NOTFOUND;
     DBMS_OUTPUT.PUT_LINE(v_azon||'   '||v_nev||'   '||v_sal||'
'||v_deptno);

  END LOOP;
END;
/
```

Futási eredmény

```
<<< adott foglalkozású dolgozók adatai >>>
Kérem a foglalkozást: CLERK
régi   6:   v_job    emp.job%TYPE := '&fogl';
új   6:   v_job    emp.job%TYPE := 'CLERK';
7369   SMITH   800   20
7876   ADAMS   1100   20
7900   JAMES   950   30
7934   MILLER   1300   10

A PL/SQL eljárás sikeresen befejeződött.
```

6. Példa

Írjuk ki dinamikus SQL segítségével, hogy az egyes részlegeknek mennyi a dolgozói létszáma

Megoldás

```
DECLARE
  TYPE reszleg_kurzor_tipus  IS REF CURSOR;
  reszleg_kurzor    reszleg_kurzor_tipus;

  sqlsztring    VARCHAR2(300);
  v_reszleg     NUMBER(2);
  v_letszam     NUMBER(2);
  v_sorszamlal  NUMBER(2);
```

```
BEGIN
  sqlsztring :='SELECT deptno, COUNT(*) as letszam'    ||
               ' FROM emp '                            ||
               ' GROUP BY deptno ';
  v_sorszamlal := 0;
  OPEN reszleg_kurzor FOR sqlsztring;
   LOOP
     FETCH reszleg_kurzor INTO v_reszleg, v_letszam;
     EXIT WHEN reszleg_kurzor%NOTFOUND;
    v_sorszamlal := v_sorszamlal +1;
    IF v_sorszamlal  <=10 THEN
  DBMS_OUTPUT.PUT_LINE(v_sorszamlal||'   '|| 'Részlegszám: '||v_reszleg ||
                          '   Létszám: '||v_letszam);
    END IF;
   END LOOP;
  CLOSE reszleg_kurzor;
EXCEPTION
   WHEN OTHERS then
     IF reszleg_kurzor%ISOPEN THEN
        CLOSE reszleg_kurzor;
     END IF;
    DBMS_OUTPUT.PUT_LINE('HIBA: '||SUBSTR(SQLERRM, 1, 200));
END;
/
```

Futási eredmény

```
1   Részlegszám: 30   Létszám: 6
2   Részlegszám: 20   Létszám: 5
3   Részlegszám: 10   Létszám: 3
A PL/SQL eljárás sikeresen befejeződött.
```

7. Példa

A végrehajtandó sztring egy PL/SQL blokkot tartalmaz.
A dinamikus utasítás egy PL/SQL blokk. Irassuk ki az aktuális dátumot. Nagyon figyeljük az aposztrófok megfelelő számára.

Megoldás

```
SET SERVEROUTPUT ON
DECLARE
  plsqlsztring   VARCHAR2(300);
BEGIN
  Plsqlsztring := 'DECLARE '                                  ||
                  '  v_datum  DATE; '                         ||
                  'BEGIN '                                    ||
                  '  SELECT sysdate INTO v_datum FROM dual;'  ||
                  ' DBMS_OUTPUT.PUT_LINE '                    ||
                  ' (''A mai dátum: ''|| TO_CHAR(v_datum,
''YYYY.MM.DD.''));'||
                  'END; ';
  EXECUTE IMMEDIATE Plsqlsztring;
END;
/
```

Futási eredmény

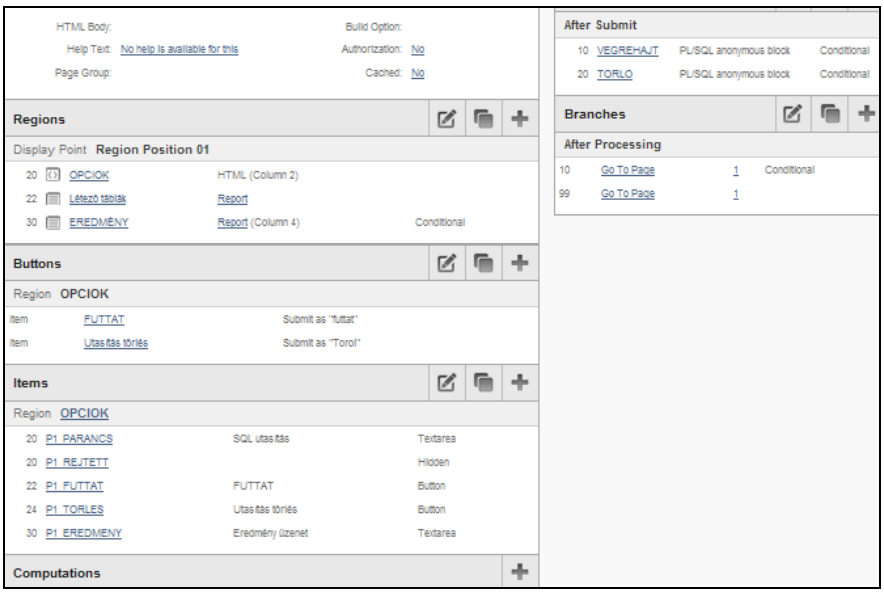
```
A mai dátum: 2010.12.07.

A PL/SQL eljárás sikeresen befejeződött.
```

Parancsfuttatás (DML, DDL)

Példa
Készítsen egy apex-alkalmazást, mely a felhasználó egy szövegbeviteli felületén keresztül beírt tetszőleges SQL utasítást (`SELECT`, és egyéb, például `INSERT`, `CREATE`, `ALTER`, stb.) végrehajt, és a futás eredményét egy külön ablakban meg is jeleníti. Az alkalmazás tartalmazzon hibakezelést, és a benne szereplő feliratok magyar nyelvűek legyenek.

Megoldás

1. A lapon hozzunk létre egy Opciok nevű HTML tartományt. Ezen legyen egy `Text Area` szövegablakot (`P1_PARANCS`) a beirandó parancs részére, és egy `Text Field` szövegmezőt (`P1_EREDMENY`) az eredmény megjelenítésére.
2. Hozzunk létre egy másik szövegmezőt (Text Area) az üzenet kiírására (`P1_EREDMENY`).



6.1 ábra. *A parancsfuttatás lapjának szerkesztő lapja*

3. Tegyünk a lapra egy gombot (`P1_Futtat`), és egy másik gombot törlés az SQL utasítás törlésére (P1_Torles) néven.
4. Hozzunk létre egy eljárást a Folyamat-Definíciós rész Processes szekciójában `Vegrehajtas` néven (6.1. ábra). Ehhez egy rejtett mezőt is fel kellett vennünk, amibe futáskor az SQL lekérdezés első hat szavát tesszük be azért, hogy össze tudjuk hasonlítani a SELECT szóval.

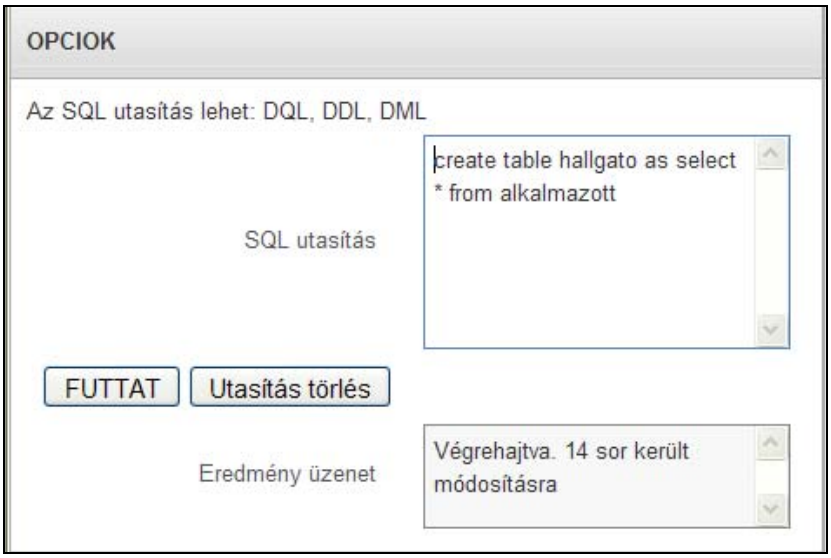
   Szét kell választanunk a select és a DML,DDL, utasításokat. A Select utasításokat meg is jelenítjük, míg az egyéb SQL utasításokat közvetlenül nem.

   A `Vegrehajt` nevű PL/SQL eljárás egy dinamikus SQL utasítást tartalmaz, mely végrehajtja a `P1_PARANCS` szövegablak tartalmát, és egy értékadást, mely a `P1_EREDMENY` szövegmezőbe írja a felhasználó által beírt és végrehajtott SQL utasítás módosított sorainak számát (amelyet az `SQL%ROWCOUNT` ad vissza) Select utasítás esetén azt, hogy "végrehajtva", lásd 6.2. ábra.

```
BEGIN
  :P1_EREDMENY := '';
  :P1_REJTETT  := UPPER(SUBSTR(:P1_PARANCS,1,6));

  EXECUTE IMMEDIATE :P1_PARANCS;

  IF :P1_REJTETT != 'SELECT'
  THEN
    :P1_EREDMENY := 'Végrehajtva. '||SQL%ROWCOUNT||' sor került módosítás-
ra';
  ELSE
    :P1_EREDMENY :=' Végrehajtva.';
  END IF;

EXCEPTION
  WHEN OTHERS
    THEN
```

```
        IF SUBSTR(SQLERRM,1,9)='ORA-06535'
          THEN
            :P1_EREDMENY := 'Nincs utasítás!';
          ELSE
            :P1_EREDMENY := 'Hiba történt! Kód: '||SQLERRM;
          END IF;
        :P1_REJTETT := 'HIBA';
END;
```

Egy DDL parancs végrehajtását látjuk a 6.2. ábrán.



6.2.ábra A program futási képe

Hozzunk létre még egy törlő eljárást, amely kitörli a szövegmezőket és eltünteti az esetleges select listát.

`Torlo` eljárása az `Utasítás törlés` gomb hatására, melynek tartalma:

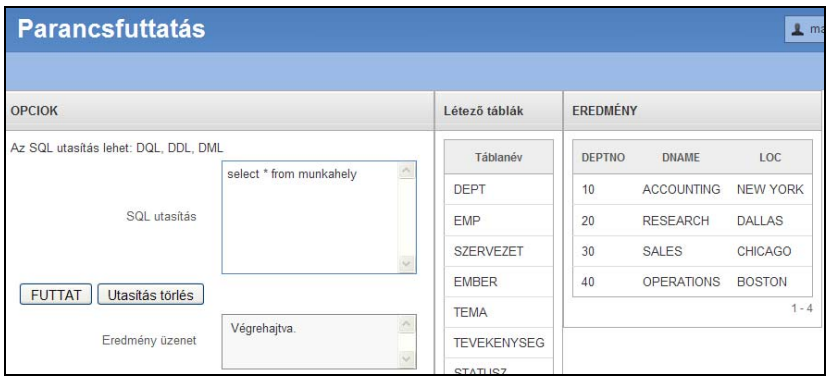
```
begin
  :P1_PARANCS  := ' ';
  :P1_REJTETT  := ' ';
  :P1_EREDMENY := 'Utasításmező törölve!';
end;
```

5.)Segítségül legyen még a lapon a létező táblák listája az opciok tartomány mellett.

A táblalista jelenjen meg egy egyoszlopos riport tartományban, amelynek forrás tartalma kiszűri a mintapélda tábláit, és esetleg az apex fejlesztő adminisztrátor tábláit.

```
select table_name as táblanév from user_tables
where table_name not like '%DEMO%' and
      table_name not like '%APEX$%'
```

6.) Jelenítsük meg a select lekérdezés eredményét is, mellette.



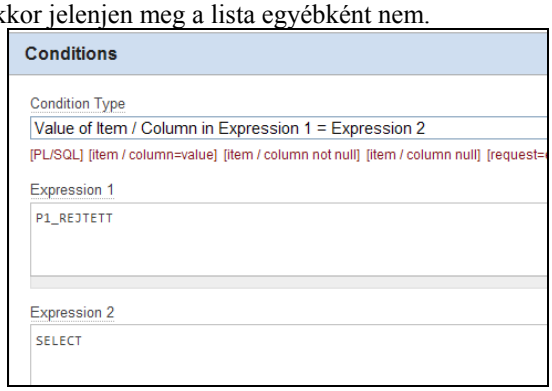
6.3. ábra A program 59_SQL_Végrehajtó.sql példa.

Az eredmény riport Típusa egy (`SQL Query (PL/SQL Function body Returning SQL Query)` forrás:

```
begin
    RETURN :P1_PARANCS;
end;
```

A `begin` és `end` szavakat nem mindig kell kiírnunk.

A riporthoz feltételt is adunk: tehát, ha az elem tartalma kifejezés1 egyenlő a kifejezés 2-vel, csak akkor jelenjen meg a riport. Tehát, ha a rejtett mezőben SELECT szócska van akkor jelenjen meg a lista egyébként nem.



6.4. ábra *Feltétel*

A program 59_SQL_Végrehajtó.sql példa a lemezen.

Dinamikus Táblalétrehozás

Alkalmazásból **dinamikus SQL** segítségével lehet táblát létrehozni.
Ez elvégezhető az alkalmazás első oldalának megjelenése előtt, de később is egy gomb lenyomásának hatására.

Példa

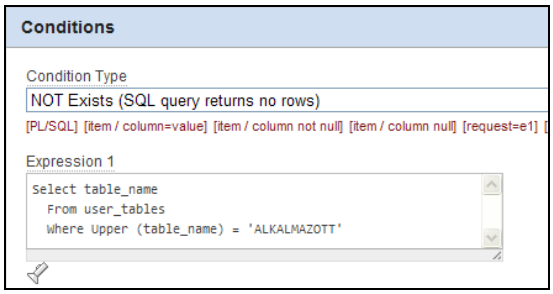
1.)Hozzuk létre egy apex alkalmazás első lapján `alkalmazott` táblát `emp` táblákból.

Hozzunk létre egy `Blank` lapot `html` mezővel, amely a gombokat tartalmazza.

A lapon legyen egy `Tábla Létrehozás` gomb, melynek lenyomására a lapon megjeleník a tábla listája. A tábla létrehozása után tünjön el a `Tábla Létrehozás` gomb és helyette jelenjen meg a `TÖRÖLD KI`! gomb.

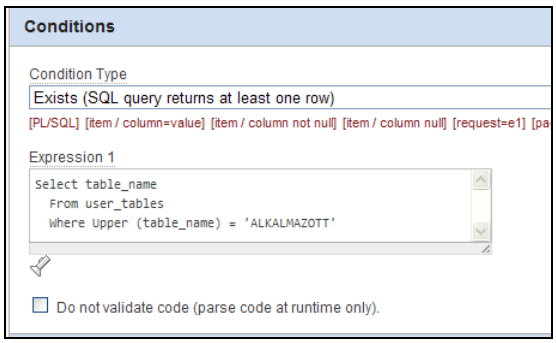
A gombok feltételei:

A `Tábla Létrehozás` gomb csak akkor jelenik meg, ha nincs tábla, a feltétel az ábrán látható



6.5.ábra *A táblalétrehozás gomb megjelenésének feltétele*

A `TÖRÖLD KI` gomb feltétel a tábla megléte:



6.6. ábra *A Töröld ki gomb feltétele*

6.7. ábra. *A táblamegjelenítő lap futási képe.*

Vezérlés

TÖRÖLD KI !

Alkalmazott tábla

Azonosító	Név	Munkakör	FőnökAzonosító	BelépésiDátum	Fizetés	Jutalék	RészlegAzonosító
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20

6.8. ábra *A létrehozott és megjelenített tábla*

Jelen esetben feladatunk a tábla létrehozása és törlése, ezért a folyamat a következő lehet:

Process Point

* Sequence 10

Process Point On Submit - After Computations and Validations

Run Process Once Per Page Visit (default)

Source

* Process [Download Source]

```
begin
 EXECUTE IMMEDIATE 'DROP TABLE ALKALMAZOTT';
exception  when others then
 EXECUTE IMMEDIATE 'Create Table Alkalmazott as Select * from emp';
 EXECUTE IMMEDIATE 'Alter Table Alkalmazott Add Primary Key (empno)';
 EXECUTE IMMEDIATE 'Alter Table Alkalmazott Add Foreign Key (mgr) references Alkalmazott (empno)';
end;
```

6.9. ábra. *Gomb lenyomására létrejövő alkalmazott tábla belépési pontja, és tartalma*

Messages

Process Success Message

```
Az Alkalmazott tabla letrehozva
```

Process Error Message

```
Sikertelen !!!!!!
```

Error message display location On Error Page

6.10. ábra *Üzenetek*

A folyamat végrehajtásának üzenetei:

FONTOS
Ügyeljünk arra, hogy az apex 4-es verzióban az üzenetekben **nem lehet ékezetes betű.**
Figyeljünk a feladatok megfogalmazására. Ha a létrehozott táblákat az alkalmazás során megváltoztatjuk, és ennek természetesen meg kell maradnia, akkor teljes vizsgálat szükséges a táblák meglétéről. Figyeljünk arra is , hogy minden alkalommal ki kell-e törölni a létrehozott táblákat, ha a lap újra betöltödik, vagy csak egyszer hozzuk létre a munkamenet folyamán (session).

Tehát a `Run process` választási lehetőséget a Végrehajtási Pont (`Process Point`) szekcióban 6.6. ábra. Két választás van:

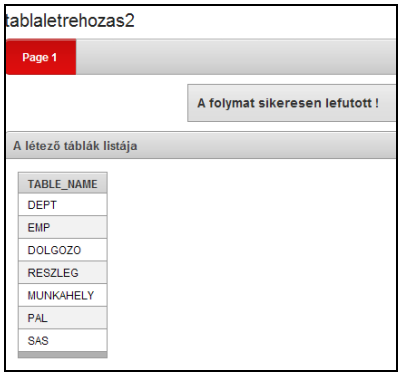
`Once Per Session or When Reset`

`Once Per Page Visit`

Session választás esetében a táblát egyszer hozza létre egy munkamenet (`session`) lefutása esetén. Ha közben kitöröljük a táblákat ellenőrzés céljából, akkor nyilván hibaüzenetet ad, hiszen a munkamenet miatt az adatok egy ideig az átmeneti tárolóban maradnak.. A hibát tehát le kell kezelni.

Amennyiben a lap szintű folyamatot (`processt`) választjuk, akkor minden lap futtatása esetén újra létrehozza a táblákat. Mivel ez nem egészen jó, hiszen a felhasználó esetleg módosíthatja a táblákat, ezért csak akkor hozzuk létre a táblákat, ha az még nem létezik. Ezt kell kivételkezeléssel megoldani. Ennek egy megoldását láthatjuk a fenti 6.6. ábrán.

Bemutatjuk a legáltalánosabb megoldást. Itt létrehozunk egy `SAS` nevű táblát az `emp` táblából és egy `PAL` nevű táblát a `dept` táblából.



6.11. ábra Futási eredmény

- A belépési pont a fejléc megjelenése előtt legyen `On Load - Before Header`
- Folyamatban:
  - ellenőrizzuk egyik vagy másik tábla meglétét, és a táblát nem dobjuk el.(ábra)
- Helyezzünk el a lapon egy a meglévő táblákat listázó riportot, hogy a táblák meglétét ellenőrizni tudjuk.

Process Point

* Sequence 10

Process Point On Load - Before Header

Run Process Once Per Page Visit (default)

Source

* Process [Download Source]

```
declare
  sastablavan   number(1);
  paltablavan   number(1);

begin
select count(*)
  into sastablavan
  from user_tables
  where table_name = 'SAS';
select count(*)
  into paltablavan
  from user_tables
  where table_name = 'PAL';

if paltablavan != 1 then
  execute immediate 'Create table Pal as select * from dept';
  execute immediate 'Alter table Pal Add Primary Key (deptno)';
end if;

if sastablavan != 1 then
  execute immediate
    'Create table Sas as select * from emp';
  execute immediate
    'Alter table Sas Add Primary Key (empno)';
  execute immediate
    'Alter table Sas Add Foreign Key (deptno) References Pal (deptno)';
  execute immediate
    'Alter table Sas Add Foreign Key (mgr) References Sas (empno)';
end if;
end;
```

6.12. ábra *Általános táblalétrehozás folyamata*

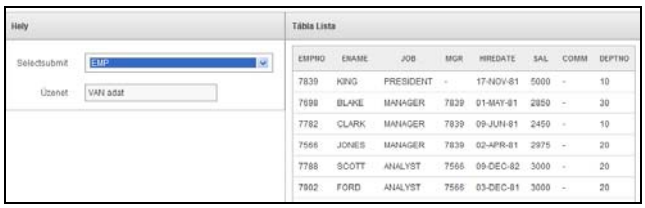
Feladat

Hozzon létre egy alkalmazást, melyben egy kiválasztó listából kijelölhető a felhasználó valamely táblája, és a kijelölés hatására közvetlenül megjelenik annak adattartalma is.

FONTOS, hogy csak megfelelően kezelt hibaüzenetek jelenhessenek meg!

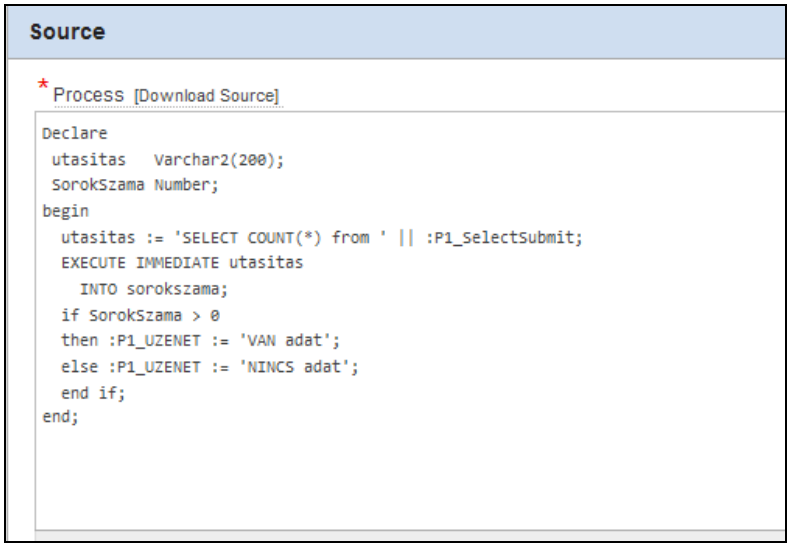
Megoldás

E feladat **50_KozvetlenKivalasztas1.sql** megoldásában különösen figyelmet érdemel annak megszervezése, ahogy az adattal nem rendelkező táblák megjelenésének automatikus hibaüzenetét kezeljük. A megoldás lényege egy olyan process létrehozása, melynek forráskódja az alábbi: A feladat futási képe:



6.13. ábra *A feladat futási képe*

A tábla ürességét ellenőrző folyamat a következő:



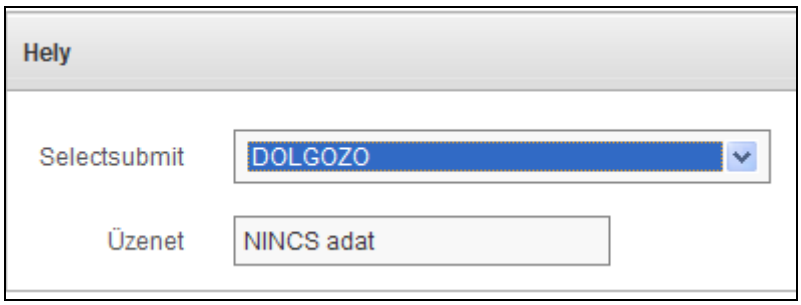
6.14. ábra *A tábla adatainak ellenőrzése*

Dinamikus SQL segítségével oldottuk meg a feladatot. Dinamikus select utasítás, amint azt már bemutattuk a következő:.

```
utasitas := 'SELECT COUNT(*) from ' || :P1_SelectSubmit;
  EXECUTE IMMEDIATE utasitas
    INTO sorokszama;
```

Az execute immediate utasításban szövegkonstansként található maga az utasítás, és kívül kapja meg a PL/SQL változuó az értéket, amely már a továbbiakban felhasználható.

Kiürítettük most a dolgozo táblát,és most futtassuk le.Kattintsunk az üres dolgozo táblára.



6.15. ábra Ellenőrzés eredménye

A hibaüzenet helyett szöveges kiírásban jelentethjük meg az adatok hiányát.

E feladat **50_KozvetlenKivalasztas2.sql** megoldásának különlegessége, hogy itt nem használunk folyamatot (`process`), hanem a vizsgálatot a lista megjelenítési feltételébe (`Conditional Display`) építjük be az alábbi PL/SQL kóddal (ahol a feltétel típusa: `PL/SQL Function Body Returning a Boolean`):

```
Declare
 utasitas   Varchar2(200);
 SorokSzama Number;
begin
  utasitas := 'SELECT COUNT(*) from ' || :P1_SelectSubmit;
  EXECUTE IMMEDIATE utasitas
    INTO sorokszama;
  if SorokSzama > 0
    then Return TRUE;
    else Return FALSE;
  end if;
exception when OTHERS
 then
   Return FALSE;
end;
```

Figyeljünk fel az `exception when OTHERS…` kivételkezelésre!

Conditions

Condition Type

PL/SQL Function Body Returning a Boolean

[PL/SQL] [item / column=value] [item / column not null] [item / column null] [request=e1] [pag

Expression 1

```
Declare
 utasitas   Varchar2(200);
 SorokSzama Number;
begin
  utasitas := 'SELECT COUNT(*) from ' || :P1_SelectSubmit;
  EXECUTE IMMEDIATE utasitas
    INTO sorokszama;
  if SorokSzama > 0
    then Return TRUE;
    else Return FALSE;
  end if;
exception when OTHERS
 then
   Return FALSE;
end;
```

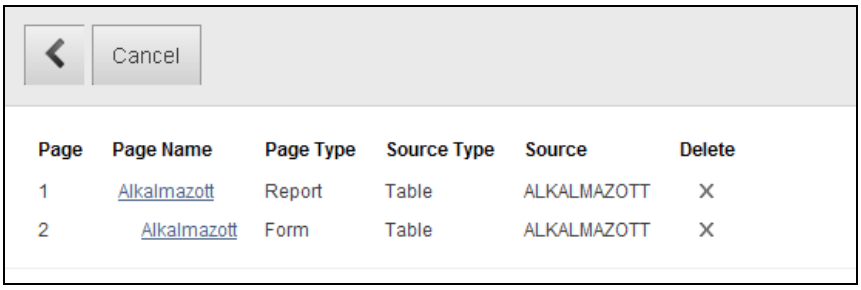
6.16. ábra A tábla tartalom ellenőrzése feltételben.

7. fejezet. Adatbevitel-adatmódosítás (Report Form)

Új alkalmazás készítése

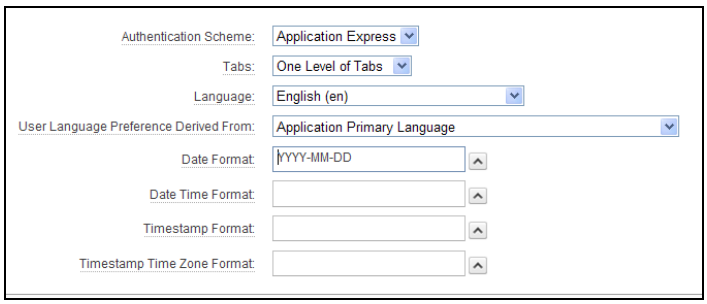
Mivel az alábbiakban bemutatásra kerülő alkalmazásfejlesztő eszközök használatához már létező adattáblákra van szükség, ezért először hozzuk létre az `Alkalmazott` és a `Munkahely` táblákat, melyek az `emp` és a `dept` táblákkal adat és megszorítás szinten ekvivalensek (annak érdekében, hogy az alkalmazás használatakor az eredeti táblatartalmak ne változzanak).

1. Készítsünk egy új alkalmazást. Indítsuk el az `Application Buildert`, és kattintsunk a `Create` gombra.
   Válasszuk a `Create Application` rádiógombot, és kattintsunk a `Next` gombra.
2. A Név (`Name`) tulajdonság legyen `RiportForm`.
   a Az alkalmazás száma maradjon az alapértelmezett.
   b A `Create Application`-nál válasszuk a `From scratch` rádiógombot.
   c A séma az alapértelmezett maradjon.
   d Majd kattintsunk a `Next` gombra.
3. Válasszuk a `Report and Form` rádiógombot
   e Adjuk meg az `Alkalmazott` tábla nevét (csak olyan táblát választhatunk ki, amelyiknek van elsődleges kulcsa).
   f Kattintsunk az `Add Page` gombra. Ekkor megjelent két lap, a `Report` és a `Form` típusú lapok, 7.1. ábra), `Next`.



7.1. ábra. *A Report and Form laptípus által létrehozott két lap*

4. Nem akarunk más alkalmazással kapcsolatba kerülni, tehát válasszuk `No` értéket a `Copy Shared Components...` beállításnál, `Next`.
5. Az Attributumok ablakban a következő tulajdonságokat állíthatjuk ba.
   -Authentikációs minta (`Authentication Schema`) maradjon az Application Express

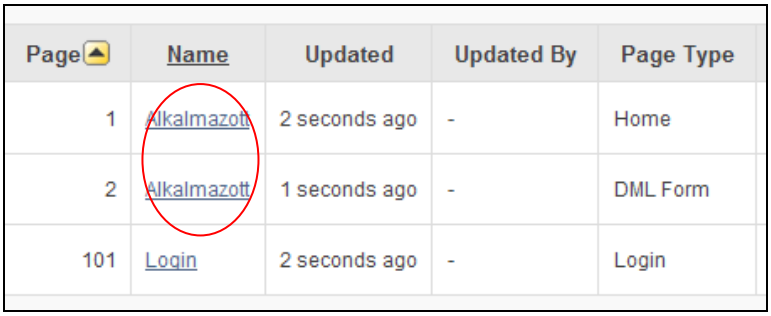


7.2. ábra Az attributumok beállítása

- Fülek beállítása (`Tabs`) legyen egyszintű (`One Level Tabs`)
- Nyelv beállítása (`Language`) maradjon az angol. (`Name` nem lehet sehol ékezetes betű)
- a preferált nyelv (`User Language` ...) legyen az alkalmazás nyelve
- dátum formátum (`Date Format`) legyen:
  a magyarhoz legjobban hasonlító `YYYY-MM-DD`

`Next`

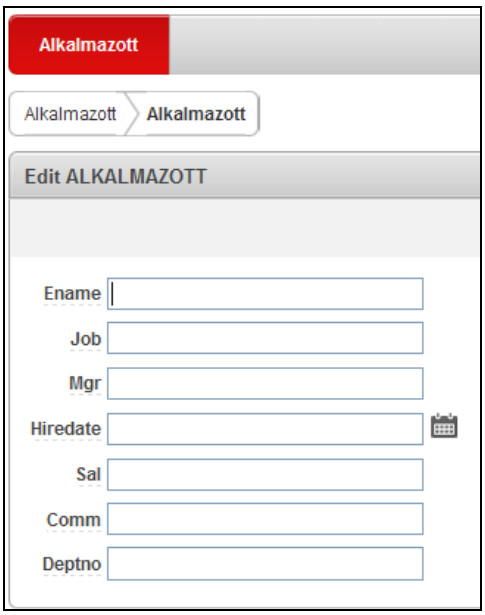
6. Válasszunk Felületi Mintát (most a `scarlet`-t), `Next`.
7. Ellenőrizzük a jellemzőket, `Create`. Ekkor megjelenik a két lap (7.2. ábra).



Page	Name	Updated	Updated By	Page Type
1	Alkalmazott	2 seconds ago	-	Home
2	Alkalmazott	1 seconds ago	-	DML Form
101	Login	2 seconds ago	-	Login

7.3. ábra. *Az elkészült két lap*

Futtassuk le (belépés magyar/Tigris-1 módon). Ha az első, a Riport lapon a `Create` gombra kattintunk, akkor megjelenik a második, az Ürlap (`Form`) lap (7.3. ábra).



7.4. ábra *A megjelenő második lap (Form)*

Mit veszünk észre? Azt hogy nincs rajta az `empno` mező. Annak érdekében, hogy ez megjelenjen, az alábbi pontban leírtak megtétele szükséges.

Először tehát a Fejlesztői Eszköztáron kattintsunk az `Edit Page 2` menüpontra illetve a 2. lapra annak érdekében, hogy módosítani tudjuk az elemek megjelenését.

Az elemek (Items) jellemzőinek megváltoztatása

1 Kattintsunk Az `Items` szekcióban kattintsunk a `P2_ENAME` menüpontra.

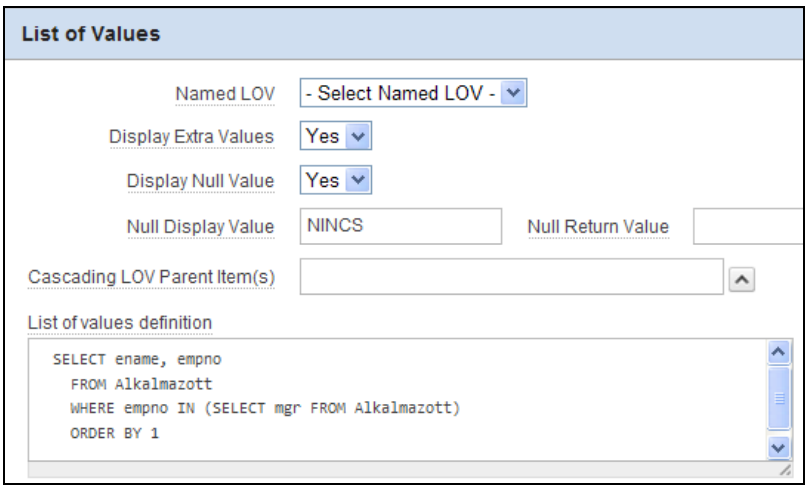
2 Legyen az `ename` kötelezően (`required`) megadandó

3 Főnökként csak már létező főnök legyen megadható, de engedjük meg azt is, hogy valakinek ne legyen főnöke. Ehhez egy kiválasztásra van szükségünk.

   a Kattintsunk az `Items` szekcióban a `P2_MGR` hiperhivatkozásra. A megjelenő szerkesztő lapon a `Name` tartományban válasszuk a `Display As` mezőnél a `Select List` értéket.

4 Menjünk lefelé, és a `List of Values` tartományban legyen a `Display Null` értéke `Yes`, valamint a `Null display value` értéke `NINCS` (7.5. ábra). Megjegyezzük, hogy azáltal, hogy a `Null return value` mezőt nem töltöttük ki, a visszatérő érték a `NINCS` kiválasztásakor `NULL` érték lesz.

   b



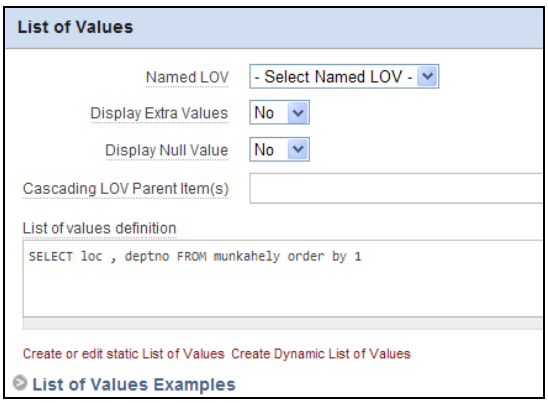
7.5. ábra. *A List of Values beállítása*

A lekérdezés, amelynek hatására `ename` jelenik meg a legördülő mezőben, de `empno`-t ad vissza az alábbi (ezután `Apply Changes`):

c
```
SELECT ename, empno
  FROM Alkalmazott
  WHERE empno IN (SELECT mgr FROM Alkalmazott)
  ORDER BY 1
```

5 Amennyiben a `Hiredate` mező mellett nem jelenik meg a naptár (a `Data Picker` ikonja), akkor azt is változtassuk meg.
Ennek érdekében kattintsunk a `Page Definition` lapon az `Items` szekcióban a `P2_HIREDATE` menüpontra. A `Name` tartományban a `Display As` legyen `DataPicker (YYYY-MM-DD)` formátum maszk érték, `Apply Changes`.

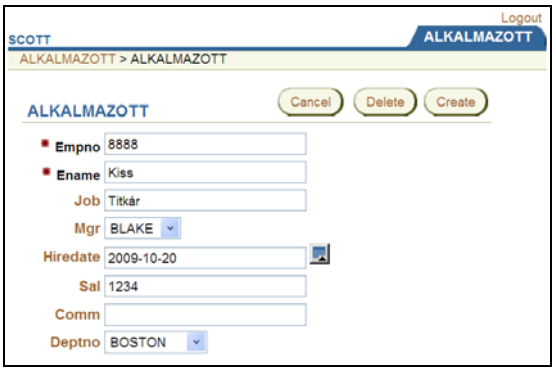
6 Végül kattintsunk az `Items` szekcióban a `P2_DEPTNO` hiperhivatkozásra.
- a A `Name` tartományban a `Display As` mezőben legyen a megjelenő `Text Field` helyett a `Select List` érték kiválasztva.
- b A `Template` legyen itt is `required`, hiszen ez idegen kulcs és kötelező megadni.
- c Menjünk le a `List of Values` tartományba, és töltsük ki megfelelően. A beállító szelekciós utasításban a telephelyet (`loc`) választjuk ki, a részlegazonosítót (`deptno`) adjuk vissza, és természetesen itt nem engedjük meg a `NULL` értéket (7.6. ábra), `Apply Changes`.



7.6. ábra. *A List of Values beállítása*

Adatbevitel

Futtassuk le az alkalmazás második lapját a jobb oldali `Run` gombbal.



7.7. ábra. *Az alkalmazás második (Report) lapja*

Vegyük fel a `Kiss` és a `Tudor` nevű dolgozókat. Ha sikeres az adatbevitel, akkor az alkalmazás visszatér az első (Riport) lapra, megjelenik az

üzenet, és a táblalistában (a Riport oldalon, 7.8. ábra) meg is találjuk a bevitt új dolgozókat. A fejlécben a felfelé mutató nyíl a rendezés szempontját és irányát jelzi. Most az `Ename` szerint növekvően rendeztünk, de bármelyik másik oszlop szerint rendezhető így a táblázat.

	Ename ▲	Job	Mgr	Hiredate	Sal	Comm	Deptno
	ADAMS	CLERK	7788	1983-01-12	1100		20
	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
	BLAKE	MANAGER	7839	1981-05-01	2850		30
	CLARK	MANAGER	7839	1981-06-09	2450		10
	FORD	ANALYST	7566	1981-12-03	3000		20
	JAMES	CLERK	7698	1981-12-03	950		30
	JONES	MANAGER	7839	1981-04-02	2975		20
	KING	PRESIDENT		1981-11-17	5000		10
	Kiss	elado		2013-10-01	1300		40
	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
	MILLER	CLERK	7782	1982-01-23	1300		10
	SCOTT	ANALYST	7566	1982-12-09	3000		20
	SMITH	CLERK	7902	1980-12-17	800		20
	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
	Tudor	elado	7698	2013-09-18	1600		40

7.8. ábra A két bevitt új rekord.

Adatrekordok szerkesztése

Egy rekord szerkesztése olymódon tehető meg, hogy a kiválasztott sor első oszlopában lévő ikonra kattintunk. Akkor az alkalmazás visszatér a Form lapra, ahol ekkor a `Create` gomb helyett az `Save` gomb jelenik meg, és megjelenő adatokat módosítani lehet. Természetesen valamely rekord törlése a `Delete` gomb segítségével tehető meg.

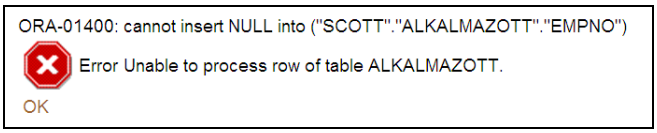
Módosítsuk a Kiss nevű dolgozó foglalkozását, és kapjon jutalékot (400-at)(7.9. ábra).

	JONES	MANAGER	7839	1981-04-02	2975		20
	KING	PRESIDENT		1981-11-17	5000		10
	Kiss	elado		2013-10-01	1300	400	40
	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
	MILLER	CLERK	7782	1982-01-23	1300		10
	SCOTT	ANALYST	7566	1982-12-09	3000		20

7.9. ábra.. *Módosítás eredménye*

Fontos!

Ha olyan `empno` értéket írunk be, amely már létezik, akkor hibaüzenetet kapunk mivel elsődleges kulcs megszorítás van az `empno` oszlopon (7.10. ábra). Hibaüzenetet kapunk akkor is ha az `empno` nem jelenik meg, mert nincs olyan folyamat, tárolt eljárás, szekvencia, vagy trigger amely automatikusan növelné annak értékét.



7.10. ábra. *Hibaüzenet*

Ezt a hibaüzenetet úgy tudjuk elkerülni például, hogy az `empno` értékét minden új belépéskor automatikusan beállítjuk, mégpedig eggyel nagyobbra, mint az aktuális dolgozók legnagyobb azonosítóját. Ezt Folymat (`Process`) megírásával állíthatjuk itt be.

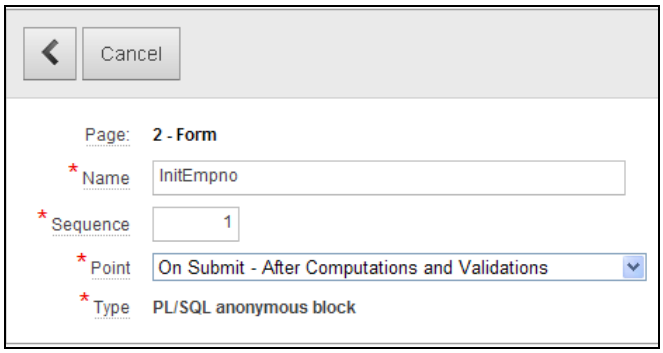
Hozzunk létre egy új folyamatot, amely előbb fut le, mint a varázsló által elkészített `Process Row of ALKALMAZOTT` nevű folyamat.(Ezt át kell számozni tehát).

Tehát a Folyamat (`Process`) szekcióban `Create (+)` jelre kattintunk,

`PL/SQL` rádiógomb választás,

`Next`

A megjelenő ablakot (7.11. ábra) töltsük ki.



7.11. ábra Az ablak kitöltése

`Name : InitEmpno`

`Sequence :` 1

`Point :` `On Submit -After Computations and Validations,`
(tehát a Kiszámító Folyamat és Ellenőrzés után fusson le)

`Next,`

Hozzuk létre a folymatot.

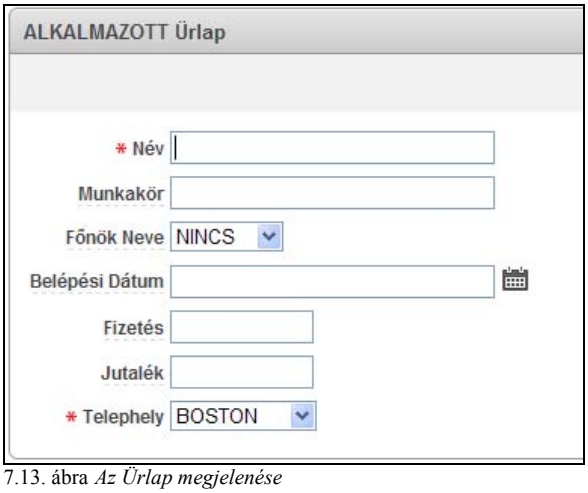
Source

* Process [Download Source]

```
DECLARE
  UjEmpno  NUMBER;
BEGIN
  SELECT max(empno) + 1
    INTO UjEmpno
    FROM Alkalmazott;
  :P2_Empno := UjEmpno;
END;
```

7.12.ábra. *a folyamatkódja*

Ekkor az `empno` beállítása nem módosítható, automatikusan eggyel nő.tehát ezt az értéket az ürlapon felesleges megjeleníteni, a listában tudjuk ellenőrizni.



7.13. ábra *Az Ürlap megjelenése*

Ne felejtsük el a már meglévő `Process` sorszámát átállítani `2`-re.

After Submit

1	InitEmpno	PL/SQL anonymous block	Conditional
2	Process Row of ALKALMAZOTT	Automatic Row Processing (DML)	Unconditional

7.14. ábra A folyamatok sorrendje

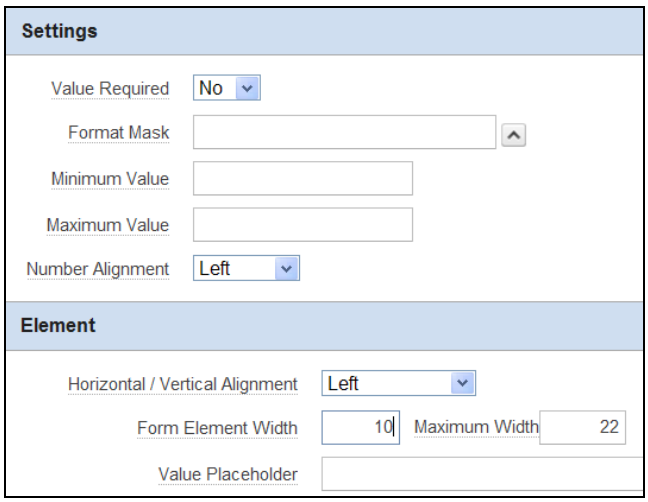
Az adatbeviteli alkalmazásunk, most már megfelelően működik.

Feliratok magyarítása

Az alkalmazásunk hiányosságának tekinthető, hogy minden felirat angol nyelvű. Változtassuk meg magyarra. Először a Riport (első) lapon megjelenő tábla oszlopnevei legyenek magyar feliratúak. Kattintsunk a Riport lap Lapszerkesztő ablakában az `Alkalmazott` hiperlinkre a tartományok (`Regions`) részben, (vagy a `report` hiperlinkre). Az így megjelenő lap `Report Attributes` fülére kattintva, a `Custom` menüpont kijelölése után az összes fejlécet átírhatjuk magyarra. Változtassuk meg az attribútumok sorrendjét is a fel és lefelé mutató nyilakkal, majd `Apply Changes`. Így megkaptuk a magyar feliratú táblát az első (Riport) lapon.

Az Űrlapon (`Form`) lapon az elemek (`Items`) neveit (`Label`) kell megváltoztatnunk.

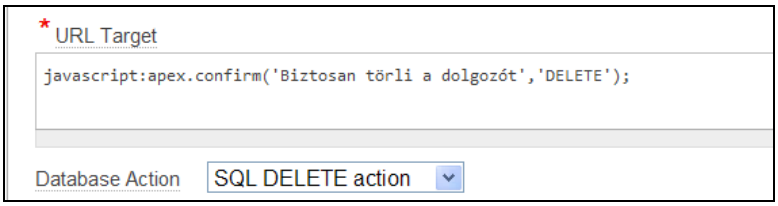
Numerikus mezők hosszát vegyük kisebbre, és a számot igazítsuk balra. (7.15. ábra)



7.15. ábra *Elem beállítása*

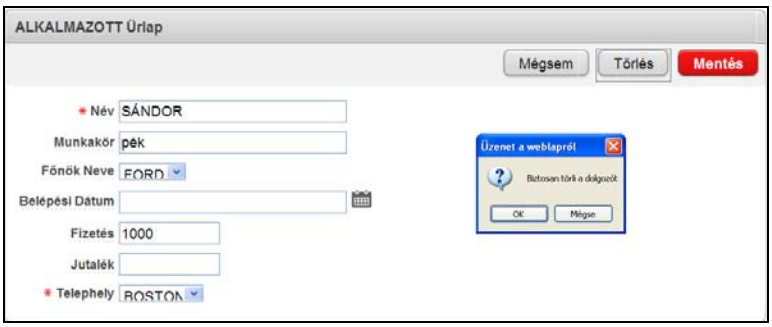
Írjuk át a gombok és az üzenetek feliratát is magyarra. A gomboknál csak a címkét (`Text Label/Alt`) szükséges átírni, a név (`Button Name`), amire esetleg hivatkozás történik természetesen változatlan marad.

Annak érdekében, hogy a törlést megerősítő kérdés is magyar nyelvű legyen, a Törlés gomb szerkesztésénél a megírt java-szkript szövegét kell átírni magyarra.



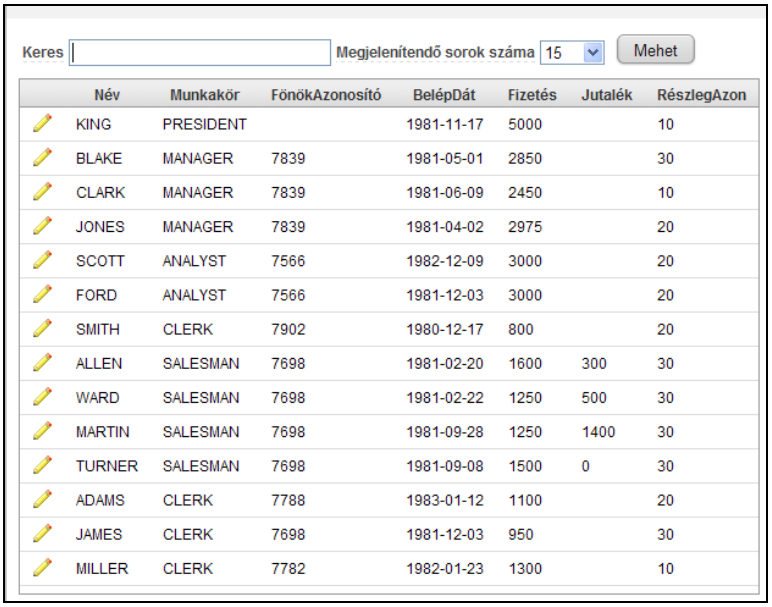
7.16. ábra *A Törlés kérdésének magyarítása*

Ekkor törlés kérdése magyarul jelenik meg:



7.17. ábra A Törlés rákérdezésének magyarítása

Az első lapon is elvégezve a megfelelő magyarításokat, a következő futási képet kajuk:



Keres | Megjelenítendő sorok száma 15 | Mehet

Név	Munkakör	FőnökAzonosító	BelépDát	Fizetés	Jutalék	RészlegAzon
KING	PRESIDENT		1981-11-17	5000		10
BLAKE	MANAGER	7839	1981-05-01	2850		30
CLARK	MANAGER	7839	1981-06-09	2450		10
JONES	MANAGER	7839	1981-04-02	2975		20
SCOTT	ANALYST	7566	1982-12-09	3000		20
FORD	ANALYST	7566	1981-12-03	3000		20
SMITH	CLERK	7902	1980-12-17	800		20
ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
WARD	SALESMAN	7698	1981-02-22	1250	500	30
MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
TURNER	SALESMAN	7698	1981-09-08	1500	0	30
ADAMS	CLERK	7788	1983-01-12	1100		20
JAMES	CLERK	7698	1981-12-03	950		30
MILLER	CLERK	7782	1982-01-23	1300		10

7.18. ábra *A Riport lap (első) futási képe*

Ez már valóban megfelelő.

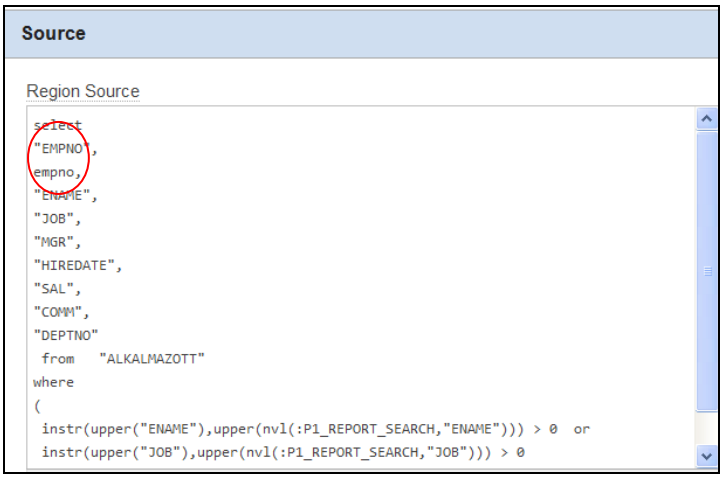
Kapcsolat a két lap között

Célszerű megjeleníteni ellenőrzésnek, az első `empno` mezőt is. Ezért kattintsunk az `empno` melletti ceruzára, Így szerkeszthetjük meg az `empno` mezőt. Természetesen az Ürlappal (`Form`) való kapcsolatnak meg kell maradnia.
Érdemes megvizsgálni a két lap közötti kapcsolatot. E kapcsolatért az első (Riport) lap ALKALMAZOTT riportja a felelős. Az alábbiakban ezt mutatjuk meg.

Tegyük fel, hogy valamilyen okból az ALKALMAZOTT riport Forrását (Source szekció) generikus oszlopnevekre állítottuk (`Use Generic Column Names...`)(Például szeretnénk

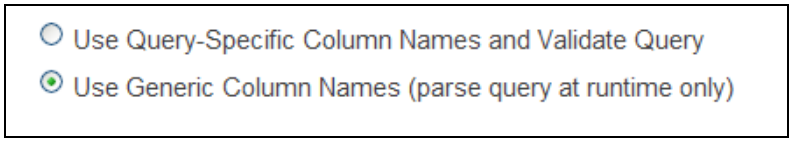
megjeleníteni az `empno` oszlopot is, tehát a forrásba beírjuk mégegyszer az `empno` oszlopot.

Ennek oka lehet például az, hogy a fölötte lévő SQL-utasításban egy még nemlétező táblára hivatkozunk, vagy egy olyan függvényre amelynek értéke valamilyen, az alkalmazás használatakor megadott értéktől függ. A generikus beállítás lehetővé teszi, hogy a fejlesztőkörnyezet által nem ellenőrizhető SQL-utasítást használhassunk.



7.19. ábra *A forrásban megkettőzzük az* `empno` *oszlopot.*

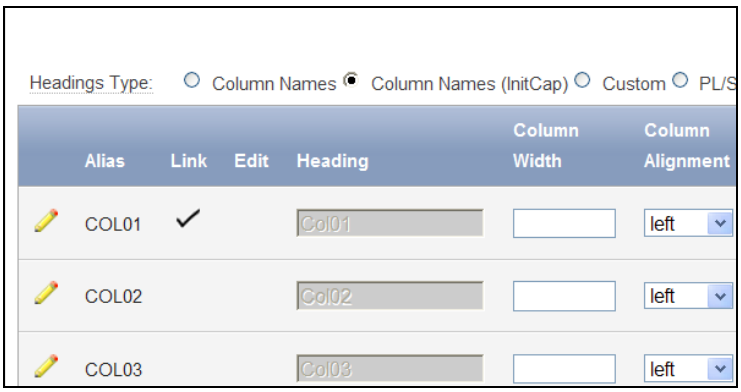
Ekkor azonban (éppen az ellenőrizhetőség hiánya miatt) az adott lapon minden automatikusan létrehozott hivatkozás elveszik! Elveszik tehát az oszlopokra vonatkozó minden beállítás, így például az oszlop-átnevezések, és a másik lapra mutató kapcsolatok is. Ennek visszaállítása meglehetősen nagy munka, de természetesen elvégezhető. Annak érdekében, hogy ez mit is jelent megmutatjuk az első lap futási képét, ami úgy jött létre, hogy felvettük az `empno` oszlopot mégegyszer, ekkor a riportot generikusra kellett állítani, és ezt elmentettük.



7.20.ábra *A forrás ellenörzése most vagy futás közben*

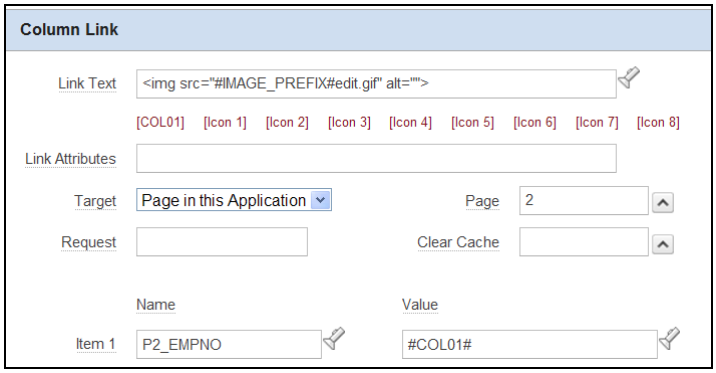
A változás láthatóan elég durva, és ez marad akkor is, ha a riportot visszaállítjuk a másik állapotba (`Use Query-Specific Column Names...`), és azt is elmentjük. A szükséges beállítások elvégzésének módja a következő.

Tekintsük e riport tulajdonság-szerkesztő lapját (az `Alkalmazott riport` hiperlinkjére kattintva), és kattintsunk a `Report Attributes` fülre. A `Heading types` legyen `Column Names` ekkor az eredeti oszlopnév fog a listában megjelenni. Itt a `Column Attributes` szekcióban belépünk az `EMPNO` illetve `COL01` oszlop szerkesztésébe legelső oszlop (az előtte lévő ceruza ikonra való kattintással).



7.21.ábra *A fejléc típusa maga az oszlop neve legyen*

E lapon a `Column Link` szekcióban kattintsunk a `LINK Text` mező mellett lévő elemlámpa ikonra (`flashlight`), és válasszuk ki azt a mezőt (`#EMPNO#`), vagy `ikont` melyet kapcsoló mezőnek szeretnénk, továbbá lapként (`Page`) a 2 értéket állítsuk be, és az `Item1` sorában pedig magát a kapcsolatot, vagyis a `Name` mezőben a `P2_EMPNO`, a `Value` mezőben pedig ismét az `#COL01#` értéket (szintén az elemlámpa ikon segítségével).



7.22. ábra *A kapcsolat frelépítése*

A megjelenő listában már visszaállítottuk a kapcsolatot. Láthatjuk hogy az `Edit` ikont választottuk kapcsoló mezőnek. Ebben az esetben a futási kép az alábbi.

Empno	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
Edit	7839	KING	PRESIDENT		1981-11-17	5000		10
Edit	7698	BLAKE	MANAGER	7839	1981-05-01	2850		30
Edit	7782	CLARK	MANAGER	7839	1981-06-09	2450		10
Edit	7566	JONES	MANAGER	7839	1981-04-02	2975		20
Edit	7788	SCOTT	ANALYST	7566	1982-12-09	3000		20
Edit	7902	FORD	ANALYST	7566	1981-12-03	3000		20
Edit	7369	SMITH	CLERK	7902	1980-12-17	800		20
Edit	7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
Edit	7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
Edit	7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
Edit	7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
Edit	7876	ADAMS	CLERK	7788	1983-01-12	1100		20
Edit	7900	JAMES	CLERK	7698	1981-12-03	950		30
Edit	7934	MILLER	CLERK	7782	1982-01-23	1300		10

7.23. ábra *A lista futási képe*

Már csak vissza kell írni a magyar fejléceket, ahogy az már ismert.

A megjelenő táblalistán már az `edit` ikon értékek ismét a megfelelő (második lapra mutató) hiperlinkkel jelennek meg. A riport képe ekkor ismét a fenti szépen beállított.

Megjegyezzük, hogy az `ikon` értékek helyett maga az `empno` is lehet kapcsoló szövegben.

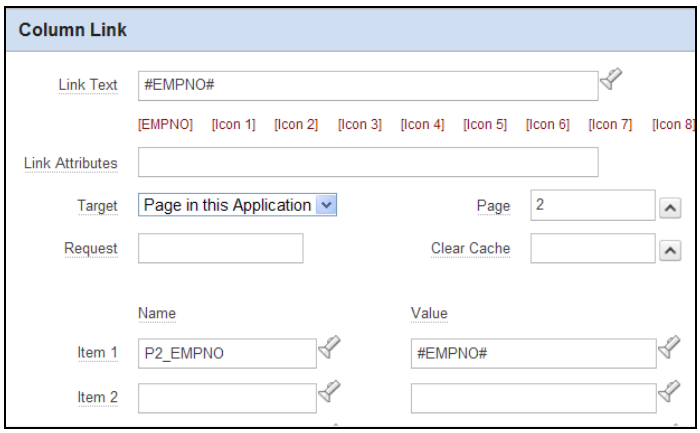
Ekkor a forráshoz nem nyúlunk. A Riport szerkesztésében a `<br/>` szócskát töröljük, helyébe az `azonosító` szót írjuk(ekkor az empno oszlopot nem kell duplázni)

Headings Type: ○ Column Names ○ Column Names (InitCap) ◉ Custom ○ PL/SQL ○ None

	Alias	Link	Edit	Heading	Column Width	Column Alignment	Heading Alignment	Show	Sum
	EMPNO	✓		 		left	center	☑	☐
	ENAME			Név		left	center	☑	☐

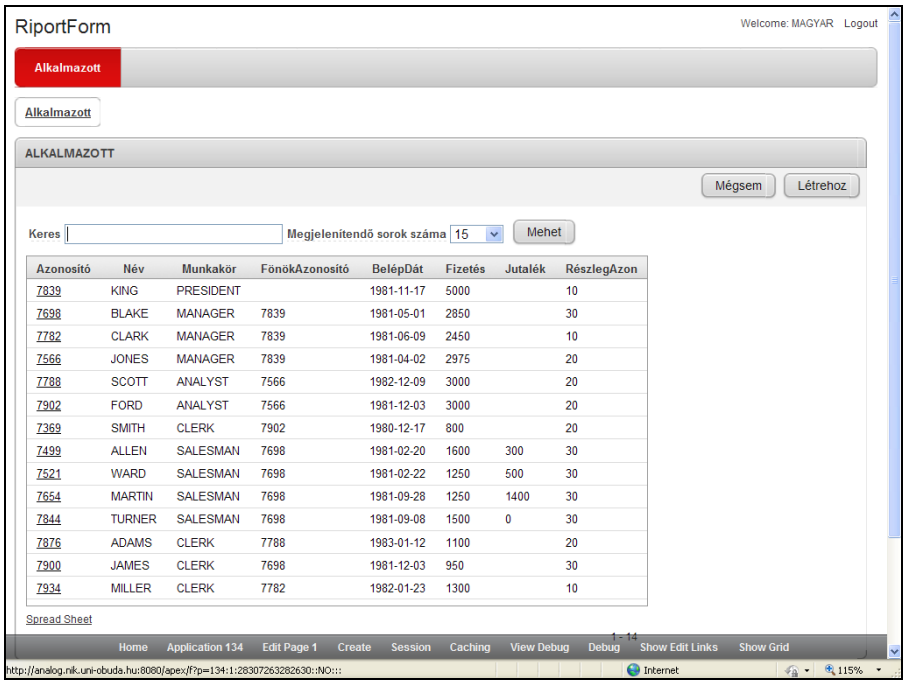
7.24. ábra *A riport szerkesztése*

A `Column Link` szekcióban a kapcsoló mező tartalom (`Link Text`) kijelölését az `#EMPNO#` választásával végezzük.



7.25.ábra Az *oszlop kapcsolójának beállítása*

Ekkor a teljesen elkészült alkalmazásunk riport lapjának futási képe.



Azonosító	Név	Munkakör	FönökAzonosító	BelépDát	Fizetés	Jutalék	RészlegAzon
7839	KING	PRESIDENT		1981-11-17	5000		10
7698	BLAKE	MANAGER	7839	1981-05-01	2850		30
7782	CLARK	MANAGER	7839	1981-06-09	2450		10
7566	JONES	MANAGER	7839	1981-04-02	2975		20
7788	SCOTT	ANALYST	7566	1982-12-09	3000		20
7902	FORD	ANALYST	7566	1981-12-03	3000		20
7369	SMITH	CLERK	7902	1980-12-17	800		20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1983-01-12	1100		20
7900	JAMES	CLERK	7698	1981-12-03	950		30
7934	MILLER	CLERK	7782	1982-01-23	1300		10

7.26. ábra Futási kép, amikor az empno a kapcsoló mező

Kattintsunk a Létrehoz gombra, megjelenik az ürlap.

ALKALMAZOTT Űrlap

Mégsem | Hozd Létre!

* Név Kelemen
Munkakör SALESMA
Főnök Neve BLAKE
Belépési Dátum 2013-10-06
Fizetés 1100
Jutalék
* Telephely CHICAGO

7.27. ábra *Kelemen dolgozó felvétele*

Vegyük fel `Kelemen` dolgozót, és kattintsunk a `Hozd Létre` gombra, akkor betette az adattáblánkba `7935 empno`-val (max(empno) + 1) értékkel `Kelemen` dolgozót.

7934	MILLER	CLERK	7782	1982-01-23	1300	10
7935	Kelemen	SALESMAN	7698	2013-10-06	1100	30

Spread Sheet

1 - 15

7.28. ábra *Részlet a riportból*

Még egyet próbáljunk felvenni

Azonosító	Név	Munkakör	FőnökAzonosító	BelépDát	Fizetés	Jutalék	RészlegAzon
7936	Magyar	ANALYST	7566	2013-10-02	2900		30

Spread Sheet

Previous row(s) 16 - 16 of 16

7.29. ábra *A Felvett 16. dolgozó*

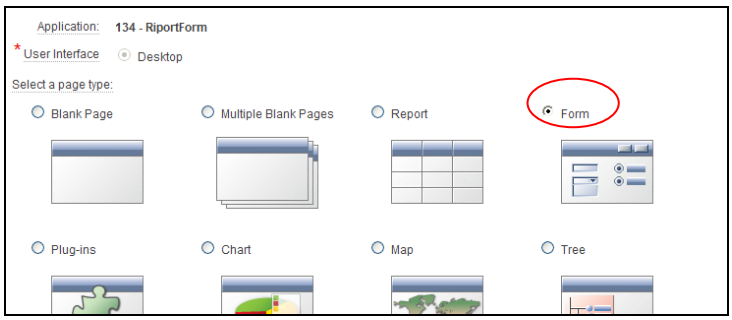
Ezt a Magyar nevű dolgozót `7936`-os azonosítóval felvette az új alkalmazottak közé. Ez az alklamzás tehát megfelelően működik.

Adatbevitel már létező alkalmazáshoz

Természetesen nem szükségszerű, hogy az adatbeviteli (módosító, törlő) funkció az első (és második) lapon legyen. Az alábbiakban azt mutatjuk be, hogyan lehet egy már létező alkalmazáshoz hozzátenni e funkciókat.

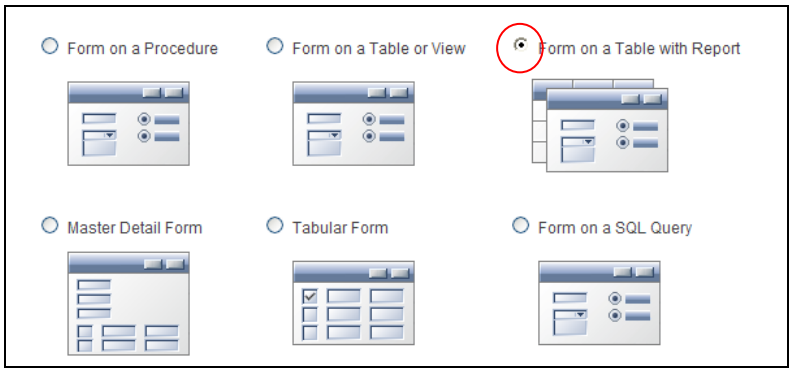
A most létrejött alkalmazásunkhoz tegyünk hozzá egy munkahely beviteli `Report` és `Form` adatlapot

Kattintsunk a lap létrehozásra (`Create Page`), majd itt válasszuk a `Form` lapot. `Next`.



7.30. ábra *Alkalmazás kiegészítése adatbevitellel*

A megjelenő ablakban válasszuk a `Form on Table with Report` rádiógombot. `Next`.



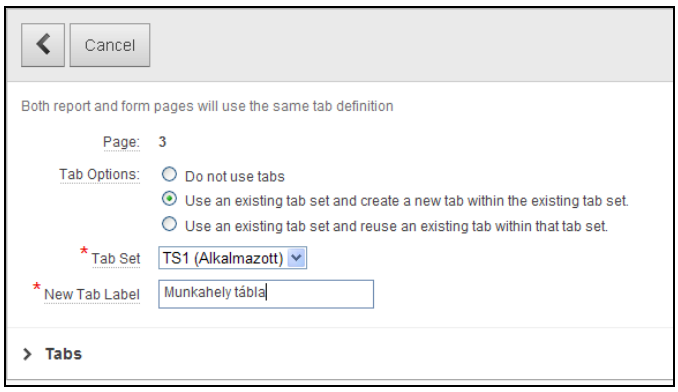
7.31. ábra *Form ona Table with Report választás*

Válasszuk az Ürlap (`Form`) laptípust, `Next`.
Innen teljesen ugyanaz, mint a `Report and Form` típus,
Válasszuk a

Megvalósitás (`Implementation`) :	Klasszikus riportot,
Lapszám (`Page Number`)	3,
Lapnév (`Page Name`)	Munkahely
Tartomány fejléce (`Region Title`)	Munkahely riportja
Tartomány sablon (`Region Template`)	Template 26 Standard
`Breadcrumb`	ne legyen

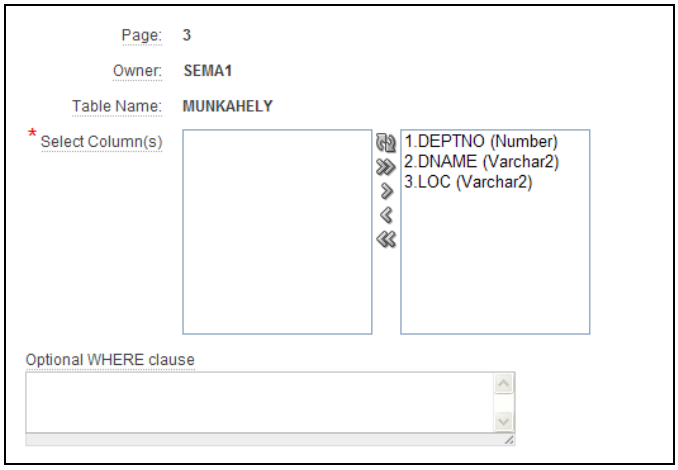
Ezek után `Next`, majd válasszuk ki a tulajdonost (`Sema1`), és a táblát (`Munkahely`) `Next`.

Kapcsolódjunk a már létező fül-készlethez (megfelelő beállítással), `Next`.



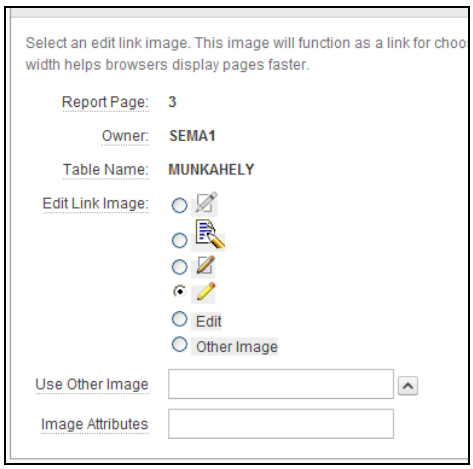
7.32.ábra A Lapnak legyen füle

Jelöljük ki (a szokásos Windows technikával) az összes oszlopot, `Next`.



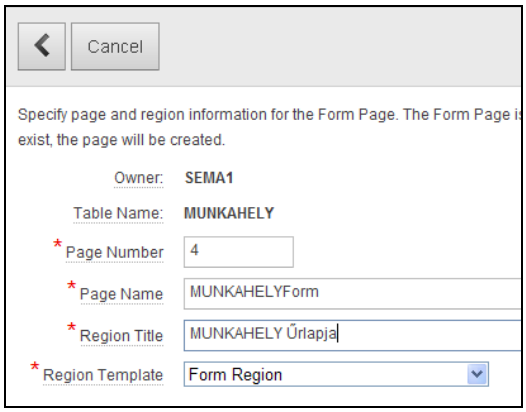
733.ábra *Mindhárom oszlopot jelöljük ki.*

Ezután válasszuk ki a "kedvenc" jelölőelemünket, `Next`.



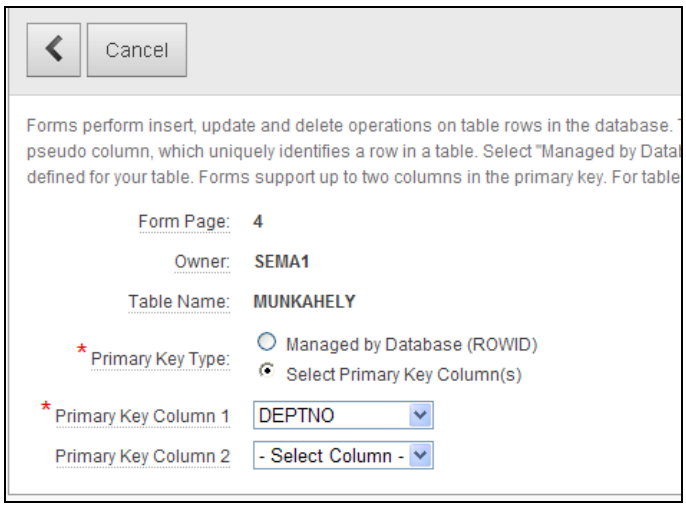
734.ábra *Kapcsoló mező választás*

A következő ablakban adjuk meg az adatbeviteli-módosító régió nevét, `Next`.



7.35.ábra Az Űrlap létrehozása

A megjelenő ablakban, meg kell határoznunk mi azonosítja a tábla sorait.



7.36. ábra *Primary Key választás*

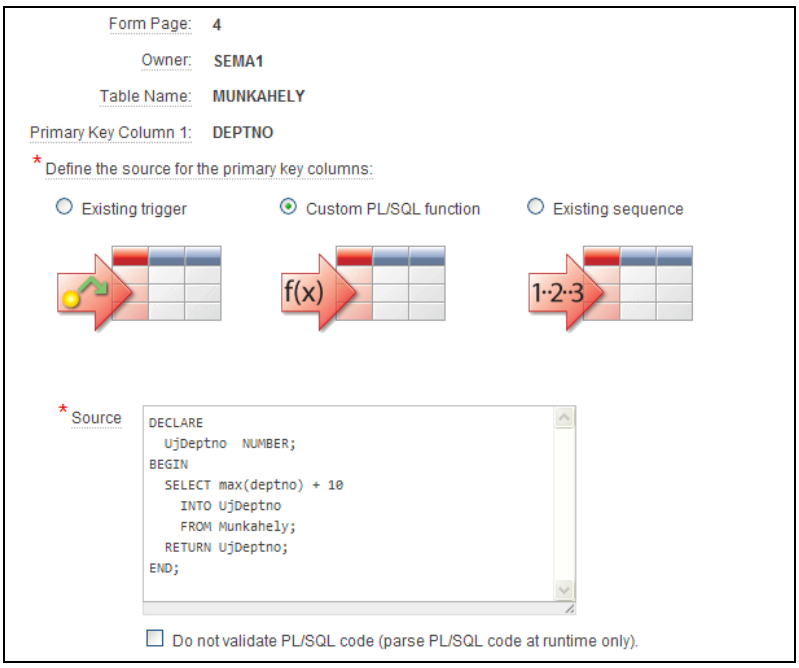
Mit is válasszunk?

Az ürlapon adatbevitel, módosítás és törlés végezhető el a tábla sorain. Ehhez a sorokat azonosítani kell. Az azonosítás történhet a táblához megadott Elsődleges megszorítással, illetve összetett elsődleges kulccsal. (`Primary Key`). Az azonosítás másik lehetősége a `ROWID` pszeudo oszlop, ez is egyedileg azonosítja a tábla sorait. Egy vagy két oszlopos Elsődleges megszorítás (`Primary Key`) esetén, a `Select Primary Key` rádiógomb választása a helyes. Amennyiben az elsődleges megszorítás kettőnél több oszlopot tartalmaz, akkor a `Managed by Database (ROWID)` választás a megfelelő.

Választásunk tehát a `Primary Key` rádiógomb (7.36.ábra), `Next`

A megjelenő ablakban válasszuk ki az elsődleges kulcsot beállító eszközt, amely lehet létező trigger, egy szokásos PL/SQL program, vagy egy létező szekvencia. Megírhatjuk a triggert is, és a szekvenciát is az alkalmazásban mielőtt ezt az alkalmazást futtatni fogjuk.

Most válasszuk a `Custom PL/SQL function` rádiógombot, és írjuk meg a PL/SQL programot, amely a legnagyobb részleg-azonosítóhoz (`deptno`) 10-et ad hozzá, és így kapjuk meg az új elsődleges kulcs értéket.

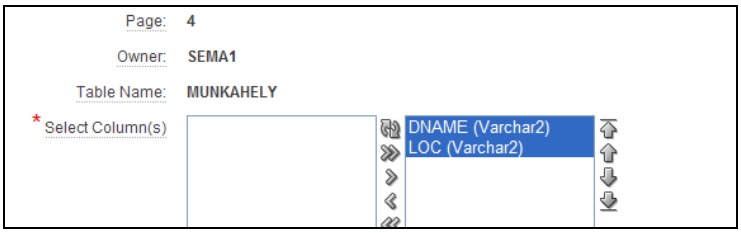


7.37. ábra *Az elsődleges kulcs beállításának lehetőségei*

Ebben az esetben egy új sor bevitelénél a `deptno` `10`-el lesz nagyobb, mint a táblában levő legnagyobb érték:

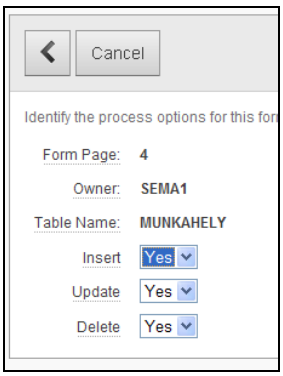
Menjünk tovább; `Next`.

Válasszuk ki a Ürlap oldalon a megjelenő többi oszlopot,



7.37.ábra Megjelenítendő oszlopok

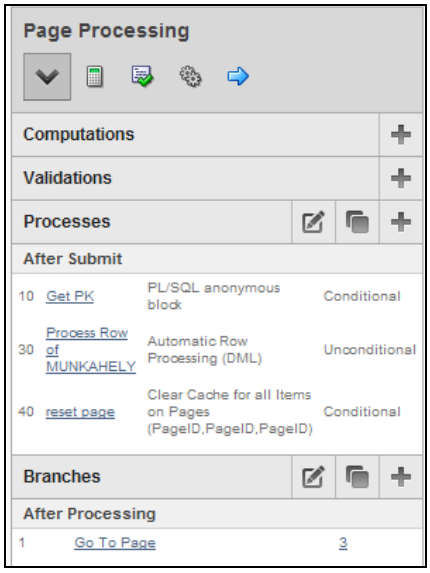
`Next`,



7.38-ábra *Milyen DDL utasításokat hajt majd végre az alkalmazásunk*

Majd a funkciókat összesítő lap jóváhagyása után `Next` és `Create`.

Figyeljük meg, hogy az alkalmazás ilyen űrlapjának létrehozásakor a rendszer egy új folyamatot (`process`) hozott létre a 4. lap `Lap-Definiciós` oldalán a `Folyamat Definiciós` szekcióban. E "`Get PK`" nevű processz egy get_pk nevű függvényt tartalmaz, melyben található az általunk megírt PL/SQL program.



7.39. ábra *A varázslóval előállított folyamat*

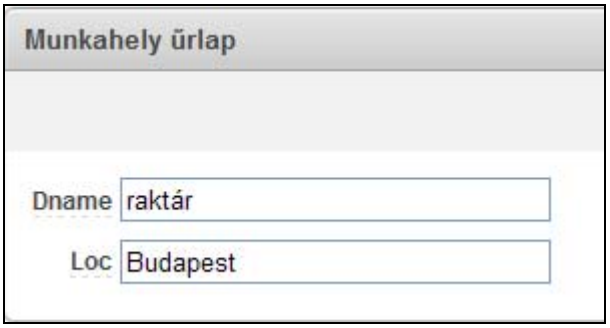
A `GetPK` folyamat tartalma.

```
-- deklaráció--ba egy függvény található
declare
    function get_pk return varchar2
    is
    begin
        DECLARE
           UjDeptno  NUMBER;
```

```
        BEGIN
          SELECT max(deptno) + 10
            INTO UjDeptno
            FROM Munkahely;
        RETURN UjDeptno;
        END;
    end;
-- főprogram
begin
    :P4_DEPTNO := get_pk;
end;
```

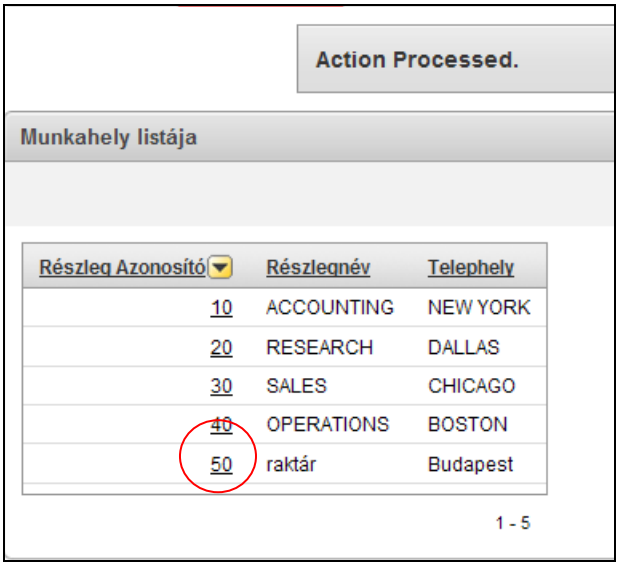
A függvény visszatérési értéke maga a neve, és ezt egy rejtett elem veszi fel:

Futtassuk le az alkalmazást. Vigyünk be új adatot. Ebben az esetben itt nem jelenik meg az elsődleges kulcs, de bevitelkor már van értéke, mert a `Get PK` processz beírja az új értéket.



7.40 ábra *Egy új részleg felvétele*

Ellenőrizzük az adatbevitelt



7.41.ábra *Sikeres Adatbevitel*

Ezért itt megjelenik a Primary Key oszlop (deptno) is. Végezzük el a feliratok magyar nyelvű átalakítását is. A végső futási eredmény:

TRIGGER LÉTREHOZÁSA

Az alábbiakban bemutatjuk (ismétlés jelleggel) a szekvencia és trigger létrehozásának SQL programjait.

A szekvencia:

```
CREATE SEQUENCE azonosito START WITH 7935
INCREMENT BY  1
MaxValue  20000;
```

A trigger:

```
CREATE OR REPLACE TRIGGER autoszam
BEFORE INSERT ON Munkahely FOR EACH ROW
BEGIN
  IF :new.deptno IS NULL
  THEN
    SELECT azonosito.nextval
      INTO :new.deptno
      FROM DUAL;
   END IF;
END;
```

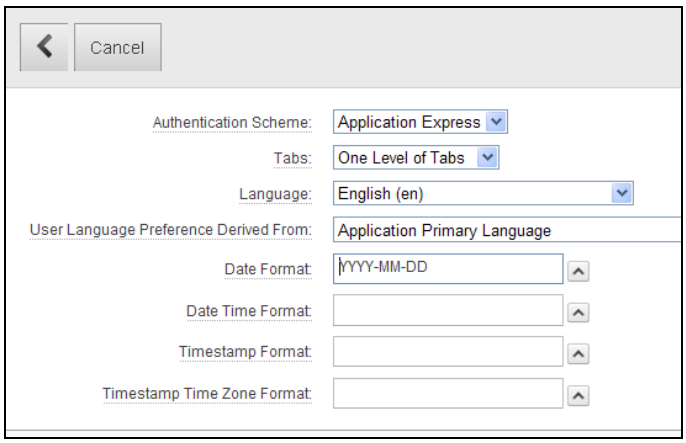
Mindkét utasítás dinamikusan:

```
EXECUTE IMMEDIATE 'CREATE SEQUENCE azonosito START WITH 7935 INCREMENT BY
1 MAXVALUE 20000';
EXECUTE IMMEDIATE 'CREATE OR REPLACE TRIGGER autoszam BEFORE INSERT ON
Munkahely FOR EACH ROW BEGIN IF :new.deptno IS NULL then SELECT
azonosito.nextval INTO :new.deptno FROM DUAL; END IF; END;';
```

8. fejezet.
Adatszintek közötti lefúrás
(Drill Down)

Készítsünk egy új alkalmazást

1 Válasszuk az `Application Builder` ikont
2 Kattintsunk `Create Application gombra`
3 Válasszuk az Adatbázis `(Database)` rádiógombot. és nyomjuk le a `Next` gombot.
   a) Az alkalmazás azonosító száma maradjon az alapértelmezett
   b) Adjunk nevet az alkalmazásnak, legyen `Kapcsol`
   c) Séma (`Schema`) maradjon változatlan
   d) Válasszuk a `Create Option` mezőnél a `Start from scratch` lehetőséget
   e) A felhasználói felület (`User Interface`) maradjon a `Desktop`
   f) Kattintsunk a `Next` gombra.
4 Vegyünk fel két lapot.
   a) *Első lap*: legyen a lap típusa: `Blank`
   b) A neve (`Page Name`) legyen (ékezetmentesen): `reszleg`
   c) Az `Add Page` gombra kattintsunk.
   d) A *második lap* típusa legyen szintén: `Blank`
   e) A neve legyen: `alkalmazott`
   f) Kattintsunk az `Add Page` gombra.
   g) Kattintsunk a `Next` gombra.
5 Ne másoljuk a megosztott komponenseket (`Copy Shared)`, maradjon `No, Next.`
6 A következő ablak maradjon változatlan, állítsuk be a dátumformátumát, `Next`

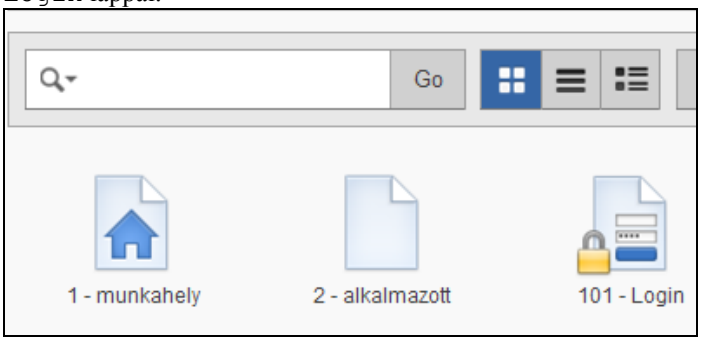


ábra

7 Felhasználói felületnek válasszuk a 21-es számú Felületi Mintát (`Standard Theme`), majd kattintsunk a `Next` gombra.

8 Ellenőrizzük a jellemzőket, majd kattintsunk a `Create` Application gombra.

Megjelenik az elkészült alkalmazás, az `1-munkahely`, a `2-alkalmazott` és a `101-Login` lappal.



ábra

Készítsük el az első munkahely lapot

A munkahely lap tartalmazza a munkahely tábla listáját, de úgy hogy az egyik oszlop tartalma hiperlink (úgynevezett kapcsoló mező) leszKattintsunk a `Create Page` gombra.

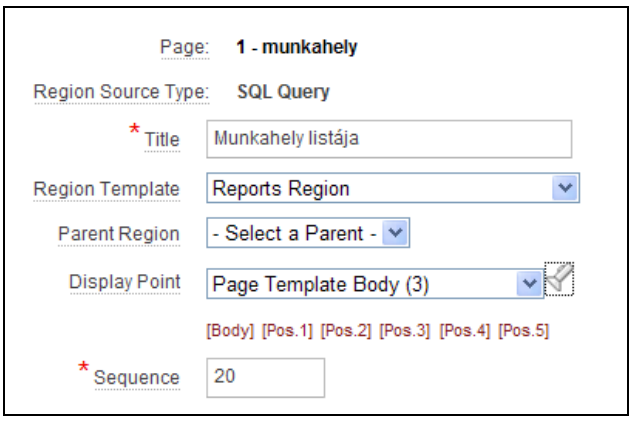
1 Készítsük el az első lapot, kattintsunk a lapra.

2 Hozzunk létre egy riport tartományt. Kattintsunk a Plusz(+) jelre.

3 Válasszuk a `Report` rádiógombot, majd kattintsunk a `Next` gombra.

4 Válasszuk a riport varázslót (`Wizard Report)` rádiógombot, `Next`.

5 Irjuk be a fejlécet , a többi maradhat változatlan.



ábra

6 Válasszuk ki a lapon megjelenő táblát

7 A következő lapon válasszuk ki a táblákat.
   a) A tábla/tulajdonos maradjon az alapértelmezett
   b) A tabla/nézet neve (`Name`) legördülő listában válasszuk először a `munkahely` táblát
   c) A `CTRL +` az egér bal gombbal kiválaszthatjuk a megjelenő oszlopokat (`deptno`, `loc, dname`), és tegyük át a megjeleníteni kívánt területre.
   d) Amennyiben egy másik táblát is hozzá szeretnénk kapcsolni, akkor újra kattintsunk a Tábla kiválasztás gombra. Ha a tábla nem jelenik meg, akkor kattinthatunk előtte a `Show Only Related Tables` –nél a `No`-ra.
   Most válaszuk az `alkalmazott` táblát és helyezzük át a `deptno` oszlopot.
   e) Kattintsunk a `Next` gombra.

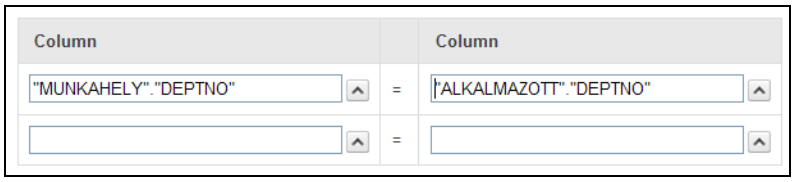
Készítsük el a második lapot

1 A lap már létezik, ezért Edit Page 2 vel tudjuk megszerkeszteni.
   a) Hozzu7nk létre egy riport tartományt, majd kattintsunk a `Next` gombra.
   b) Válasszuk az `Wizard Report` rádiógombot, majd kattintsunk a `Next` gombra.

2 A lap azonosítása
   a) A tartomány fejléce legyen `A kiválasztott részleg alkalmazottai`
   b) a többi maradjon változatlan
   c) Next

3 A táblakiválasztás.
   a) Válasszuk az `alkalmazott` táblát
   b) CTRL bal egér gombbal válasszuk ki az összes oszlopot
   c) A mutasd csak a vele kapcsolatban levő táblákat
   (`Show related Tables only`) legyen `No` így ki tudunk vele más táblát is választani
   d) Válasszuk a `munkahely` táblát is, és helyezzük át a `deptno` oszlopot, `Next`

e) Válasszuk ki a kapcsolatokat, kattintsunk a nyilra, majd a search gombra kattintsunk a megfelelő oszlopra.



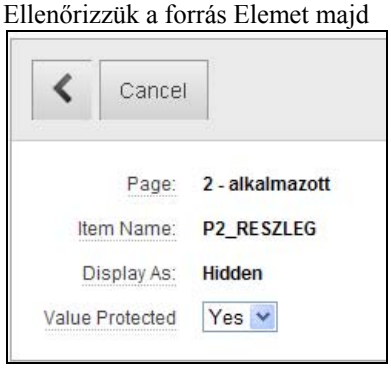
. ábra. *Az összekapcsolási feltétel*

f) Kattintsunk a `Next` gombra.
g) Majd `Create Region` gombra kattintunk.
h) Futtassuk le. (megjelenik a teljes alkalmazott tábla , két deptno oszloppal)

Csak adott részlegszámú dolgozó adataira vagyunk kíváncsiak. Ezért adjunk hozzá egy rejtett Elemet, amelybe a részlegszámot fogja kapni az első lapról.

Rejtett Elem hozzáadása

1 Menjünk az `Edit Page 2` gombra a fejlesztői eszköztáron, kattintsunk rá.
2 Az `Items` szekcióban kattintsunk a `Create` (+) gombra.
3 Válasszuk a `Hidden` rádiógombot, majd Next
   a) Legyen a neve. `P2_reszleg`
   b) A `Sequence` maradhat a felajánlott
   c) A `Region` legyen `A kiválasztott részleg alkalmazottai`
   d) Kattintsunk a `Next` gombra.
   e) Ellenőrizzük a forrás Elemet majd `Next`

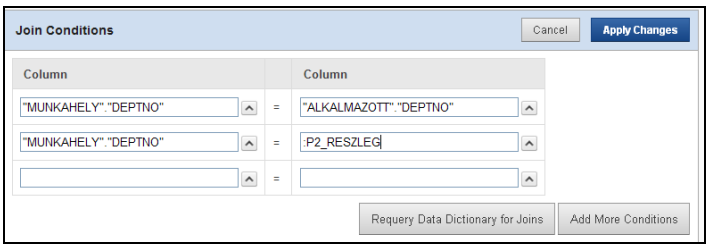


ábra

f) `Create Item`

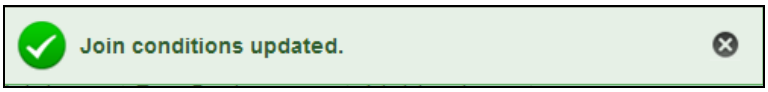
Feltétel hozzáadása

1 Kattintsunk a második lapon a `region` szekcióban a `A kiválasztott részleg alkalmazottai` hiperhivatkozásra.
2 Válasszuk a `Query Definition` fület.
Kattintsunk a `Modify Join Conditions` (8.4. ábra)
3 A megjelenő ablakban egy feltétel látszik, mégedig a szokásos `munkahely.deptno = alkalmazott.deptno`. Adjunk hozzá még egy feltételt, mégpedig a a `P2_reszleg` tartalma (`:`), legyen a munkahely táblában kijelölt részleg. A rejtett elemet be kell írni, míg a tábla megfelelő oszlopa kiválasztható.



8.4. ábra. *A feltételekbeállítása*

4 Kattintsunk az `Apply Changes` gombra. (8.4. ábra)
Megjelenik az üzenet (8.5. ábra.) Join condition updated.



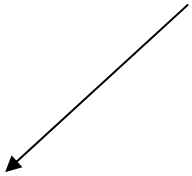
8.5. ábra. feltételek megváltoztatása

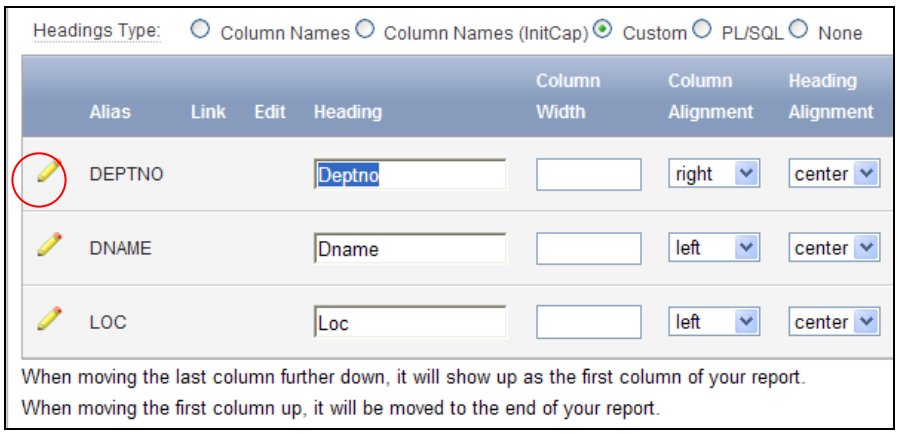
5 Menjünk vissza az első lapra..

Összekapcsolás

Végül kapcsoljuk össze a két lekérdezést. Az összekapcsolást a `munkahely` lapon kell megtennünk.
Menjünk az első lapra úgy, hogy beírjuk az `1` lapszámot, és kattintsunk a `GO` gombra (8.5. ábra)
A tartomány (`region`) szekcióban válasszuk ki a `munkahely listája` hiperhivatkozást, vagy közvetlenül a `Report` hiperhivatkozást
Válasszuk a `Report Attributes` fület.

1 Szerkesszük a `deptno` oszlopot (8.6. ábra).





8.6. ábra.*A* `deptno` *oszlop módosítása*

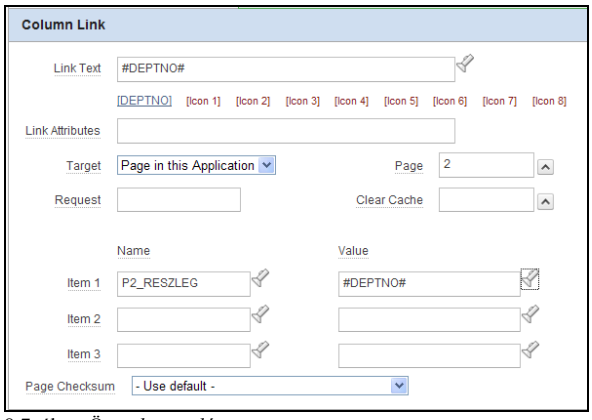
2 Menjünk lefelé a `Column Link` területre, és állítsuk be a következőket (8.7. ábra).

a) A kapcsoló szöveg (`Link Text`) legyen a `#DEPTNO#`

b) A lapszám `2`

c) Az Elemnév a rejtett Elem (`Item 1`), azaz `P2_reszleg`

d) Az Elem (`Item 1`)értéke (`Value`) értéke pedig, szintén az átadott `#DEPTNO#`, ezek az elemel az elemkeresővel (`Find Item`) történhet.



8.7. ábra. *Összekapcsolás*

e) Menjünk vissza a lap tetejére, és `Apply Changes` gombra kattintsunk.

3. Futtassuk az alkalmazást (8.8. ábra)

Munkahehy listája

Deptno	Dname	Loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

1 - 4

8.8. ábra. *A „lefúrás" alkalmazás első lapjának futása*

Láthatjuk, hogy megjelent a `deptno` oszlopban a hiperhivatkozás, ha rákattintunk, akkor a `2-részleg` lapon megjelennek a kiválasztott részlegben dolgozók adatai.
Kattintva a `20`-as részlegre, megjelennek a `20-`as részleg dolgozói.(8.9.ábra)

A kiválasztott részleg alkalmazottai

Empno	Deptno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7566	20	JONES	MANAGER	7839	1981-04-02	2975		20
7788	20	SCOTT	ANALYST	7566	1982-12-09	3000		20
7902	20	FORD	ANALYST	7566	1981-12-03	3000		20
7369	20	SMITH	CLERK	7902	1980-12-17	800		20
7876	20	ADAMS	CLERK	7788	1983-01-12	1100		20

ábra

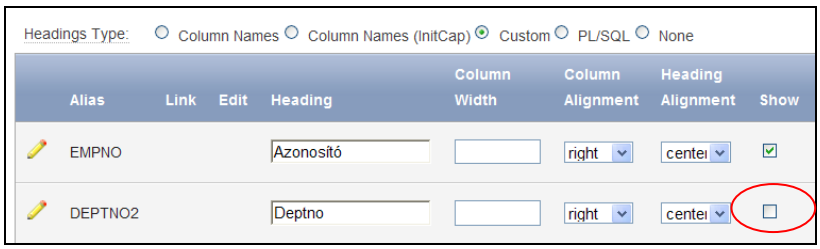
Fejléc magyarítása

Most már csak a következő van hátra:

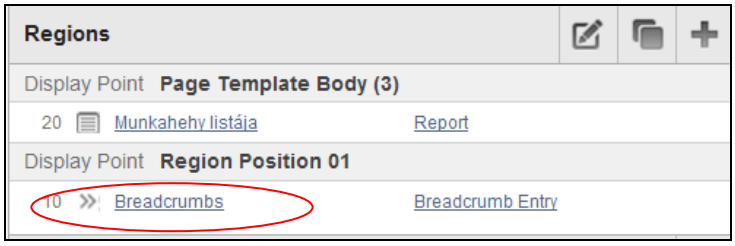
- az alkalmazott táblá listájában csak egyszer jelenjen meg a deptno oszlop, Valamint a fejlécek magyarítása min a két mlistában.

  a.) `Edit Page 2`

  b.) Tartomány (`Regions`) szekcióban `Report` hiperhivatkozásra kattintva a `Show` oszlopban a `Depno 2`-nél a kis `Check Boxra`, az oszlop nem fog megjelenni, és írjuk át magyarra a fejléceket , majd `Applay Changes`



ábra



c.)Végül töröljük a lapokról a visszalépő sávot, mert nincs rá szükségünk.

-`Edit Page 1`

-Kattintsunk a Visszalépő sáv (`Breadcrumbs`) hiperhivatkozássra , majd kattintsunk a `Delete` gombra, majd a következő lapon `Delete Region` gombra.

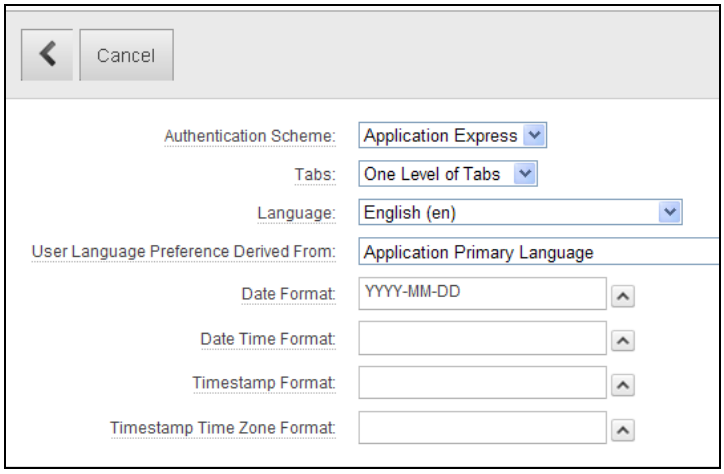
-Ugyanezt végezzük el a második lapon is.

Ennek módszernek az előnye, hogy "automatikusan" létrehozza a riportot és a második lap esetén a két megadott tábla között "automatikusan" felajánlja a kapcsolatot is (az attribútumnevek szerinti természetes összekapcsolás alapján).

E "varázslós" megvalósításnak azonban hátránya is van, ugyanis a későbbi módosítást nagyon nehézkesen, bizonyos módosításokat pedig egyáltalán nem tesz lehetővé.

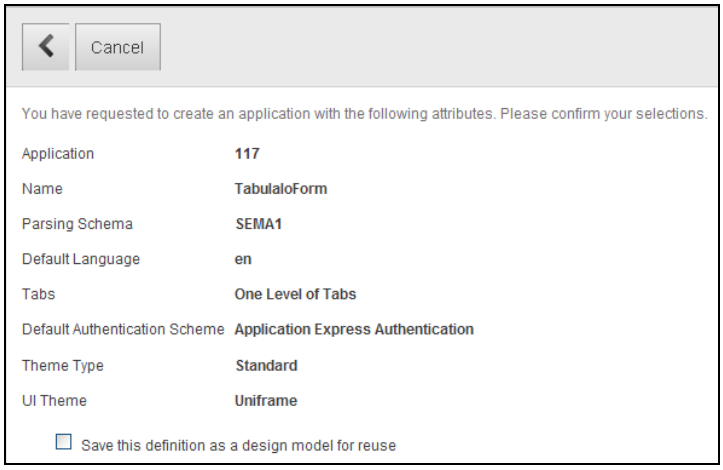
9. fejezet.
Táblázatos Forma készítése
(Tabuláló Form)

1. Készítsünk egy új alkalmazást.
   Indítsuk el az `Application Buildert`, és kattintsunk a `Create` gombra.
2. Válasszuk a `Create Application` rádiógombot, és kattintsunk a `Next` gombra.
3. Adjuk meg az alkalmazás nevét
4. `Name` legyen `Tabulalo Form`
5. Az alkalmazás száma maradjon a felajánlott érték.
6. Válasszuk a `Create Option` legördülő menüből a `From Scratch` rádiógombot
7. A séma maradjon az alapértelmezett
8. Kattintsunk a `Next` gombra
9. Válasszuk a `Tabular Form` rádiógombot
10. Válasszuk ki a táblát, ez legyen az `alkalmazott`
11. Kattintsunk az `Add Page` gomra
    a) Kattintsunk a `Next` gombra.
12. A `Copy Shared Componens` menüben válasszuk a `No` rádiógombot, és kattintsunk a `Next` gombra.
13. A tulajdonságok ablakban válasszuk a `One Level of Tabs`, válasszunk dátumformátumot, többit ne változtassuk.(9.1.ábra)



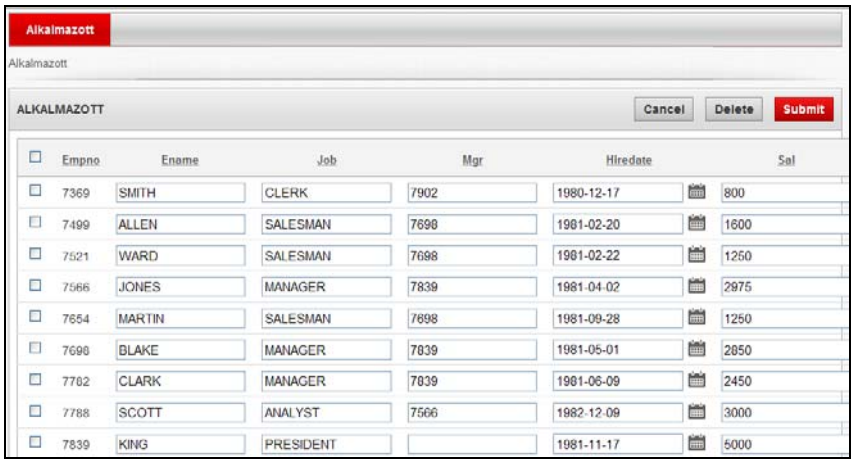
9.1.ábra Egyszintű Fül, és dátumformátum választás

14. Kattintsunk a `Next` gombra
15. A felhasználói felületnek válasszunk témát, kattintsunk a `Next` gombra.



9.2. ábra Ellenőrizzük

16. Ha átnéztük a választott form tulajdonságait, kattintsunk a `Create Application` gombra.
17. Futtassuk le.
18. Feladatunk az empno érték automatikus beállítása, a fejlécek és gombok magyarítása



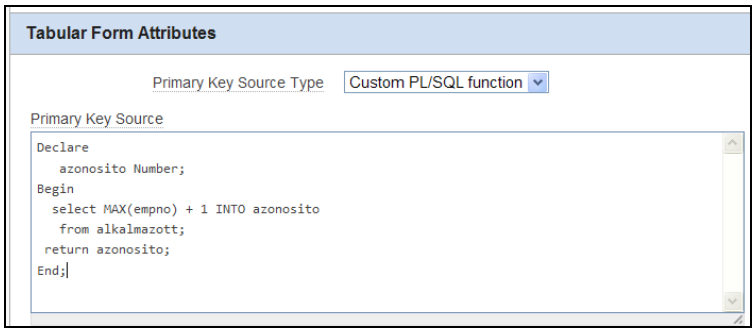
9.3. ábra. *A Táblázatos Forma alkalmazás futtatása*

Az `empno` (elsődleges kulcs) érték automatikus beállítása

```
1.Edit Page 1
```

2.Tartomány szekcióban a Report - ra kattintsunk
3.Figyeljük meg, hogy a varázsló létrehozott még egy emno oszlopot (EMPNO_DISPLAY) néven, és ez fog megjelenni, viszont ez nem szerkeszthető (nincs mellette pipa)
4.Kattintsunk az eredeti empno oszlop szerkesztő ikonjára

5.Menjünk le a Tabuláló Form tulajdonságai (Tabular Form Attributes) szekcióba
6.Válasszuk az elsődleges kulcs forrásának (Primary Key Source Type) a Custom PL/SQL function Sem létező Triggerünk, sem szekvenciánk nincsen. Ezek előállítását lásd alább.
7.Írjuk meg az azonosító (empno) automatikus növelését

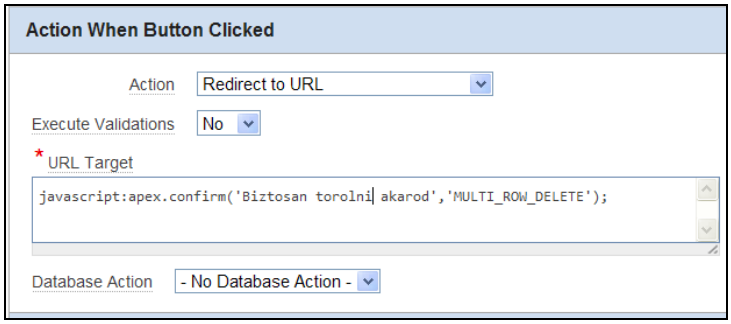


9.4. ábra *A forrás PL/SQL programja*

8.Applay Changes
9.Írjuk át magyarra a fejléceket, Applay Changes
10.Kattintsunk az Item szekcióban a gombokra és írjuk át a Felírat (Text Label/ALL) mezőt magyarra.

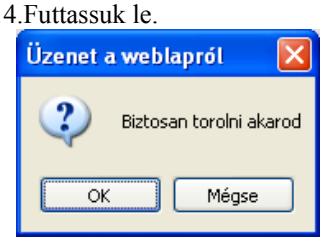
Cancel	Mégsem
Submit	Végrehajt
Add Row	Sor Hozzáadás

11.A Törlés (Delete) gomb esetén a megerősítő kérdést is írjuk át, menjünk lejjebb.
12.az Action When Button Clicked szekcióhoz és az *URL Target szövegdobozban írjuk át az üzenetet magyarra. Figyeljünk arra, hogy **ékezetes betű ne legyen** benne!!!



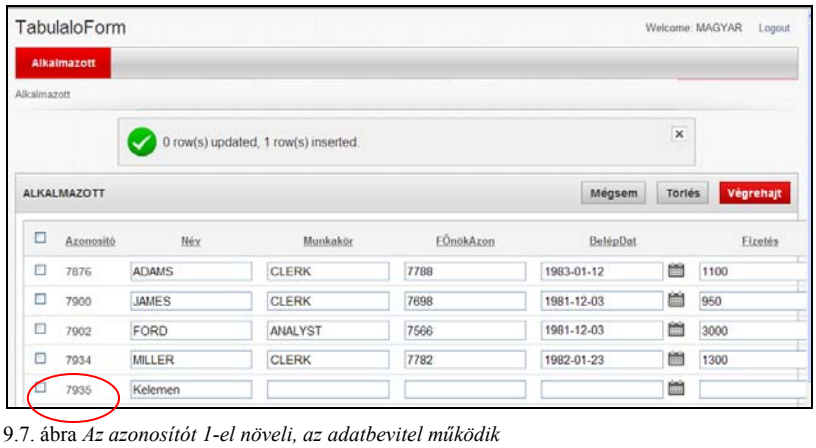
9.5. ábra *A Törlés megerősítésének átírása*

13.Applay Changes
14.Futtassuk le.



9.6. ábra *A Törlés megerősítésének widows ablaka*

Adjuk hozzá `Kelemen` alkalmazottat, láthatjuk, hogy jól működik



9.7. ábra *Az azonosítót 1-el növeli, az adatbevitel működik*

Oszlop-lista választása

Az alábbiakban megmutatjuk, miként lehet például a `deptno` oszlop tulajdonságait úgy megváltoztatni, hogy az egyes részlegek illetve telephelyek listából legyenek kiválaszthatóak, ne legyen szükség ezeknek beírására

Amikor a `Tabulár Form` varázsló létrehozta a formot, a módosítható oszlopok alapértelmezetten `Text` mező típusúak, azonban lehetséges egy oszlop típusát úgy megváltoztatni, hogy egy listából legyen kiválasztható. Ennek folyamata az alábbi:

1 Menjünk vissza a második lapra a fejlesztői eszköztár segítségével (`Edit Page 2`).

2 Kattintsuk a tartomány (`Region`) szekcióban a `report` hiperhivatkozásra (9.2. ábra.)

Headings Type: Column Names Column Names (InitCap) Custom PL/SQL None

Alias	Link	Edit	Heading	Column Width	Column Alignment	Heading Alignment	Show	Sum	Sort	Sort Sequence
[row selector]		✓	Sor választó		left	center	☑	☐	☐	-
EMPNO_DISPLAY			Azonosító		left		☑	☐	☑	-
EMPNO		✓	Empno		right	left	☐	☐	☑	1
ENAME		✓	Név		left		☑	☐	☑	-
JOB		✓	Munkakör		left		☑	☐	☑	-
MGR		✓	FőnökAzonosító		right		☑	☐	☑	-
HIREDATE		✓	BelépDat		left		☑	☐	☑	-
SAL		✓	Fizetés		right		☑	☐	☑	-
COMM		✓	Jutalék		right		☑	☐	☑	-
DEPTNO		✓	Részleg Azonosító		right		☑	☐	☑	-

When moving the last column further down, it will show up as the first column of your report.
When moving the first column up, it will be moved to the end of your report.

9.8. ábra *A Report hiperlinkre kattintva megjelenik a szerkeszthető lista*

3 Kattintsunk a szerkesztő ikonra a balszélen a `deptno` oszlopban

4 Menjünk lefelé a `Tabular Form Element` részig, és a `Display As` mezőben a `Text Field` helyett válasszuk: `Select List (query based LOV)`

5 Menjünk lejjebb a `List of Values` részig, és írjuk be a lekérdezést. például:

List of Values

Named LOV: - Select Named LOV -

Display Extra Values: No Display Null: No

Null display value: Null return value:

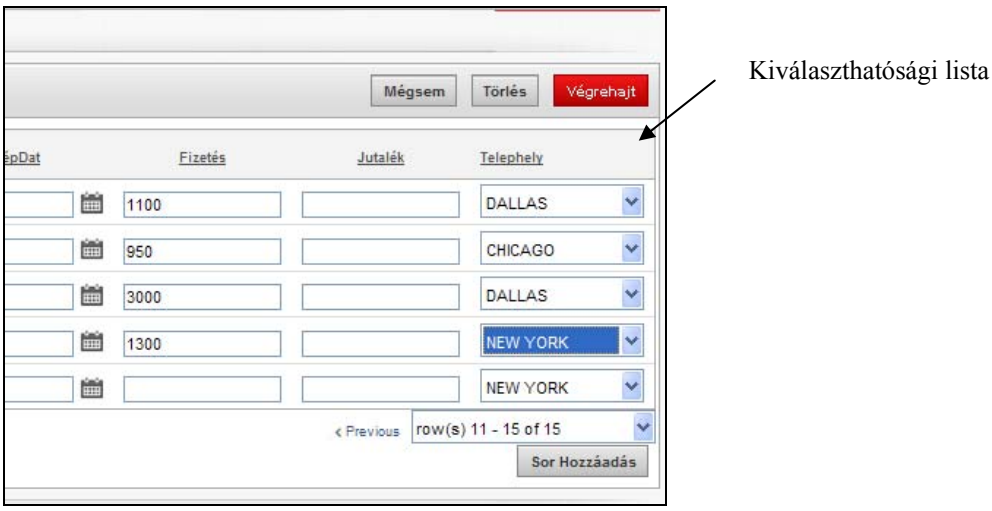
List of values definition

```
select loc, deptno from munkahely
```

9.9 ábra A legördülő kiválasztó elem forrása

6 Menjünk a lap tetejére és kattintsunk az `Apply Changes` gombra.

7 Futtassuk le. Ennek eredménye a 9.10.ábrán látható.

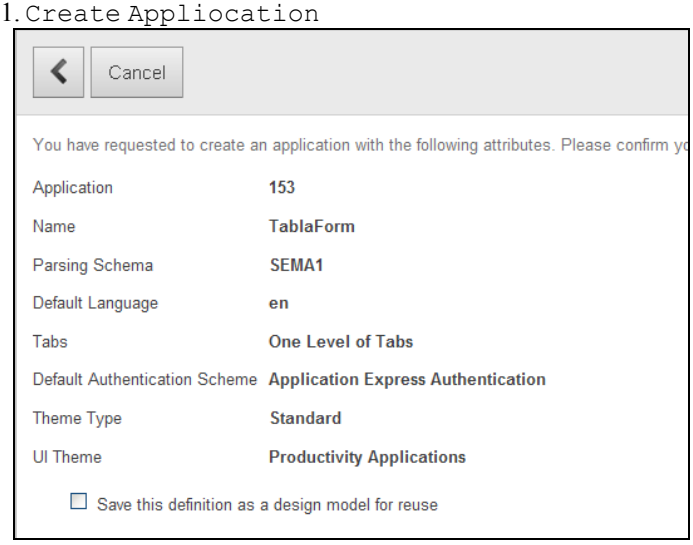


9.10. ábra. *A lista megjelenése*

Úgyanígy a munkakör és a főnök kódja helyett a főnök nevét kiválaszthatóvá tehetjük. Ezt a következő példában be is mutatjuk

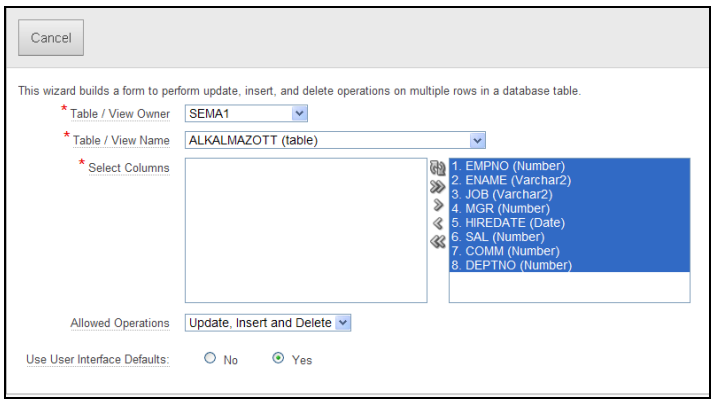
Táblázatos Forma (Tabuláló Form) hozzáadása egy alkalmazáshoz

1. Készítsünk egy új alkalmazást.
   Indítsuk el az `Application Buildert`, és kattintsunk a `Create` gombra.
2. Válasszuk a `Create Application` rádiógombot, és kattintsunk a `Next` gombra.
3. Adjuk meg az alkalmazás nevét
4. `Name` legyen `TablaForm`
5. Az alkalmazás száma maradjon a felajánlott érték.
6. Válasszuk a `Create Option` legördülő menüből a `From Scratch` rádiógombot
7. A séma maradjon az alapértelmezett
8. Kattintsunk a `Next` gombra
9. Válasszuk a `Blank` rádiógombot
10. Kattintsunk az `Add Page` gomra
    b) Kattintsunk a `Next` gombra.
11. `Create Appliocation`



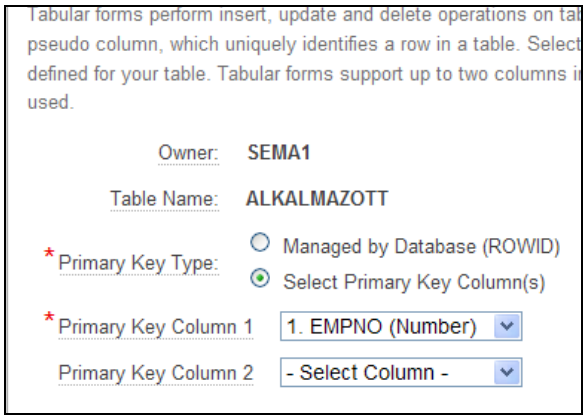
9.11. ábra *A létrehozandó alkalmazás tuéajdonságai*

12. `Create Page`
13. Form rádiógomb, `Next`
14. `Tabular Form` rádiógomb, `Next`
15. Válasszuk ki a táblát illetve megjelenítendő oszlopait.



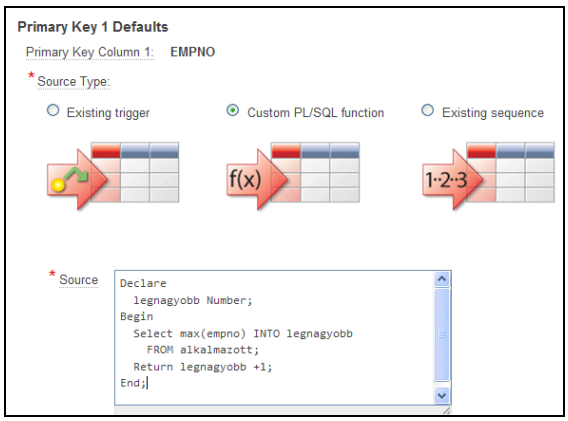
9.12. ábra *Tábla kiválasztása, oszlopok átvitele*

16. `Next`
17. Válasszuk az Elsődleges Kulcs (`Primary Key`) rádiógombot, és az empno oszlopt.a többit nem változtassuk



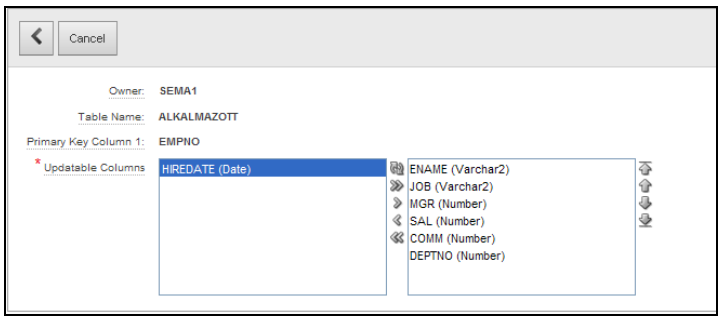
9.13. ábra *Az elsődleges kulcs megadása*

18. Válasszuk a `Custom PL/SQL Function` rádiógombot, és írjuk meg az `empno` automatikus növelésére a programot.



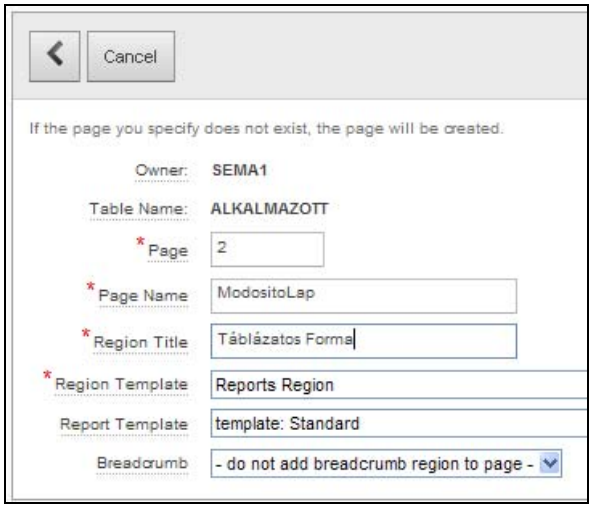
9.14. ábra *Az azonosító automatikus növelése*

19. `Next`,
20. Jelöljük ki a módosítható oszlopokat, `Next`



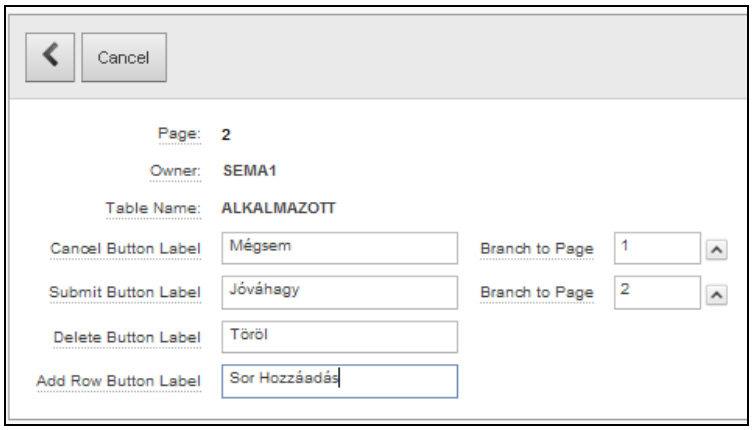
9.15. ábra *A megjelenítendő oszlopok*

21. Adjuk meg a lap nevét és a tartomány fejlécét, `Next`



9.16. ábra A Lap tulajdonságai

22. Válasszuk a létező Fül készletet (`tab set`), adjuk meg a Fül nevét, legyen ez `Módosítás`
23. `Next`,
24. Írjuk át agombokat magyarra.



9.17. ábra *A varázsló által létrehozandó gombok feliratai*

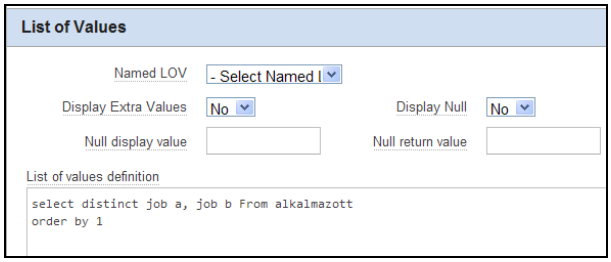
25. `Next`
26. Ellenőrizzük a választásokat és `Create`.
27. Futtassuk le (`Run Page`)

Ezzel alkalmazásunk már működöképes, de még nem magyar és nem kiválaszthatóakegyes beviendő adatokat.

Ezeket még végezzük el.

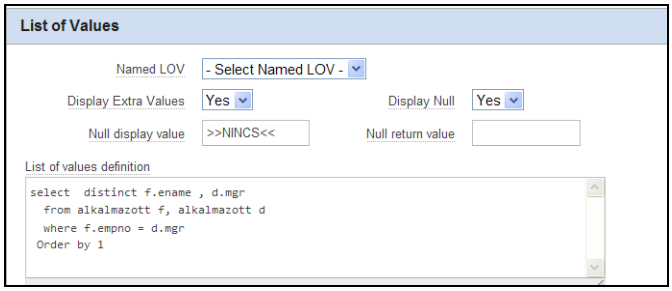
1. Alul `Edit Page 2`
2. Tartomány szekcióban a `Report` hiperlinkre kattintsunk
3. Írjuk át az oszlopok fejléceit magyarra. Kivéve az eredeti `empno`-t. Figyeljük meg, hogy automatikusan megjelent egy `Dislay_empno` oszlop is, hogy láthassuk a felvett új dolgozó azonosítóját.
4. `Applay Changes`
5. Legyen kiválasztható
    a. munkakör
    b. Főnök azonosítója helyett a meglévő főnökök nevei
    c. részleg helyett telephely
6. Ezek megvalósítása a következő
7. Kattintsunk a tartomány szekcióban a `Report` hiperlinkre, szerkesszük meg a megfelelő oszlopot. Kattintsunk Ceruza ikonra.
8. Az oszlop tulajdonságai (`Column Attributes`) szekcióban a `Display as` kiválasztó listára, és válasszuk a `Select List (query based LOV)` típusú kiválasztó Elemet.
9. Írjuk meg a megjelenő értékek forrását.

a. Munkakör esetén



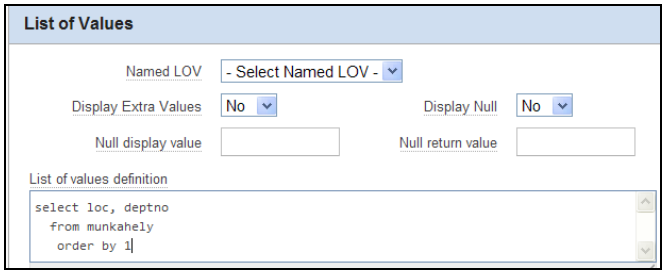
9.18.ábra *Munkakör esetén a legördülő elem forrása*

b. főnök azonosító esetén, jelenjen meg a főnök neve



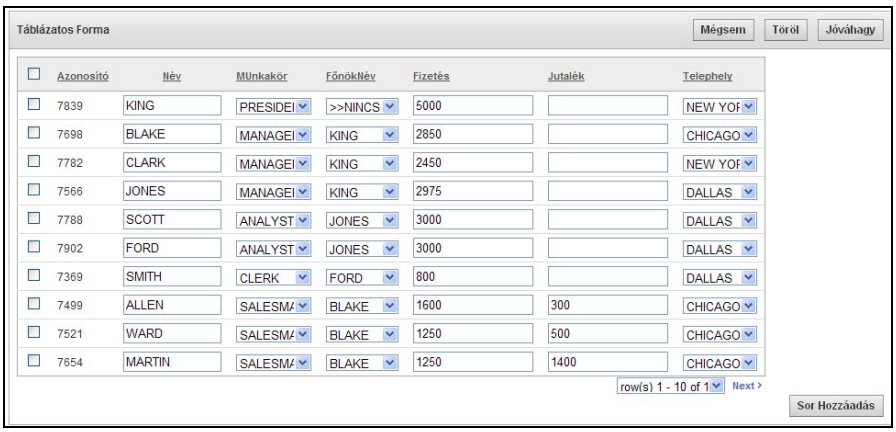
9.19.ábra *Főnöknév esetén a legördülő elem forrása*

c.Részlegazonosító helyett pedig a telephely legyen kiválasztható



9.20.ábra *Részleg esetén a legördülő elem forrása*

10. Futtassuk le.

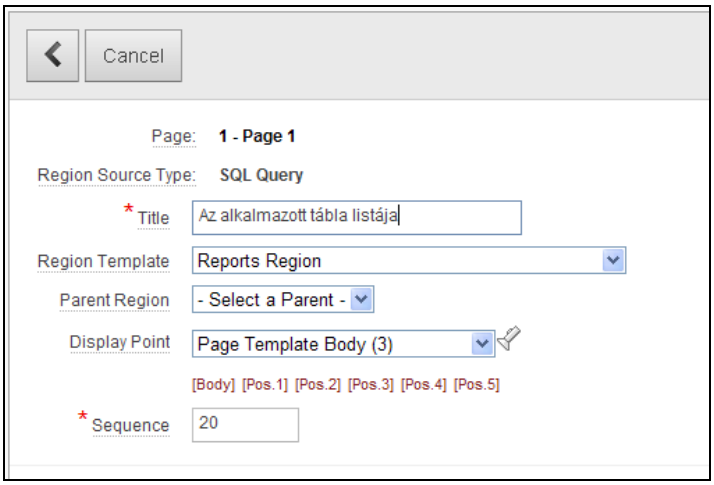


9.21.ábra *Futási lista*

Most már a Táblázatos Forma megfelelő.

Az Apex alkalmazásunk első lapján jelenítsük meg az alkalmazott tábla listáját, és ellenőrizzük, hogy a bevitt adatok benn maradnak-e a táblában.

1. `Edit Page 1`
2. Tartomány szekció `(+)` Létrehozás
3. `Report` rádiógomb választása , `Next`
4. Klaszikus riport rádiógomb, `Next`,
5. Adjuk meg a Fejléc



9.22.ábra *A lap adati*

6. Irjuk meg a forrást, `Create Region`
7. A tartomány szekcióban törljük ki a `BreadCrumb` lapot, Kattintsunk a hiperlinkre és `Delete` gomb
8. A tartomány szekcióban kattintsunk a `Report` hiperlinkre, majd írjuk át a fejléceket magyarra.

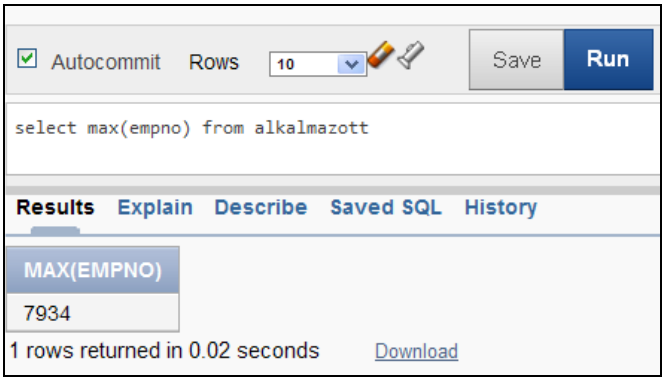
9. Futtassuk le az alkalmazást.

9.23 ábra *Az Lista lap futási képe*

Adatbeviteli trigger létrehozása a rekord azonosítására

Létre kell hozni egy triggert, amelyik a legnagyobb empno értékéhez mindig hozzáad 1-et. Ennek elkészítése a következő:

1 A fejlesztői eszköztáron válasszuk az `Edit Application` menüpontot.
2 Menjünk vissza a Visszalépő Sávon a `Home` hiperhivatkozásra kattintással.
3 Kattintsunk az `SQL` ikonra.
4 Kattintsunk az `SQL Commands` ikonra.
5 Írjuk meg a lekérdezést, amelyik meghatározza a legnagyobb azonosító értékét:
```
SELECT MAX(empno) FROM emp
```
6 Futtassuk le `Run` gombra, vagy a `CTRL Enter` gombra kattintással. (9.4. ábra.)
7 Az alábbiakban elkészítünk egy szekvenciát az `empno` oszlopra.

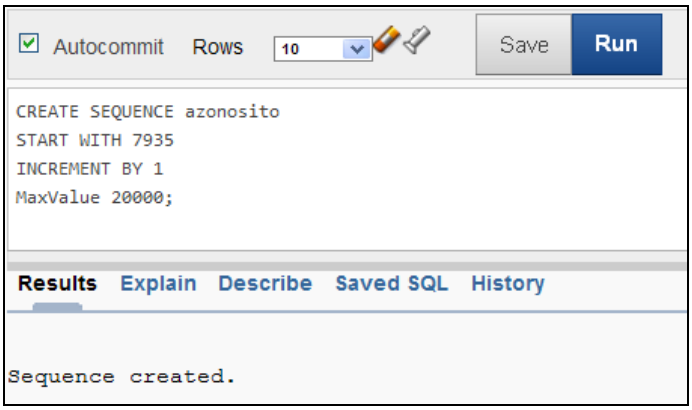


9.24. ábra. *A legnagyobb azonosító meghatározása*

Megjegyzés:

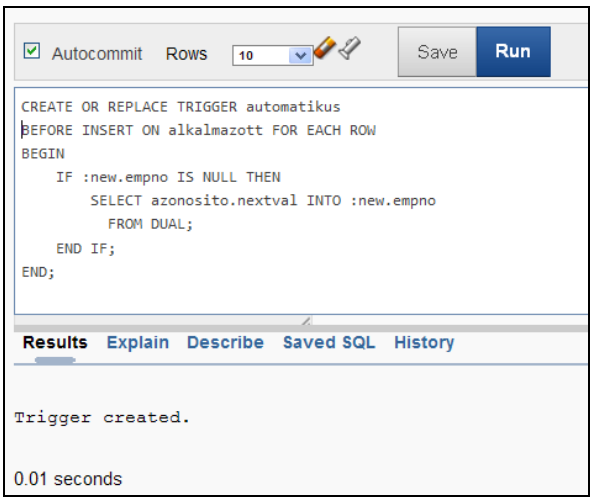
A szekvencia (`Sequence`) egy olyan sorozatot definiál, melynek segítségével egymást követő egyedi számértéket rendelhetünk a rekordokhoz (sorokhoz). A szekvencia létrehozása például a következő módon történhet (9.5. ábra):

Az alapértelmezett növekmény (`INCREMENT`) az `1`, melyet nem kötelező kiírni, és a `MaxValue` értéket sem kötelező kiírni.



9.25. ábra. *Szekvencia létrehozása*

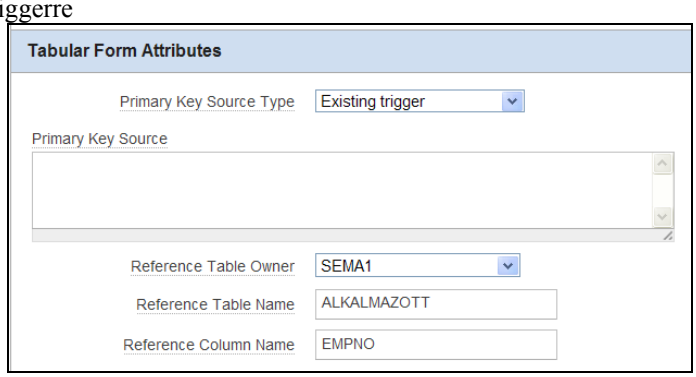
A szekvenciának két pszeudooszlopa létezik, egyik a `NextVal` (következő érték), a másik a `CurrVal` (az aktuális érték). Ezek és a `dual` tábla segítségével lekérdezhetjük, előállíthatjuk a következő azonosító értékét. Az erre való hivatkozás az `Azonosito.nextval` értékkel történik. Hozzuk létre (szintén az SQL parancsmezőben) az adatbeviteli triggert, amely előállítja az új `empno` értékét (9.6. ábra):



9.26. ábra. *A trigger létrehozása*

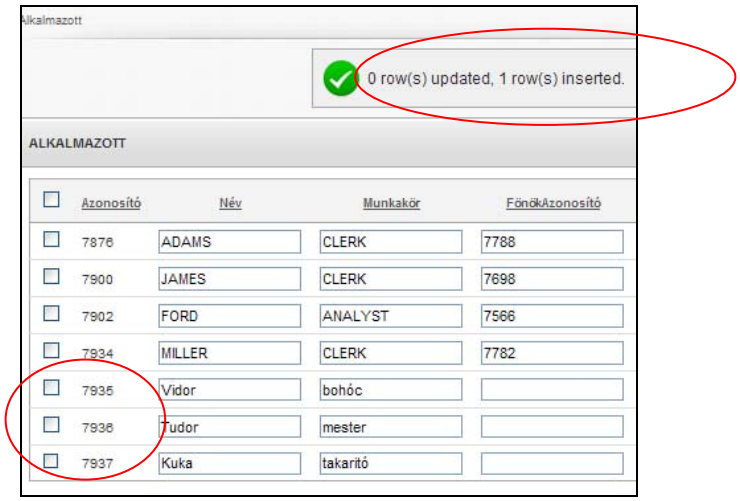
Ezzel létrejött az automatikus nevű triggerünk.

Most állítsuk át az azonosító előállításánál az elsődleges kulcs forrás típusát a létező triggerre



9.27 Létező Trigger választása, amikor az első lapon van a Táblázatos Forma

Menjünk vissza az alkalmazásra, vagyis a Visszalépő Sávon keresztül az `Application Builder`, `Tabuláló Form` alkalmazásra és futtassuk le. Kattintsunk a `SorHozzáadás` gombra. Megjelenik egy üres sor. Írjuk be az adatokat, válasszuk ki a `deptno` értéket és kattintsunk a `Jóváhagyás` gombra. Megjelenik ugyanaz a lap a következő felirattal 0 sort módosított egy sort beszúrt (9.7.ábra).



9.28. ábra. *Az új sor beviteli eredménye*

Tehát alkalmazásunk megfelelően működik.

Megjegyzés

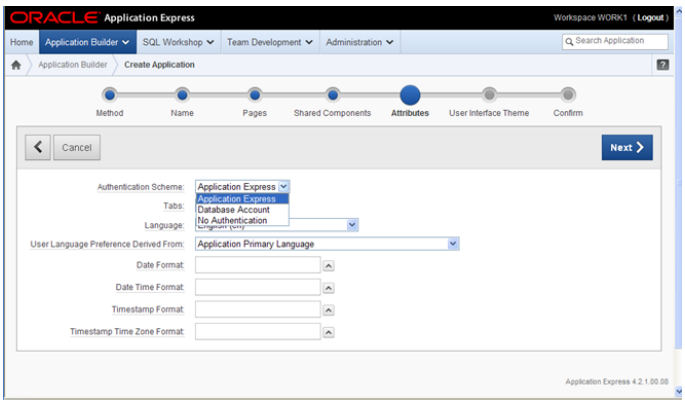
Az adatbeviteli triggert dinamikus SQL utasítással is létrehozhatjuk a tábla létrehozást követően. A gyakorlatban azonban nem ez történik. Féléves feladatnál illetve ellenőrző feladatoknál természetesen jobb a dinamikus módszer, mert könnyebb a feladatot ellenőrizni egy .sql fájl futtatásával.

10. fejezet. Hitelesítés és engedélyezés

(Hitelesítés=Authentication, Engedélyezés=authorizatiton)

Az alkalmazás létrehozásánál választanunk kell a felhasználók alkalmazás-használati jogait beállító *hitelesítési osztályok*, más néven *autentikációs sémák* (`Authentication Scheme`) közül. Az hitelesítési osztályok segítségével állíthatjuk be például a belépési lehetőségeket.

Lépjünk be az APEX környezetbe például a szokásos `magyar/Tigris-1` felhasználóként. Megjegyezzük, hogy az alábbiak szempontjából nincs jelentősége annak, hogy az alkalmazást "létrehozó" felhasználó milyen adatbázis-használati jogosultságokkal (`user privileges`) rendelkezik.



10.1. ábra. *Autentikációs séma választása az alkalmazás létrehozásakor*

Egy alkalmazás létrehozásakor – amint az a 10.1. ábrán ábrán látható – háromféle autentikációs séma adható meg:

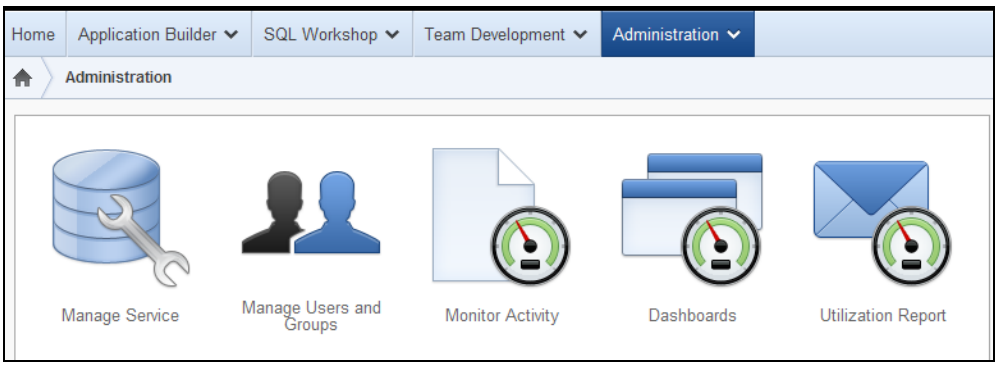
- `No authentication`: Az e sémával létrehozott alkalmazás futtatása nincs semmilyen módon korlátozva; felhasználói név és jelszó nélkül lehet elindítani az alkalmazást.
- `Database Account`: Csak rendszer-adminisztrátori (`Administration\Database Users`) szinten már definiált (létrehozott) felhasználók indíthatják el az alkalmazást. (E lehetőséggel tehát az apex felhasználók (lásd később) sem rendelkeznek.) Ezt a továbbiakban adatbázis fogjuk nevezni.
- `Application Express Authentication`: Az alkalmazásba a létrehozó felhasználón kívül csak az ő általa apex felhasználóként létrehozott felhasználók (lásd ké-

sőbb) indíthatják el az alkalmazást. (E lehetőséggel tehát a létrehozó felhasználón kívül a rendszer-adminisztrátori szinten definiált többi felhasználó sem rendelkezik.) Ezt röviden csak apex autentikációnak nevezzük.

Az eddigiekben elkészített alkalmazásaink mindegyikének autentikációja `Application Express Authentication` volt. Az alábbiakban megmutatjuk, hogy miként lehet további felhasználókat létrehozni, akik az apex autentikációjú alkalmazásokat használni (futtatni, illetve az APEX fejlesztői környezetben módosítani is) tudják.

Apex fejlesztő lehet adminisztrátor, fejlesztő és végfelhasználó.

Az Adminisztrátor menüpont funkciói. A funkciók jellemzői találhatók meg itt.



10.2. ábra. *Adminisztrátor fejlesztő lehetőségei*

A végrehajtható feladatok a lap jobb szélén található task ablakban találhatók.

Tasks		Tevékenységek
Change My Password		Saját jelszó változtatása
About Application Express		Application Express jellemzői
Manage User Groups		Felhasználói csoportok kezelése
Create User	⇨	Felhasználó létrehozása
Create Multiple Users		Több felhasználó létrehozása
Delete Multiple Users		Több felhasználó törlése
Manage Interactive Report Settings		Interaktív riport beállításának kezelése
Manage Export Repository		Export adatraktár kezelése
Websheet Database Objects		Az adatbázis objektumok weblapjai

10.3. ábra. *Adminisztrátor fejlesztő lehetséges tevékenységei*

Célunk apex felhasználó illetve felhasználói csoport létrehozása, ezért nem törekszünk a `Task` ablak teljes leírására.

Felhasználó létrehozása

Kattintsunk a `Task` ablak `Create User` menüpontra. Természetesen ne fejetsük el, hogy a `magyar` adminisztrátori felügyelet alatt vagyunk a `work1` munkaterületen a `sema1`-ben.

Legyen a felhasználó neve : `Toth`

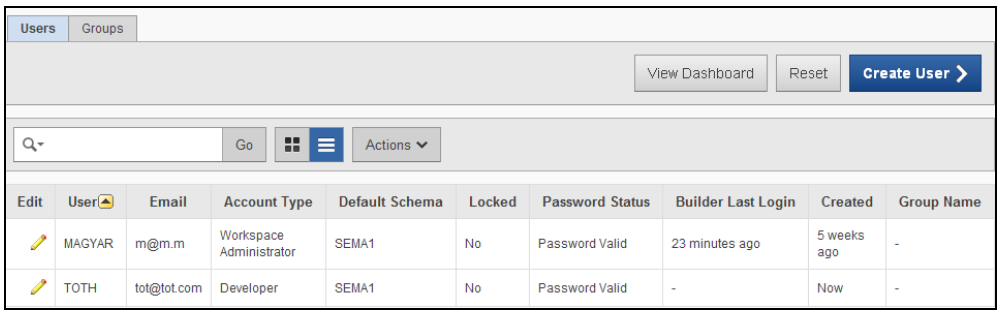
és a jogai az ábrán láthatóak,---> tehát síma fejlesztő lesz.

Account Privileges

Default Schema	SEMA1
Accessible Schemas (null for all)	
User is a workspace administrator:	○ Yes ◉ No
User is a developer	◉ Yes ○ No
Application Builder Access	Yes
SQL Workshop Access	Yes
Team Development Access	Yes
Set Account Availability	Unlocked

Password: `Tigris-2`

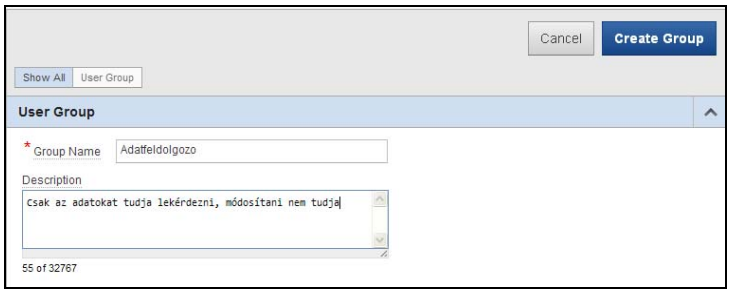
Hozzuk létre a felhasználót, `Create User`



10.2. ábra. *Az apex fejlesztők*

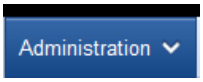
Figyeljük meg, hogy itt módosíthatjuk a felhasználók jellemzőit a (`ceruza`) ikonra kattintva, törölhetjük is a fejlesztőket. Utólag is hozzárendelhetjük a már létező csoporthoz.

Megjegyezzük, hogy apex felhasználói csoportokat is létrehozhatunk (`Manage User Groups`), és az egyes felhasználókat a megfelelő csoportokhoz rendelhetjük (például Főnök és Beosztott vagy Vezeto, Adatfeldolgozo).

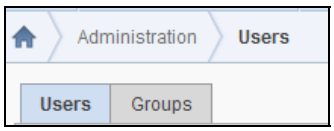


10.3. ábra. *Az apex fejlesztői csoport létrehozása*

`Create Group` gombra kattintva a csoport létrejön.
Rendeljük az Adatfeldolgozo csoporthoz a létrehozott `Toth` nevű felhasználót.

- Nyissuk meg az Adminisztration fül melletti nyilat 

- Kattintsunk a `Manage Users and Groups` menüpontra, Itt módosíthatjuk létrehozhatjuk a felhasználókat, illetve a másik fülön a csoportokat.



10.4. ábra. *Felhasználó, és csoportkok kezelése*

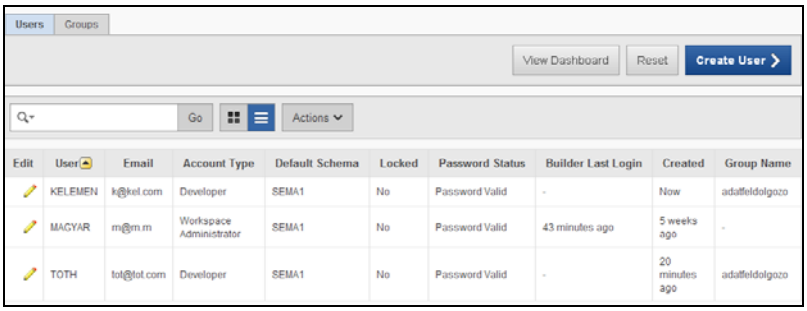
- Kattintsunk a `Toth` melletti ceruza ikonra, menjünk lefelé az utolsó részig, mert itt található a `Users Group` ablak. Helyezzük át az Adatfeldolgozo csoportot a nyíl segítségével jobbra, majd menjünk fel és érvényesítsük (`Applay changes`) a változást.

Hozzunk létre még egy `Kelemen/Tigris-3` nevű felhasználót. `Toth` és `Kelemen` felhasználó tartozzon egy csoportba (`Adatfeldolgozo`)

- Felhasználó létrehozása (`Create User`)
- Adjuk meg a nevet és az e-mail címet (`Kelemen`)
- Válasszuk a felhasználó legyen fejlesztő rádiógombot,
- Jelszó legyen. `Tigris-3`
- A csoport itt már eleve adott , de ezt ki is kell jelölni, ha azt szeretnénk, hogy a csoporthoz tartozzon.

10.5. ábra A felhasználó létrehozásakor már a csoporthoz rendeljük

Kattintsunk a `Manage Users and Groups` gombra., és válasszuk a `Users` fület



10.6. ábra. *A két felhasználó létrejött*

Mint láthatjuk, névsorba rendezve megjelenik az apex adminisztrátor felhasználó, és a két fejelsztő is.

`Logout`.

Lépjünk be `Kelemen/Tigris-3` felhasználó munkaterületére. Mint láthatjuk elsőre , az összes - `Magyar` által létrehozott- alkalmazást látja.

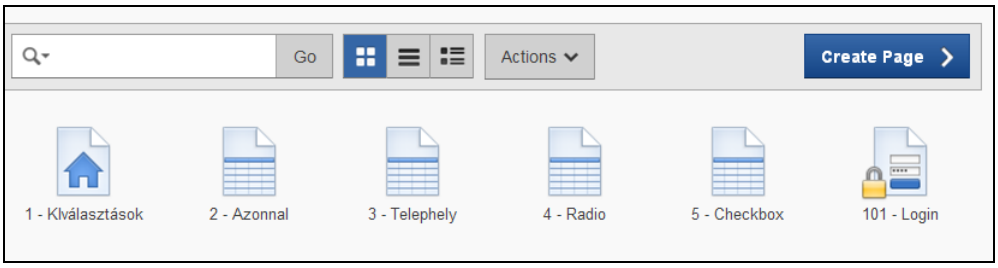
`Toth` és `Kelemen` felhasználó látja az eddig létrehozott összes alkalmazást. El tudja indítani, módosítani és törölni is tudja.

A `Toth` felhasználó kiegészítette az alkalmazást adatbevitelei összetett lehetőséggel. Lépjünk ki. Léjünk be `Magyar` felhasználóként, majd `Kelemen` felhasználóként, azt veszszük észre, hogy mind a három fejlesztő használni , és módosítani tudja az alkalmazást. Az alkalmazás neve Jogok_kelemen

Az alkalmazás hitelesítésének megváltoztatása

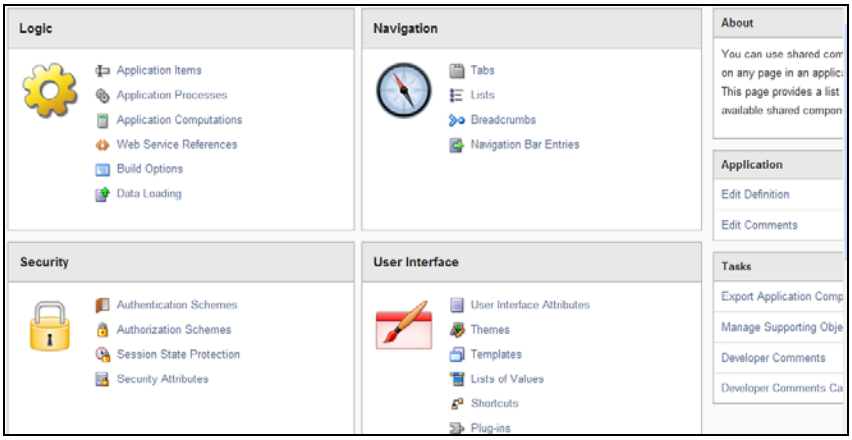
(authentication)

Tekintsük a Toth fejlesztő által kiegészített, immár 8 lapból álló 32_Kiválasztások alkalmazást, (10.7. ábra), amely tehát apex autentikációval rendelkezik.



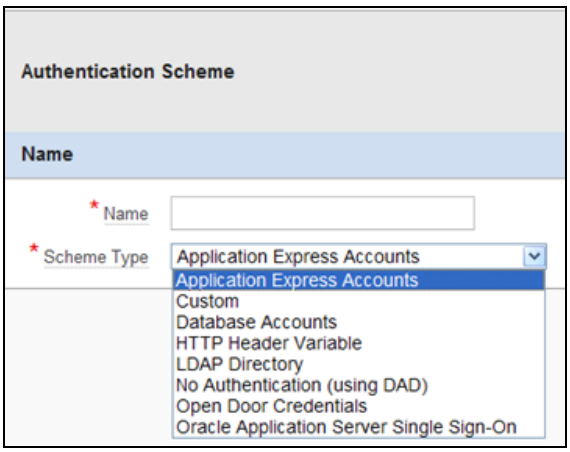
10.7. ábra. *Egy 6-lapos alkalmazás*

Kattintsunk a Megosztott Komponensek (Shared Components) ikonra, melynek eredményeként megjelenik az összes megosztott komponens (10.8. ábra).



10.8. ábra. *A megosztott komponensek*

A megosztott komponensek ablakban kattintsunk a Biztonság (Security) rész Engedélyezési osztály (Authentication Schemes) menüpontra. Itt létrehozhatunk engedélyezési osztályokat a Létrehozás (Create) gomb segítségével. A létrehozás folyamatában a galériából választhatunk, vagy már létező engedélyezési osztályt másolhatunk. Mivel nekünk apex autentikációra van szükségünk, ezért ezzel részletesen nem foglalkozunk.



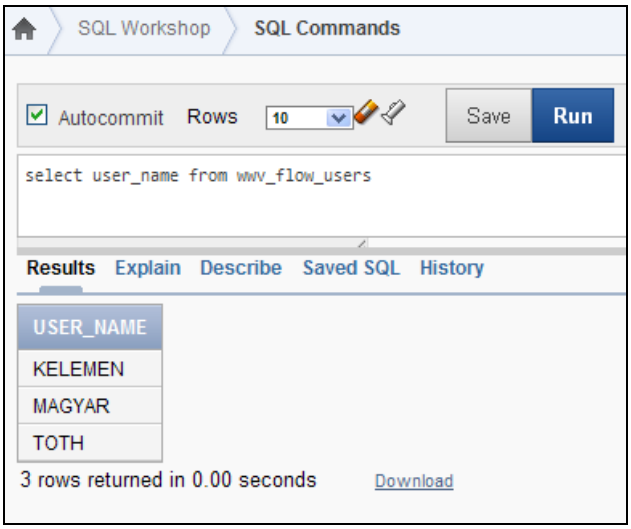
10.9. ábra. A létező autentikációs sémák

Engedélyezési osztály létrehozása

(Authorization Schema)

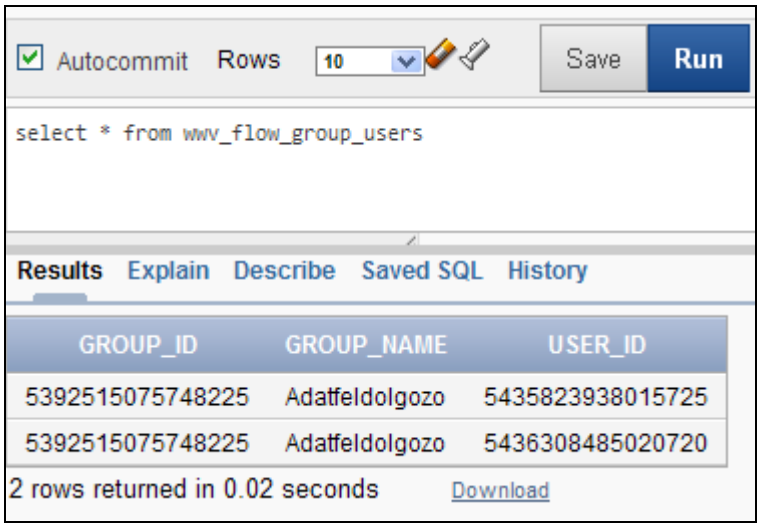
Ahhoz, hogy létre tudjuk hozni az engedélyezési mintát, tudnunk kell, hogyan hivatkozhatunk a felhasználókra, a felhasználói csoportokra.

- Az alkalmazásban az apex fejlesztő környezet eltárolja a felhasználó nevét egy `APP_USER` nevű globális változóban.
- Az Oracle adatbázis-kezelő a felhasználókat a `wwv_flow_users` táblában tárolja el (10.10. ábra.). A tábla lekérdezhető.



10.10. ábra. *Az Oracle-ből lekérdezett felhasználók*

- A felhasználói csoportokat a `wwv_flow_group_users` táblában tárolja el (10.11. ábra.). Ez a tábla is lekérdezhető.



10.11. ábra. *Az Oracle-ből lekérdezett felhasználói csoportok*

Példa

Feladatunk engedélyezési osztályok létrehozása a kiválasztások többlapos alkalamzásban
1.) hogy a `Magyar` fejlesztő minden laphoz hozzáfér
2:) `Toth` felhasználó a harmadik lapot (`Telephely`)nem látja
3.) `Kelemen` felhasználó pedig a negyedik (`Radio`) lapon a rádiógombokat nem is látja
4.) Hozzunk létre egy `Vezeto` csoportot, akihez még Tudor fejlesztő is tartozik.
Módosítani csak a `vezető` csoporthoz rendelt fejlesztők tudjanak, tehát az Adatfeldolgozó csoporthoz tartozók nem. . Nézzük meg az eredeti alkalmazást.(10.12. ábra.).

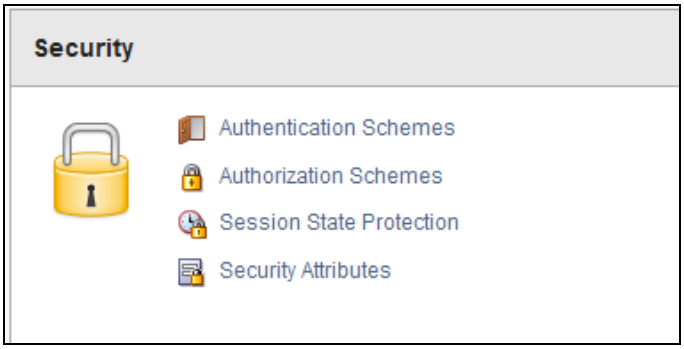
10.12. ábra. *A harmadik lap*

1. LÉPÉS
Mivel mindhárom felhasználó apex autentikációval rendelkezik, így az alkalmazásba mindhárman be tudnak lépni.

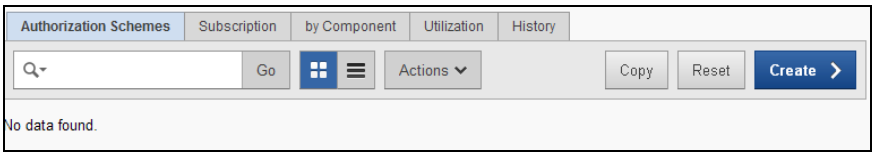
2. LÉPÉS
A `Toth` felhasználó ne láthassa a harmadik lapot (`Telephely`)(10.12. ábra).

- Hozzunk létre Engedélyező (`authorization`) osztályt.

Kattintsuk az alkalmazásunkra, majd a Megosztott Komponensek (`Shared Components`) ikonra, majd a Titkosítás (`Security`) ablakrészben az Engedélyezési minta (`Authorization Schemes`) menüpontra (10.13.ábra.) megjelenik az engedélyezést létrehozó ablak (10.14. ábra).



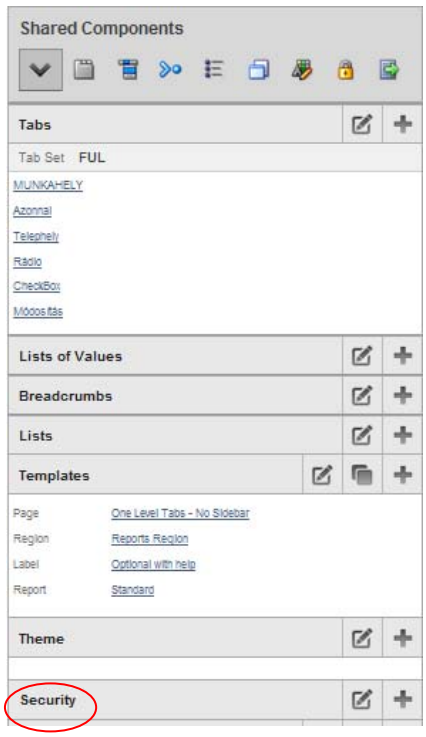
10.13. ábra. *Biztonság ablak*



10.14. ábra. *Engedélyezés létrehozásának első lépése*

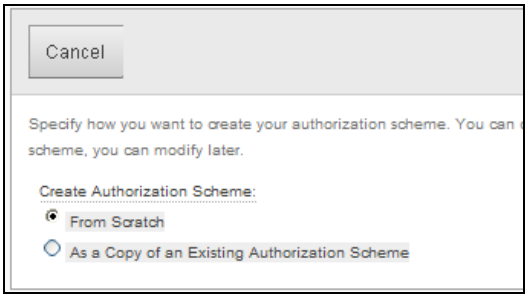
Megjegyzés

A 10.14-es ábrához más úton is eljuthatunk.
Kattintsunk bármelyik lapon a lap szerkesztésébe és itt a a Megosztott Komponensek (jobboldali oszlop) Biztonság (`Security`) részben a Létrehozsás (+) jelre kattintunk (10.13. ábra).



10.15. ábra. Engedélyezés létrehozásának másik induló útja

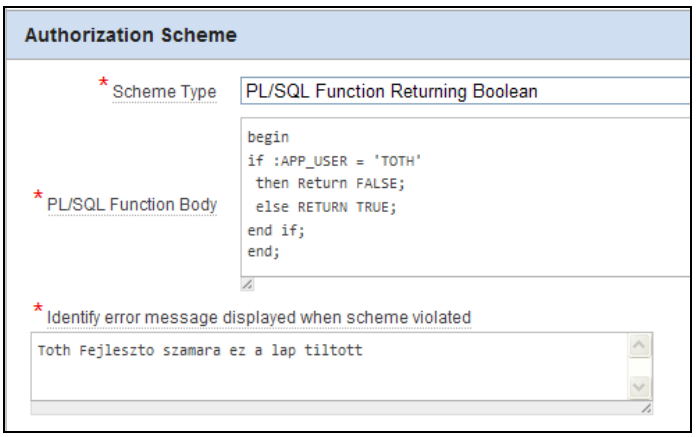
Válasszuk a kiindulástól (`From Scratch`) rádiógombot (10.16. ábra), és lépjünk tovább (`Next`).



10.16. ábra. Második lépés

Adjuk meg a kötelező adatokat (10.14. ábra):

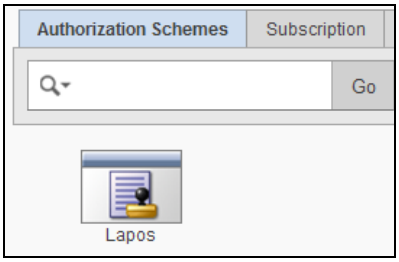
- Válasszunk nevet (`Name`) a mintának (`Lapos`)..
  (Feladatunk az hogy a harmadik lapot Toth elől letiltsuk. (10.17. ábra)).
- Válasszuk ki a Minta típusát (`Scheme Type`), ez legyen `PL/SQL Function Returning Boolean`.



10.17. ábra. *Harmadik lépés*

- Írjuk meg a függvényt (10.17. ábra), amely azt valósítja meg,
- Feltétlenül meg kell adni a hibaüzenetet is (`Identify error message`…), majd kattintsunk a Létrehozás (`Create Authorization Scheme`) gombra.

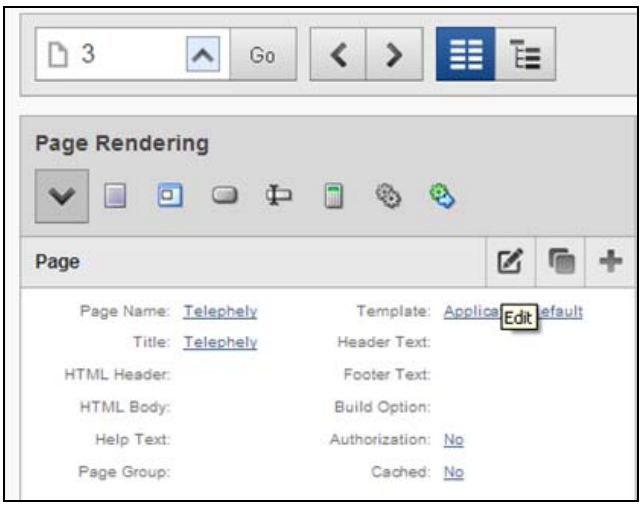
Létrejött a Lapos nevű Engedélyező (`authorization`) osztály (10.18. ábra).



10.18. ábra. *Az engedélyező lap ikonja* `lap`

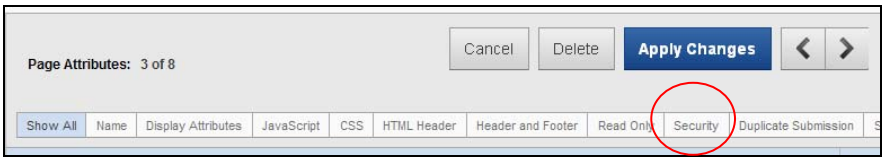
Ez azonban még **nem elegendő** ahhoz, hogy a lapot aToth felhasználó ne érje el. Szükséges még az adott (`Telephely`) laphoz is hozzárendelni a `lapos` nevű engedélyezőosztályt.

- Menjünk az alkalmazás harmadik lapjára. Kattinsunk az `Edit Page 3` gombra majd az `Edit` ikonra..



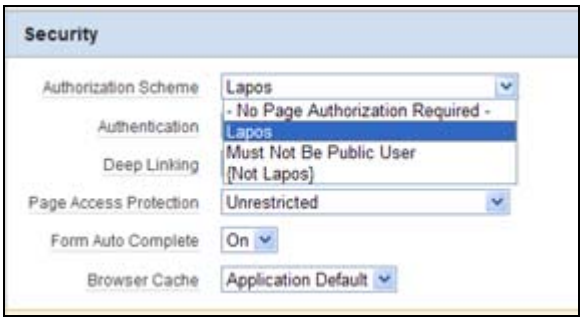
10.19. ábra. *A lap szerkesztési folyamatai*

- A szerkesztő lap tetején (10.20. ábra) kattintsunk a Biztonság (`Security`) menüpontra, vagy gördítsük le az oldalt a `Security` ablakrészig.



10.20. ábra. *A lap szerkesztési folyamatának következő lépése, a Biztonság ablak elérése*

- Itt válasszuk ki az Engedélyezési minták (`Authorization Scheme`) legördülő mezőből a most létrehozott `Lapos` nevű mintát (10.21. ábra), és alkalmazzuk a változásokat.



10.21. ábra. Engedélyezési minta választás

- Lépjünk ki `logout` segítségével, majd indítsuk el a programot `Toth` felhasználó nevével és jelszavával (`Tigris-2`). Menjünk a harmadik lapra. Itt a biztonság miatt a

`Toth` már nem láthatja ezt az lapot, csak az általunk megadott hibaüzenetet (10.22. ábra).

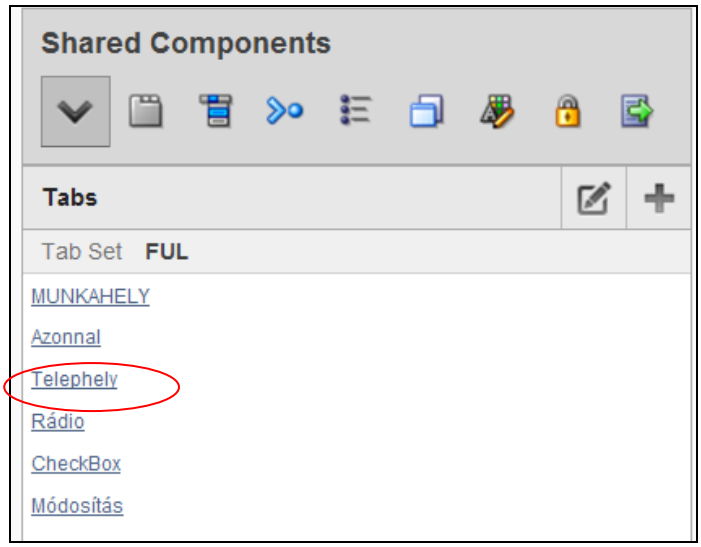
10.22. ábra. *Az általunk megadott hibaüzenet, ha a felhasználó Nagy*

Tehát ezt az lapot a `Toth` felhasználó valóban nem láthatja.

Fül letiltása

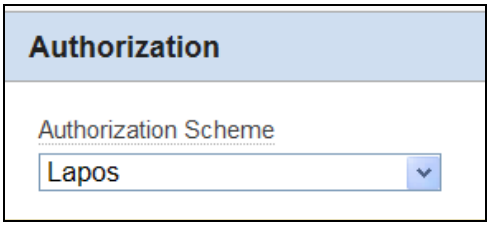
A fül objektum esetében is beállíthatjuk ugyanezt a sémát, és akkor a harmadik lap füle egyáltalán meg sem jelenik a `Toth` felhasználó számára (de ekkor persze nem láthatjuk a hibaüzenetet).

- Szerkesszük alul harmadik lapot (`Edit Page 3`).
- A szerkesztő lap jobb oldalán a megosztott Komponensek Fül (`Tabs`) részében kattintsunk a `Telephely` nevű fül hiperhivatkozásra (10.23. ábra).



10.23. *A Megosztott Komponens Flüek l ablaka*

- A megjelenő szerkesztő lapon keressük meg az Engedélyezés (`Authorization`) ablakrészt, és a gördülő menüből válasszuk ki (10.24. ábra) az általunk létrehoztt `Lapos` nevűt, majd alkalmazzuk a változtatásokat.



10.24. *Az Engedélyezési Minták*

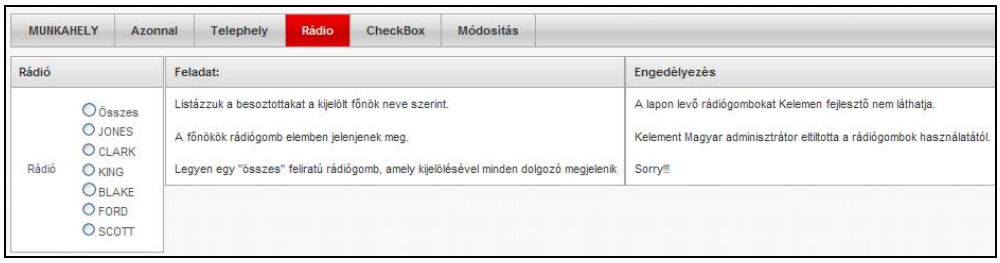
- Futtassuk le ismét az első laptól kezdve (természetesen `Toth` felhasználó indításával.

10.25. ábra Toth *felhasználó előtt a harmadik fül meg sem jelenik*

3. LÉPÉS

A következő lépés annak biztosítása, hogy a `Kelemen` felhasználó egy lapon belül ne érjen el egy **objektumot**, jelen példában a negyedik lapon egy `rádiógroup` csoportot.

Nézzük meg az eredeti lapott, ahogyan azt a `Magyar` fejlesztő látja (10.26. ábra).



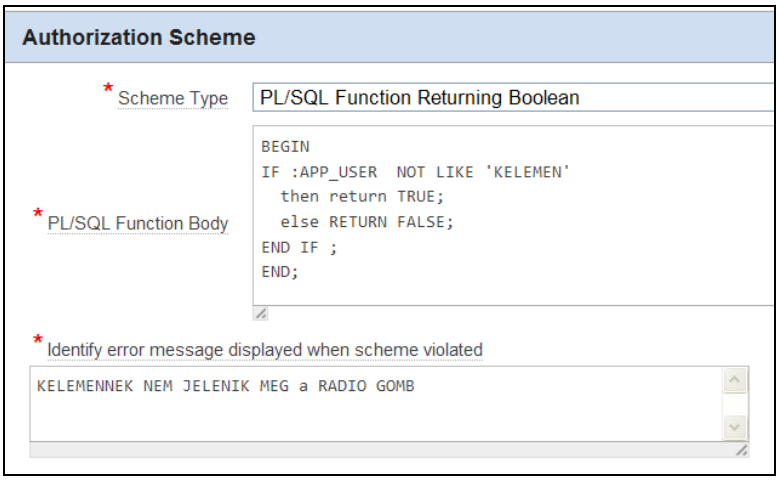
10.26. *A második oldal tartalma*

Először létre kell hozni egy új engedélyezési osztályt (`authorzization`).

Az előbbiek alapján, elismételve:

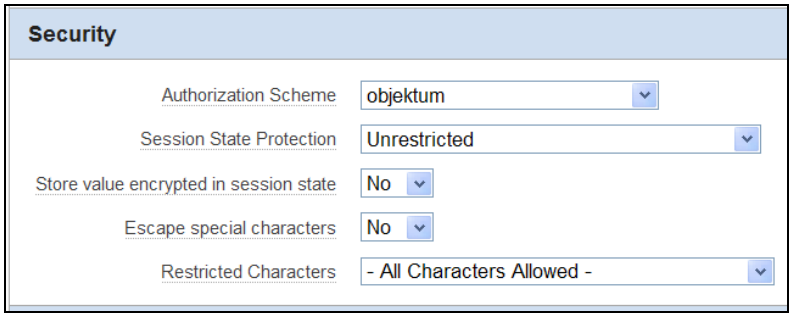
- Kattintsunk az `alkalmazásunkra`, ezután a
- Megosztott komponensek (`Shared Components`),
- Biztonság (`Security`) ablak,
- Engedélyezés (`Authorization Schemes`) menüpontra.
- Majd Létrehozás (`Create` gomb) ,
- Kiindulási (`From Scrath`) gomb, Next.

- A létrehozandó minta neve `objektum` legyen, és típusa szintén `PL/SQL function Returning Boolean.`
  A függvény pedig a következő (10.27. ábra)



10.27. ábra. *Objektum letiltása Kelemen számára*

- Ne felejtsük el megírni a hibaüzenetet!
- Végül hozzuk létre a mintát (`Create Authorization Scheme`)
- Itt se feledkezzünk meg arról, hogy az adott lap adott objektumánál, jelen esetben a rádiógomboknál beállítsuk az engedélyezést illetve tiltást. Menjünk a `4.` lapra, szerkesszük a rádiógombot, azaz kattintsunk a `P4_RADIO Rad(...)` (10.28. ábra) hiperlinkre, majd kattintsunk a `Security` menüpontra, vagy gördítsük lefelé a lapot a `Security` ablakig.



10.28. ábra *Engedélyezési osztály kiválasztása*

- Az engedélyezési osztály (`Athorization Scheme`) legördülő mezőben válasszuk ki az objektum osztályt (1.29. ábra)
- `Applay Changes`

Lépjünk ki `logout` segítségével, majd lépjünk be `Kelemen/Tigris-3` felhasználóként, és futtassuk le az alkalmazást. Láthatjuk, hogy a negyedik lapon a rádiógombok csoportja nem jelenik meg (10.29. ábra.), amikor Kelemen belépett.

10.29. ábra. *Kelemen nem használhatja a rádiógombokat*

4. LÉPÉS

Ha csak a `Magyar` fejlesztő számára érhető el a módosítás lap, akkor feladatunk egyszerű a, már ismert engedélyezések alapján.

Készítünk egy új engedélyezési mintát és ezt hozzárendeljük a 6. laphoz.
Legyen a neve `Modosit`.

Létrehozása ismételve az előbbieket:
- Megosztott komponens,
- Biztonság (Security),
- Engedélyezési minta,
- Create gomb. .
- From Scratch rádiogomb, Next,

Az engedélyezési minta ablakban :
- Nevet adunk
- Minta Típust választunk, PL/SQL Function Returning Boolean
- Megírjuk a PL/SQL ....
- Megírjuk a hibaüzehetet
- Create

Authorization Scheme

* Scheme Type: PL/SQL Function Returning Boolean

* PL/SQL Function Body:

```
BEGIN
  IF :APP_USER = UPPER('MAGYAR')
     then RETURN TRUE;
     else RETURN FALSE;
  end IF;
End;
```

* Identify error message displayed when scheme violated

```
Csak a fejleszto adminisztrator modosithtaja
```

10.30. ábra Csak a Magyar fejlesztő tud módositani

Felhasználói Csoport kezelése

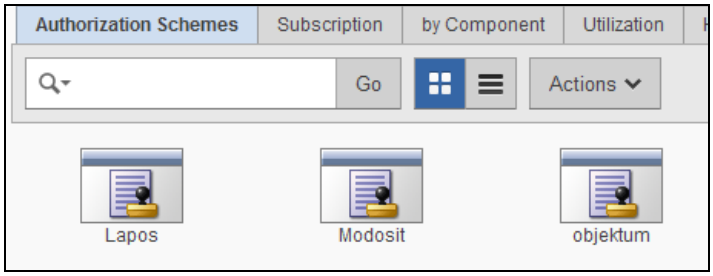
Amennyiben a feladatunk a vezetői csoport engedélyezése,

- Létre kell hoznunk a már ismert módon egy `Vezeto` csoportot,
- Létrehozunk még egy felhasználót (legyen `Tudor/Elefant`).
- Mindkettőt a `Vezeto` csoporthoz rendeljük, és akkor készítünk egy új engedélyezési osztályt.

Edit	User	Email	Account Type	Default Schema	Locked	Password Status	Builder Last Login	Created	Group Name
	KELEMEN	k@kel.com	Developer	SEMA1	No	Password Valid	4 hours ago	4 hours ago	adatfeldolgozo
	MAGYAR	m@m.m	Workspace Administrator	SEMA1	No	Password Valid	21 seconds ago	5 weeks ago	VEZETO
	TOTH	tot@tot.com	Developer	SEMA1	No	Password Valid	2 hours ago	4 hours ago	adatfeldolgozo
	TUDOR	TUD@ t.com	Developer	SEMA1	No	Password Valid	-	8 minutes ago	VEZETO

10.31. ábra Vezeto és Adatfeldolgozo csoport tagjai

Mint az ábrán láthatjuk két vezetőnk van (`Magyar`, `Tudor`) és két Adatfeldolgozo (`Toth` és `Kelemen`).

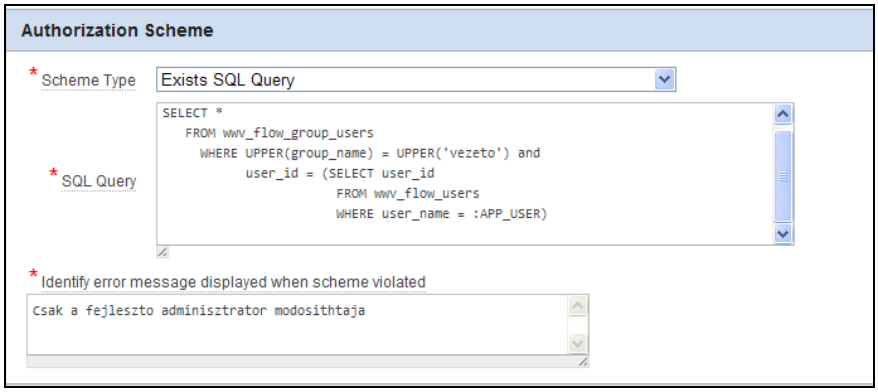


10.32. ábra Létrehozott engedélyezési osztályok ikonjai

- Mivel van már egy `Modosit` engedélyezési mintánk, ezt változtassuk meg csoportok engedélyezésére. A csoport használata azért jobb, mert ha felveszünk a céghez egy új alkalmazottat, akkor csak egyszerűen társítjuk az adott csoporthoz, más egyéb teendőnk nincs.

- Kattintsunk a `Modosit` ikonra.
- Keressük meg az Engedélyezési Osztály ablakot
- Osztály típusa legyen : `Exists SQL Query`
- Írjuk meg a lekérdezést:

Válasszuk ki a csoportokból a `vezeto` csoportot, azonban ez nem elegendő. Szükség van És kapcsolatban még feltételre. A feltétel a csoporthoz tartozó `user_id` azonos legyen az alkalmazást indító felhasználó `user_id` számával, amelyet a nevével való egyezőség alapján választunk ki.



10.33. ábra A módosító csoport (Vezeto) tagjailétezésének ellenőrzése

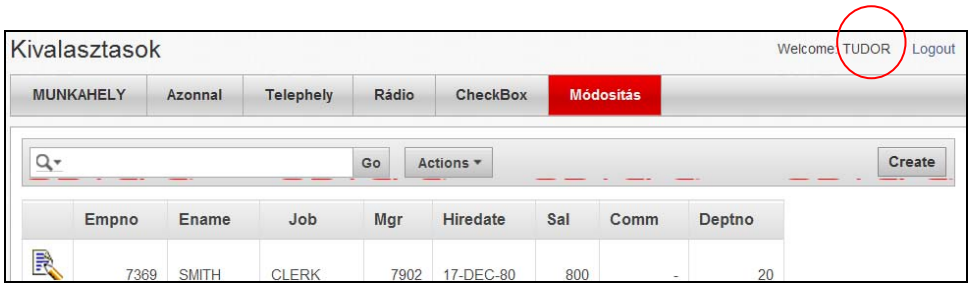
- Irjuk meg a hibaüzenet azonosítását
- Futtassuk le

10.34. ábra Toth fejlesztő nem tud belépni

Próbálkozásaink során, **futtatáskor** lépjünk ki egyik felhasználótól, lépjünk be másik felhasználó jelszavával, és így indíthatjuk el az alkalmazást.

-Lépjünk ki

-Lépjünk be `Tudor/Elefant` jelszó párossal



10.35. ábra Toth fejlesztő

Láthatjuk, hogy `Tudor` képes már adatokat módosítani.

11. fejezet. Összefoglaló feladat

Példa

Készítsen egy többoldalas KFT nevű APEX-alkalmazást az alábbi feladatokra. Ebben a **feliratok és az oszlopnevek magyar** nyelvűek legyenek, és alkalmazzon **hibakezelést**.

1. Az alkalmazás első, **Bemutatkozo** nevű lapja minden felhasználó számára közvetlenül (tehát belépési név/jelszó páros nélkül) legyen elérhető.
2. A **Bemutatkozo** lapon rövid bemutatkozó szövegként szerepeljen a saját neve, e-mail címe és a cég tevékenységének rövid ismertetése. Ez alatt jelenjen meg a `dept` táblával struktúra-, megszorítás-, és adattartalom szinten egyező, de az Ön **keresztnevével** rendelkező tábla. Ennek bármely soráról legyen elérhető a **Drill Down** eszköz segítségével az `emp` táblával struktúra-, megszorítás-, és adattartalom-szinten egyező, de az Ön **vezetéknevével** rendelkező tábla megfelelő résztáblája a **Dolgozo** nevű lapon. (E táblák létrehozását egy **process**-ben végezze el!)
3. Szintén a **Bemutatkozo** lapon legyen egy **Belépés a Karbantartó lapra** nevű bejelentkezési gomb, mellyel a karbantartói csoport tagjai (`Varga/Tigris-2`) jelszóval beléphet és karbantarthatja a lapot (felvehet, törölhet és módosíthat alkalmazottat). A karbantartókon kívül senki más nem érheti el a ezt a lapot, tehát sem magyar, sem az adatkezelői csoport bármelyik tagja a `Kelemen/Tigris-3` belépési-név/jelszó páros megadásával nem érheti el a karbantartási munkákat lehetővé tevő **Karbantarto** lapot.
4. A **Karbantarto** lapon a **Tabuláló Form** eszköz segítségével legyen lehetőség az Ön **vezetéknevével** rendelkező (`emp`-ekvivalens) táblán adatbevitelt, módosítást és törlést végezni. E táblán
   – jelenjenek meg az `empno` értékek is,
   – az új rekord felvitelénél az `empno` érték legyen automatikusan 1-gyel nagyobb az addigi legnagyobb értéknél, és
   – alkalmazzon kiválasztó elemeket (legördülő menüket) a munkahely, a főnöknév, a telephely, és a dátum adatoknál.
   E lapon legyen egy **Visszalépés** nevű gomb, melynek segítségével vissza lehet jutni a **Bemutatkozo** nevű lapra.

Megoldás

A feladat megoldó személy neve legyen Sas Pál.

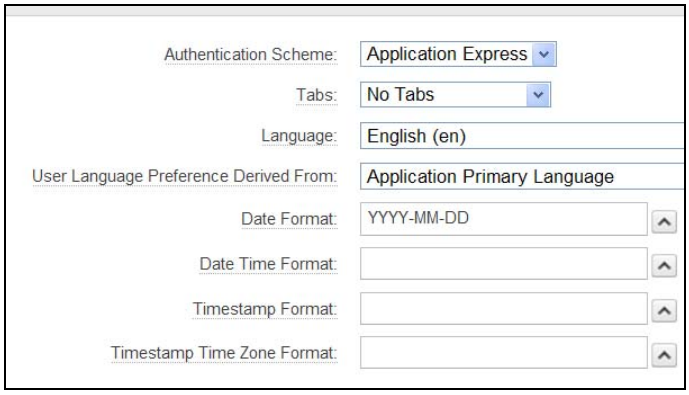
A `Bemutatkozo` lap létrehozása

1. Egy új alkalmazás létrehozása

Az alábbiakban bemutatjuk a KFT nevű alkalmazás létrehozását. Először hozzuk létre az első két lapját két üres (`Blank` típusú) lappal.

- Az APEX fejlesztő eszköz elindítása után a korábban létrehozott `magyar/Tigris-1`)felhasználóként való belépéssel (az alkalmazásfejlesztő eszköz (`Application Builder`) kiválasztása után kattintsunk a `Create` gombra
- A `Database` rádiógomb kijelölése, `Next`
- Az alkalmazás nevének megadás; `Name`: `KFT`

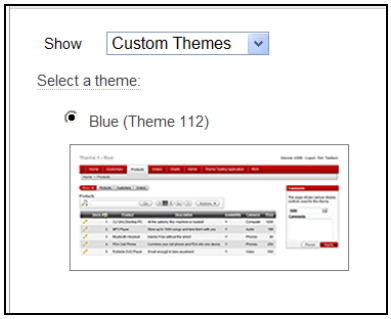
  `Create Option` : Válasszuk a `Start from scratch`, `Next`
- `Blank` típusú lap kiválasztása, `Page Name : Dolgozo, Add Page gomb,`
- Az első lap neve (`Page Name`) legyen `Bemutatkozo`, majd kattintsunk az `Add Page` gombra.
- A második lap neve legyen `Dolgozo`, majd `Add Page`, és `Next`
- `Shared Components` legyen `No` rádiógomb,
- A jellemzők (`Attributes`) ablakban válasszuk belépési jogosultságként az apex autentikációt (`Application Express Authentication`)
- Válasszuk Fül nélküliséget `Tabs: No Tabs` , nyelv maragjon angol
- Válasszunk megfelelő dátumformátumot



11.1. ábra. *Az alkalmazás főbb jellemzői*

- `Next`,
- Felhasználói mintaként válasszuk a 112-as jelűt, `Next`

- Az eddigi beállítások jóváhagyásaként a `Create Application` gombbal hozzuk létre a `KFT` nevű alkalmazásunkat (11.2. ábra).

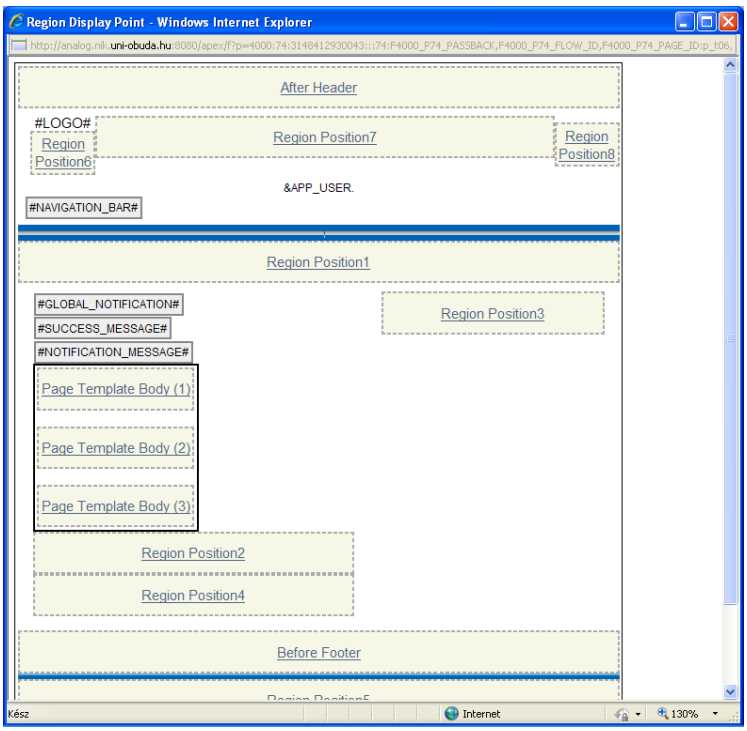


11.2. ábra. *A létrejöt tKFT nevű alkalmazás*

2. Szerkesszük meg a `Bemutatkozo` lapot

Most ismertetjük a `Bemutatkozo` lap cégismertető szövegrészének elkészítését.

- Kattintsunk a `Bemutatkozo` lapra
- A megnyíló ablakban Lap-Megjelenítési (`Page Rendering`) csoportjának Tartomány (`Regions`) szekciójában a jobboldali (+) jelre való kattintással hozzunk létre egy új `HTML` tartományt, `Next`
- Megint HTML és `Next`,
- Írjuk be a bemutatkozó szöveg címet (`Title:Bemutatkozás`), válasszuk ki a megjelenítés helyét a lap tetején (`Page Template Body 1`), (11.3. ábra). Figyelműkbe ajánljuk, hogy minden Mintának más és más az elrendezése, ezért ezt mindig nézzük meg az elemlámpa segítségével. `Next`



11.3. ábra. A HTML tartomány, neve helye

- Írjuk meg a cégismertetőt a felkínált szövegmezőben, `Create Region`, majd futtassuk le.
- Futtatás után a lap a 11.4. ábra szerinti.

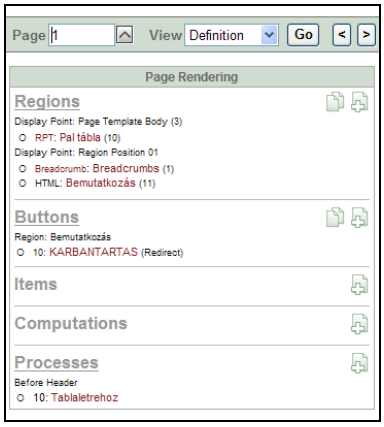
11.4. *A bemutatkozo lap*

- Következő lépésünk a "lefúráshoz" a `Pal` tábla megjelenítése, azonban még nincs semmilyen táblánk. Ezért először egy folyamat oldali process segítségével létre kell hoznunk a táblákat, melyek az `emp` és `dept` táblával megegyező struktúrával és adattartalommal rendelkeznek.

3. A táblalétrehozó folyamat elkészítése

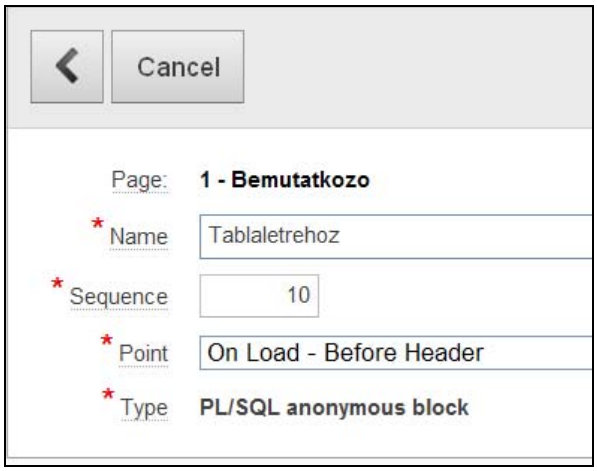
Az alábbiakban létrehozzuk az `emp` táblából a `Sas` táblát, a `dept` táblából pedig a `Pal` táblát.

- Fejlesztő eszköztár (a lap alján); `Edit Page 1`
- Az első lapon hozzunk létre egy folyamatot (`Page Rendering`, `Processes`, (+), 11.5. ábra)



11.5.ábra. *A szerkesztett első lap*

- Legyen a folyamat PL/SQL, `Next`
- Irjuk be a folyamat nevét (`Tablaletrehoz`), majd a folyamat végrehajtási pontját válasszuk ki (`On Load Before Header`, 11.6.ábra), melynek hatására a folyamat a lap betöltésekor, de a fejléc megjelenése előtt hajtódik végre, `Next`



11.6. ábra. *A folyamat adatainak beállítása*

- A szövegmezőbe írjuk be az alábbi PL/SQL programot (a táblalétrehozó és megszorítást hozzáadó eljárást) (11.7. ábra), `Next`

Source

* Process [Download Source]

```
declare
  sastablavan   number(1);
  paltablavan   number(1);

begin
select count(*)
  into sastablavan
  from user_tables
  where table_name = 'SAS';
select count(*)
  into paltablavan
  from user_tables
  where table_name = 'PAL';

if paltablavan != 1 then
  execute immediate 'Create table Pal as select * from dept';
  execute immediate 'Alter table Pal Add Primary Key (deptno)';
end if;

if sastablavan != 1 then
  execute immediate
    'Create table Sas as select * from emp';
  execute immediate
    'Alter table Sas Add Primary Key (empno)';
  execute immediate
    'Alter table Sas Add Foreign Key (deptno) References Pal (deptno)';
  execute immediate
    'Alter table Sas Add Foreign Key (mgr) References Sas (empno)';
end if;
end;
```

11.7. ábra *A táblalétrehozó folyamat*

- Írjuk meg a sikeres (`Success`) és a nem sikeres létrehozáshoz (`Failure`) tartozó üzeneteket (NE ÍRJUNK ÉKEZETES betüket), `Next`
- A fenti, táblalétrehozási üzeneteket generáló PL/SQL program akkor NE fusson csak le, ha mindkét tábla létezik. Ezt a feltételt állítjuk itt be oly módon, hogy a feltétel tipusa (`Condition Type`) legyen "`PL/SQL function body Returning a Boolean`", a feltétel vizsgálatát megvalósító PL/SQL függvény törzse (`Expres-`

`sion 1`) pedig legyen az alábbi. Ennek beírása után kattintsunk a `Create Process` gombra.

Condition Type

PL/SQL Function Body Returning a Boolean

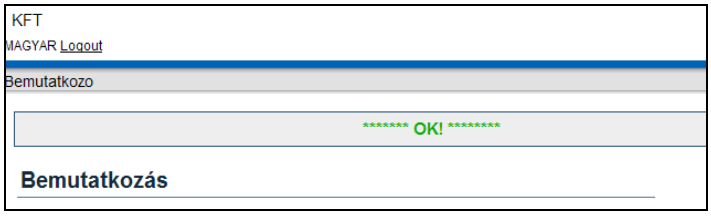
[PL/SQL] [item / column=value] [item / column not null] [item / colum

Expression 1

```
declare
  TablaSzam   number(1);

begin
  select count(*)
    into TablaSzam
    from user_tables
    where (table_name='SAS') or (table_name='PAL');

  if TablaSzam = 2
    then return false;
    else return true;
  end if;
end;
```

11.8. ábra A táblalétrehozó folymat feltétele

- Ezután futtassuk le az alkalmazást (11.9. ábra) Láthatjuk, hogy a táblákat létrehozta. Többször nem kell létrehozni.



11.9. ábra. *Sikeres tábla-létrehozás*

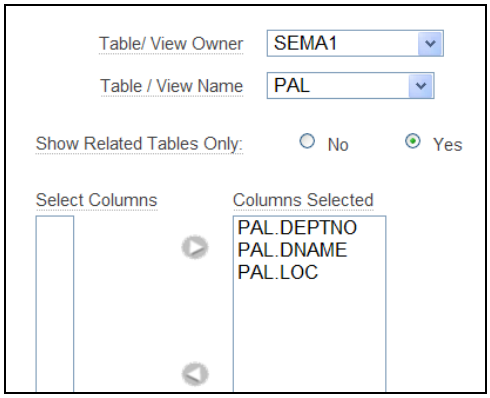
4. A `Pal` tábla megjelenítése

Most ismertetjük a `Pal` tábla `Bemutatkozo` lapon történő megjelenítését. A tábla tartalma helyezkedjen el a cégismertető szöveg mellett. E tábla lesz az úgynevezett "lefúrás" (`Drill down`) eszközének fő táblája.

- Kattintsunk a `Bemutatkozo` lapra, vagy `EditPage 1`,

11.10. ábra *Az alkalmazás lapjai*

- Kattintsunk a Tartomány szekcióban a (`+`) gombra
- Válasszuk a (`Report`) tartományt, `Next`
- A létrehozáshoz használjunk varázslót (`Wizard Report`), `Next`
- Adjuk meg a fejléc nevét: `PAL tabla`, valamint a Megjelenítési (`Display`) Pontot. Legyen ez a Bemutatkozó szöveg mellett.
  Ezért válasszuk a `Page Template Body 1, Next`
- A megjelenő lapon válasszuk ki a `Pal` táblát (`Table/View Name`) , és az elkészítendő listában megjelenő oszlopainak neveit oly módon, hogy az `Select Columns` oszlopban jelöljük ki az összes oszlopnevet, és a jobbra mutató nyilra kattintva vigyük át azokat a `Columns Selected` oszlopba (11.11. ábra), `Next`

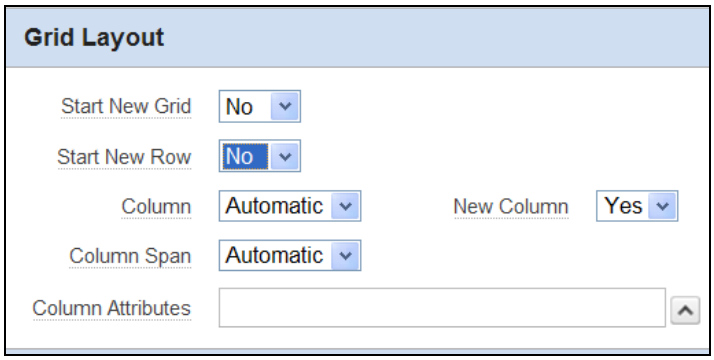


11.11. ábra. *A tábla-kijelölés*

- Riport jellemzőit, a lista megjelenítés paramétereit állíthaztjuk be. Ezeket változatlanul hagyhatjuk, de beállíthatunk természetesen más kinézetű listát is (11.12. ábra), `Next`

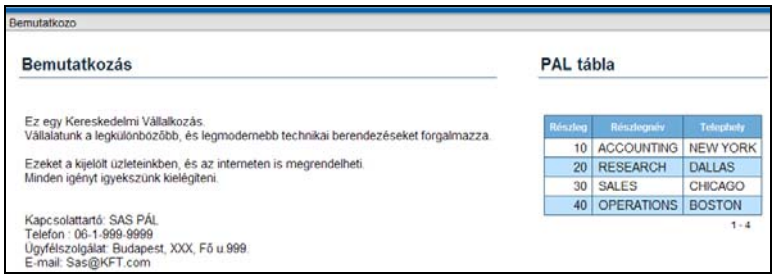
11.12. ábra *Riport jellemzői*

- Hozzuk létre a listát (`Create Region`)
- Mielőtt lefuttatnánk, állítsuk be a Pal tabla riportját, úgy hogy a bemutatkozó szöveg **mellett** legyen.
  - Tartomány `PAL tábla`, majd állítsuk át a `Grid Layout` ablakban a rácszsrekezetet (11.13. ábra) (Ne legyen új Rács, Ne legyen új sor, hanem új oszlop legyen)



11.13. ábra *A megjelenítés az előző elem MELLETT*

- `Applay Changes`,
- Kattintsunk a Tartomány szekcióban (`Regions`) a `Report` linkre. A `Heading Type` legyen `Custom`, és a megjelenő fejléceket változtassuk meg magyarra, Megint `Applay Chages`.
- Futtassuk le az alkalmazás módosított első lapját (`Run Page`). Láthatóan megjelenik a `Pal` tábla.



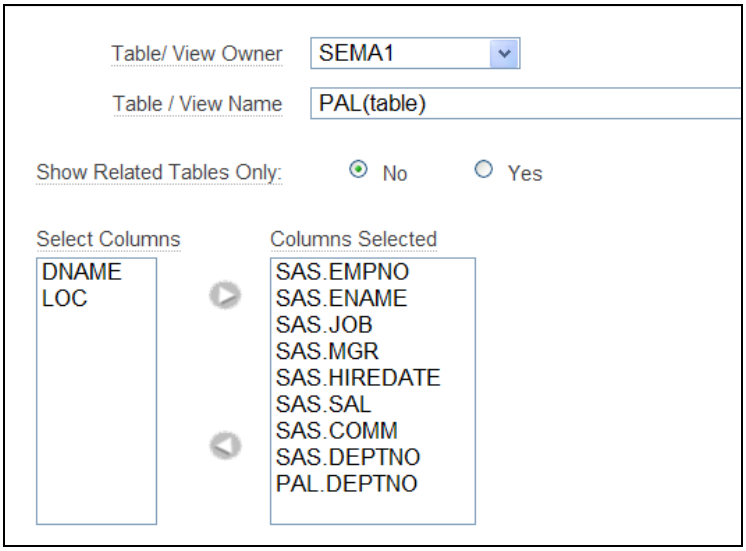
11.14. ábra. *A Bemutatkozó lap futási képe*

5. A részletező tábla megjelenítése

Az alábbiakban a `Drill Down` eszköz részletező tábláját fogjuk megjeleníteni az alkalmazás második (`Dolgozo` nevű) lapján.

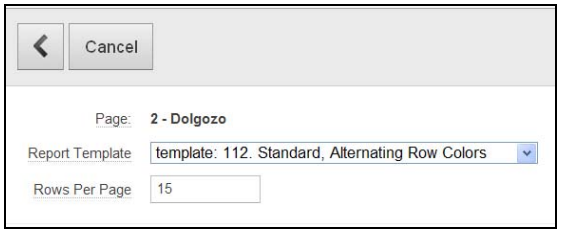
- Menjünk vissza a Fejlesztői Eszköztáron az `Edit Page 2` menüpontra
- Kattintsunk a `Tartomány` szekcióban a `(+)` jelre gombra
- Legyen a tartomány típusa (`Report`), `Next`
- A létrehozáshoz használjunk varázslót (`Wizard Report`), `Next`
- Adjuk meg a fejlécet `Sas Tabla részlete`, a `Region Template : Reports Region: Alternativ 1`, Belépési pontja maradhat a felajkánlott (`Body 3`), `Next`
- A megjelenő ablakban válasszuk ki a `Sas` táblát és jelöljük ki minden oszlopát, majd váltsunk rádiógombot (`Show Related Tables Only`): Legyen `No`. Ekkor meg tud-

juk jeleníteni a `Pal` táblát is. Tegyük át a jobb oldalra a nyíl segítségével a `Pal` tábla `deptno` oszlopát (11.15. ábra), `Next`



11.15. *A részlezező lap varázslója*

- A megjelenő ablakban látjuk az összekapcsolási feltételt, `Next`
- Megváltoztathatjuk a riport kinézetét, és a megjelenő sorok számát. (11.16. ábra), `Next`



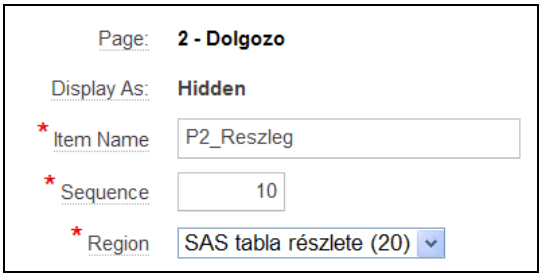
11.16. ábra *Jellemzők*

- `Create Region`
- Futtassuk le az alkalmazás módosított második lapját (`Run Page`).
- Írjuk át az oszlopneveket magyarra (Fejlesztői Eszköztár, `Edit Page 2`, `Page Rendering`, `Regions`, `Sas tábla részlete`, `Report attributes`, `Heading` oszlop), és a `Deptno` oszlop első előfordulásánál a `Show` oszlopban töröljük a megjelenítést. Az `Empno` oszlopnévtől kezdve a magyar oszlopnevek legyenek a következők: `Azonosító`, `Név`, `Munkakör`, `FőnökAzon`, `Dátum`, `Fizetés`, `Jutalék`, `RészlegAzonosító`.
- Ezután `Apply Changes`.
- Futtassuk le ismét az alkalmazás módosított második lapját. Ekkor a teljes `Sas` tábla látható, de a későbbiekben már a kijelölt részleghez tartozó dolgozókra vonatkozó részlet fog csak megjelenni.

6. Rejtett elem hozzáadása

Az alábbiakban létrehozandó rejtett elem segítségével fogjuk összekapcsolni a `Pal` és a `Sas` táblát.

- Fejlesztői Eszköztár, `Edit Page 2`. Ezután az Elemek (`Items`) szekcióban kattintsunk a Létrehozás (+) jelre, és válasszuk a rejtett (`Hidden`) elemet, `Next`.
- Töltsük ki a mezőket. A rejtett elem neve legyen `P2_Reszleg`, és a helye legyen a `Sas` tábla részletéhez tartozó tartomány (11.17. ábra), `Next`, `Next`,

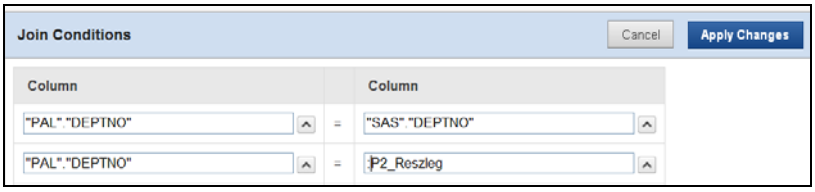


11.17. ábra. *A rejtett elem jellemzői*

- `Create Item`.

7. Feltétel hozzáadása

- Kattintsunk a tartomány szekcióban (`Page Rendering`, `Regions`) a `Sas Tábla részlete` hiperhivatkozásra, és válasszuk a Lekérdezés szerkesztése (`Query Definition`) fület.
- Válasszuk a `Modify Join Condition` menüpontot. Itt látható a fejlesztő környezet által automatikusan létrehozott idegen kulcs kapcsolat a `Pal` és a `Sas` tábla között. Írjunk be egy új feltételt, amely szerint a `Pal` táblabeli részlegazonosítót összekapcsoljuk a rejtett elemmel (`:P2_Reszleg`), és érvényesítsük a változásokat (`Apply Changes`, 11.18. ábra). Ez a kapcsolat teszi majd lehetővé a `Pal` tábla részlegazonosítójára való kattintás hatására a `Sas` tábla e részlegbeli dolgozóinak megjelenítését, vagyis a "lefúrást".



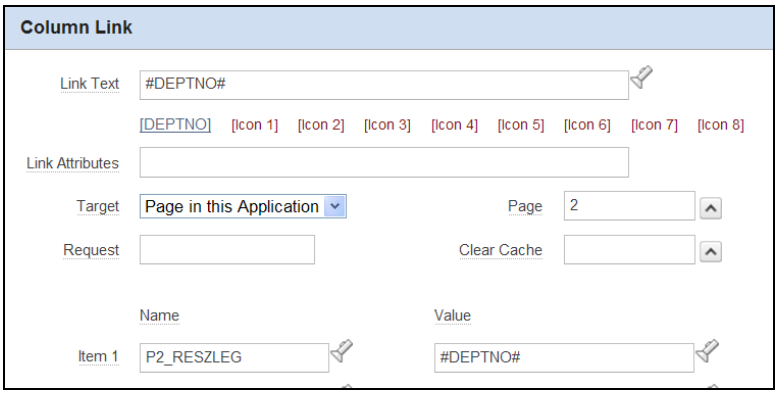
11.18.ábra. *Az összekapcsolási feltételek*

8. Összekapcsolás

Kapcsoljuk össze a `Bemutatkozo` és a `Dolgozo` lapon lévő lekérdezéseket.

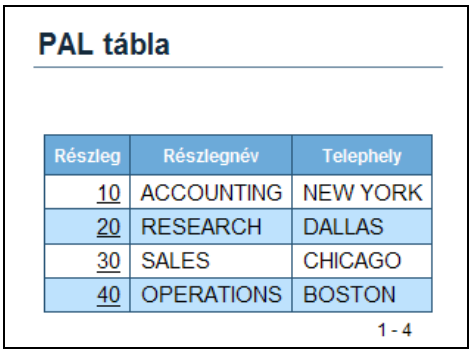
- Menjünk vissza az első (`Bemutatkozo`) lapra (például a lapszerkesztő < gombja segítségével), és kattintsunk a `Pal tábla` hiperlinkre a tartomány szekcióban

- Válasszuk a `Report Attributes` fület, és szerkesszük a `DEPTNO` oszlopot (a szerkesztő ikonra való kattintással)
- Menjük lefelé a az oszlop kapcsolás (`Column Link`) ablakig
- Válasszuk ki az elemlámpa segítségével az összekapcsoló oszlopot, majd adjuk meg, hogy melyik laphoz kapcsolódik (`Page:2`) a rejtett elemen keresztül a 11.19. ábrán látható módon.



11.19. ábra. *Az oszlopok összekapcsolása*

- Menjük fel a lapon, és érvényesítsük a változtatásokat (`Apply Changes`)
- Futtassuk az alkalmazásunkat (11.20.ábra)



11.20. ábra. *Az alkalmazás futásának riportja*

Az összekapcsolás révén a részlegazonosító hiperlink lett. Ha rákattintunk a részlegazonosító oszlop valamelyik elemére (például a `30`-as részlegre), akkor az ahhoz tartozó részleg dolgozóit mutatja a `Dolgozo` lapon (11.21.ábra).

SAS tabla részlete

Azonosító	Név	Munkakör	FőnökAzon	Dátum	Fizetés	Jutalék	Részleg Azonosító
7698	BLAKE	MANAGER	7839	1981-05-01	2850		30
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7900	JAMES	CLERK	7698	1981-12-03	950		30

1 - 6

11.21. *A részletező második lap*

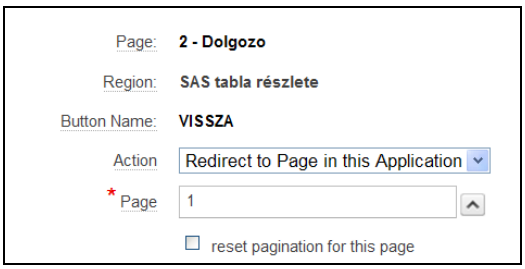
Ezzel az első két lapunk már működik.
Még egy `Vissza` gombot célszerű a lapra helyezni, hogy egyszerűen visszatérhessen a felhasználó az első lapra. Ennek létrehozása az alábbi módon történhet:

- Fejlesztői eszköztár, `Edit Page 2`, Gomb (`Button`) szekció Létrehozás (+)
- Válasszuk ki a gomb helyének a `Sas tábla részlete` tartományt, `Next`
- Válasszuk a `Create a button in a region position` gombot, `Next`
- A gomb neve és a felirata legyen: `Vissza`, 11.22.ábra), `Next`

Page: **2 - Dolgozo**
Region: **SAS tabla részlete**
* Button Name: VISSZA
[Cancel] [Next] [Previous] [Apply] [Submit] [Delete] [Fin
* Label: Vissza
Button Style: Template Based Button
Button Template: Button, Alternative 2
Button Type: Normal
Button Attributes:

11.22.ábra. *A Vissza gomb adatai*

- Maradhat a tartomány alján a gomb (`Botton of Region`), `Next`
- Az első lapra (`Page:1`) kell visszaugrania (11.23. ábra).
- Feltételt nem adunk meg így `Create Button`

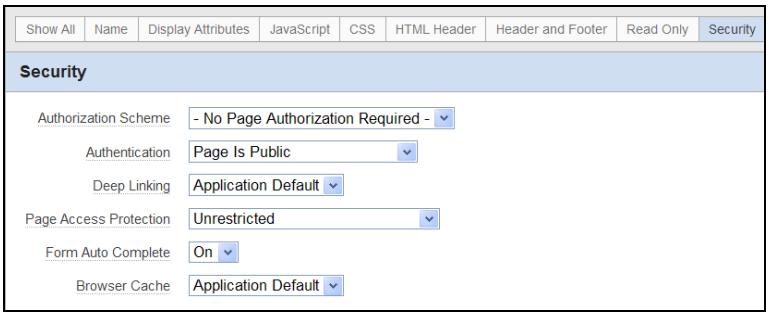


11.23.ábra. *A Vissza gomb az első oldalra ugrik*

9. A `Bemutatkozo` lap nyilvánossá tétele

Ezután állítsuk be a megfelelő jogosultságokat, hitelesítéseket, engedélyezéseket.
Mivel a Bemutatkozó lapot minden felhasználó számára elérhetővé szeretnénk tenni, ezért a `magyar` felhasználó kizárólagosságát meg kell szüntetnünk. Ennek megvalósítását mutatjuk be az alábbiakban.
(Megjegyezzük, hogy az ehhez tartozó biztonsági beállítások utólag is megváltoztathatóak. (A `Shared` komponens, `Security` rész, `Authentication Schemes` menüpontban három biztonsági beállítás van alapértelmezésben. Hozhatunk létre újat is, de a `Change Current` menüponttal is megváltoztathatjuk az alkalmazásunk biztonsági beállításait.)

- Első (`Bemutatkozo`) lap szerkesztése (`Edit Page 1`)
- A Lap (`Page`) szekcióban a Szerkesztés (`Edit`) ikonra kattintsunk, majd válasszuk a Biztonsági Beállítás (`Security`) menüpontot, és itt állítsuk be (`Authentication`) a `Page Is Public` lehetőséget (11.24. ábra), majd `Applay Changes`



11.24. ábra. *A lap publikus lesz*

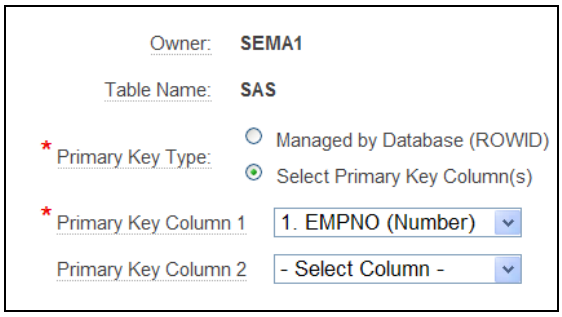
- Ugyanezt végezzük el a `Dolgozo` lapon is

A `Karbantarto` lap elkészítése

Először hozzuk létre a `Karbantarto` lapot. Vegyünk fel az alkalmazásunkba egy harmadik, `Tabular Form` típusú lapot. Ennek módját az alábbiakban mutatjuk be.

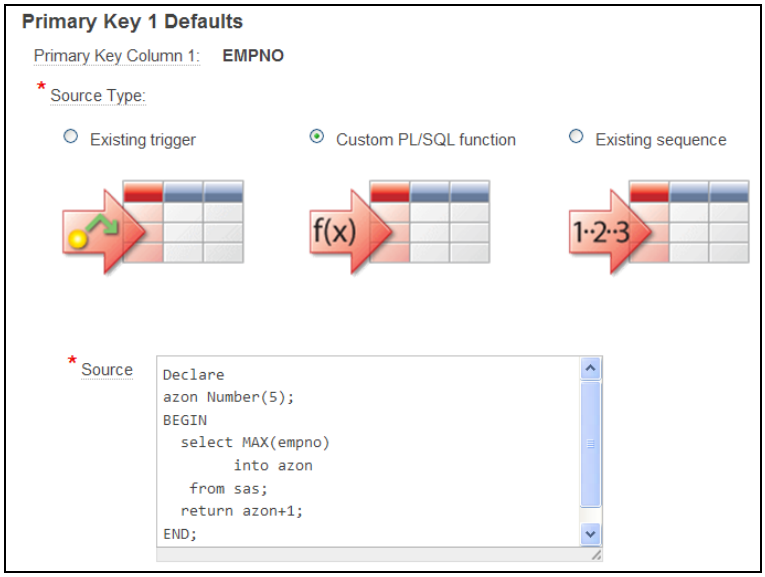
- A Fejlesztői eszköztárban válasszuk ki az `Create` menüpontot, válasszuk a `New Page` rádiógombot, `Next`

- Jelöljük ki a `Form` laptípust, `Next`
- Jelöljük ki a `Tabular Form` rádiógombot, `Next`
- A tábla tulajdonosa változatlanul `sema1`,
- Táblának válasszuk ki a `Sas` táblát, `Next`
- Jelenítsük meg minden oszlopát (kijelöléssel, ez az alapértelmezés), `Next`
- Az adatbázis-művelet legyen a lehetséges összes (`Update`, `Insert and Delete`), `Next`
- Válasszuk a `Primary Key Type` rádiógombot. (11.25. ábra). Az `empno` maradjon az elsődleges kulcs, `Next`



11.25. ábra. *A lap publikus lesz*

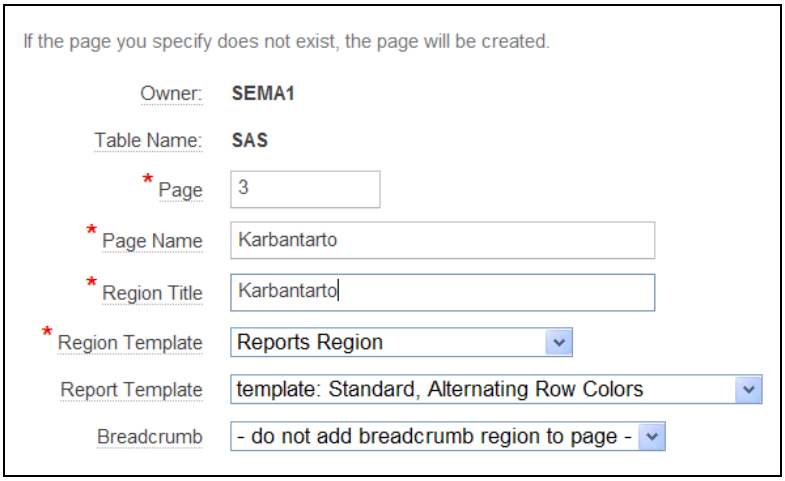
- A forrás kijelölésére válasszuk a PL/SQL függvényt (`Custom PL/SQL function`), és ugyanitt írjuk is meg magát az eljárást is, mely az elsődleges kulcsot a legnagyobbtól egyesével növeli (11.26. ábra), majd `Next`



11.26. ábra. *Az azonosító beállítása*

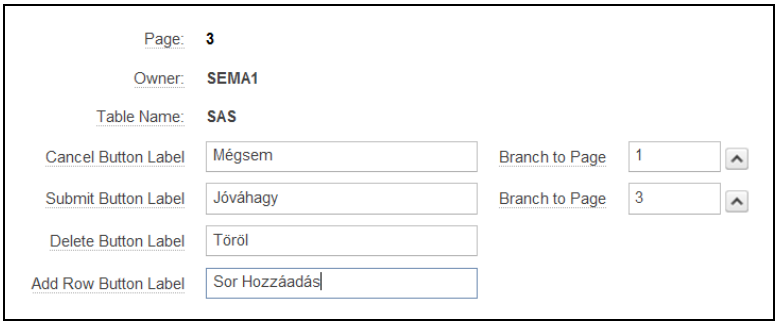
Ügyeljünk arra, hogy a függvény megadásában sehol ne szerepeljen üres sor (utolsó sorként SE)!
Megjegyezzük, hogy a beírt függvény utólagos módosítására is van lehetőség, mégpedig a következő módon: `Page Rendering`, `Regions`, `Karbantarto`, `Report Attributes`, az `empno` oszlop szerkesztése az ikonnal a `Tabular Form Attributes` ablakban, a `Primary Key Source` szövegmezőben.

- A megjelenített oszlopok mindegyike legyen engedélyezett változtatásra (kijelöléssel, ez az alapértelmezés), `Next`
- Az oldal legyen a harmadik (`3`), a lap neve `Karbantarto`, és a megjelenítésben az egymást követő sorok legyenek váltakozó színűek (…`Alternating Row Colors`, 11.27. ábra), `Next`



11.27.ábra. *A harmadik oldal adatai*

- A létrehozandó lapon ne legyenek fülek, `Next`
- A megjelenő ablakban írjuk át a gombok feliratait magyarra (11.28. ábra), Fogadjuk el, illetve írjuk be melyik lap jelenjen meg a gomb lenyomása után, `Next`



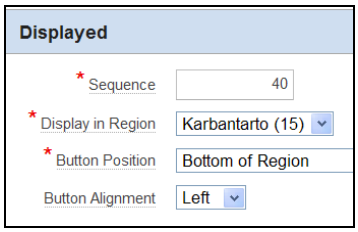
11.28. ábra. *A gombok magyar feliratai*

- Ellenőrizzuk, majd `Create`,

- Run. Futtassuk le a lapot és ellenőrizzük a funkciókat

Tegyük láthatóvá a Sor Hozzáadása gombot, mert nagyon a lap szélén jobb oldalon van.

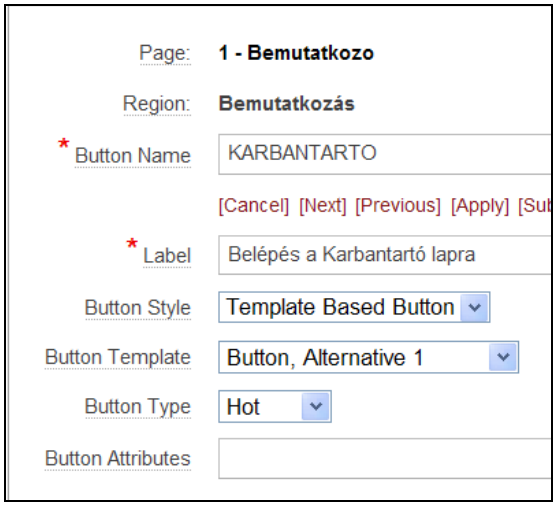
- Edit Page 3, a Buttons szekcióban kattintsunk Sor Hozzáadás gombra, például igazítsuk balra



11.29. ábra. *A gombok balra igazítása*

A Mégsem gomb az eléső lapra ugrik vissza (lásd 11.28. ábra), de onnan nem tudunk még a Karbantarto lapra jutni. Ennek érdekében vagy Füleket, vagy gombot kell helyeznünk az első lapra. Most tegyünk fel egy gombot, melynek segítségével jutunk a 3. Karbantarto lapra

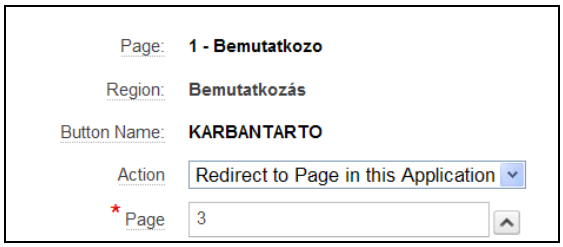
- Edit Page 1
- Kattintsunk a Gomb (Button) szekcióban a Létrehozás (+) ikonra, Next
- Válasszuk a Bemutatkozás tartományz a gomb helyéül, Next
- Create Button in region Position radiógombot jelöljük ki, Next
- Írjuk meg a gomb tulajdonságait (11.30. ábra), Next



11.30. ábra. *A gombok balra igazítása*

- Legyen a tartomány alján, Next

- Állítsuk be az akciót, és a megjelenítendő lapot, azaz melyik lap jelenjen meg közvetlenül a kattintásra.(11.31. ábra)



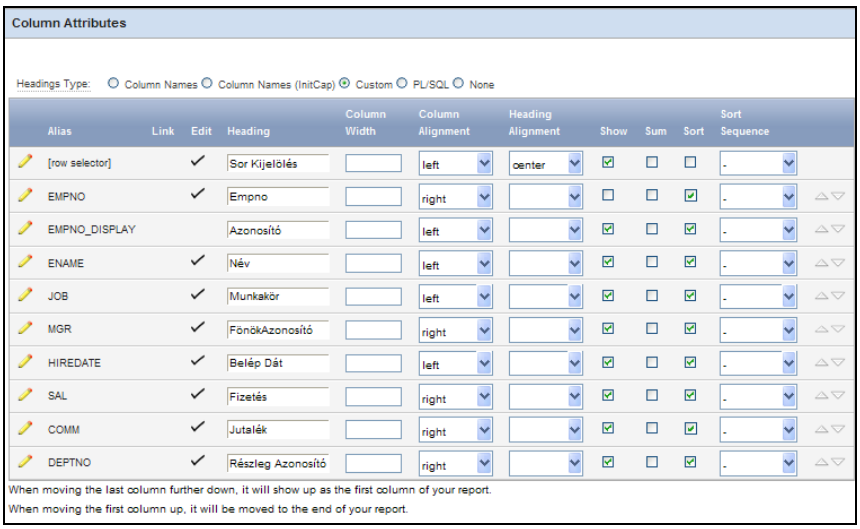
11.31. ábra. *A gombok balra igazítása*

- `Create Button`

A `Karbantarto` lap külalakjának utólagos megváltoztatása

Végezzük el a `Karbantarto` lapon az oszlopnevek magyarítását, és hozzunk létre kiválasztó elemeket (legördülő menüket) a munkahely, a főnöknév, a telephely adatoknál.
Az első lépés tehát az oszlopnevek magyarítása.

- `Edit Page 1`, majd válasszuk ki a 3. oldalt (`Page 3`), ezután `Go`, és kattintsunk a tartományon belül a `Karbantarto` riportra, itt a `Report Attributes` fülre, és a `Heading` oszlopban írjuk át a `Sas` tábla oszlopneveit magyarra (11.32.ábra).
  Ügyeljünk arra, hogy az `EMPNO` oszlophoz tartozó `empno` fejléc beállítást NE módosítsuk. Ez utóbbi (tehát az `EMPNO` értékek) fejlécét az `EMPNO_DISPLAY` (kizárólag megjelenítésre szolgáló virtuális) oszlopnál adjuk meg. (Az `EMPNO` oszlophoz esetlegesen megadott fejlécnév egyébként meg se jelenik).
  Megjegyezzük, hogy a `Heading` oszlopba olyan oszlopneveket írjunk be, melyeket a megjelenített lista fejlécében látni szeretnénk. Arról, hogy a megjelenített lista egyes fejléceinek oszlopaiban valóban helyes adatok szerepeljenek, a kiválasztó listákhoz tartozó szelekciós utasítások gondoskodnak (lásd alább).



11.32. ábra. *Az oszlopnevek magyarítása*

A következő lépésben állítjuk be a kiválasztó elemeket. A kiválasztó elemek (a dátum kivételével) az engedélyezett mezőértékek egyszerű felsorolt listáját tartalmazzák. Megjegyezzük, hogy célszerű minden oszlopbeállítás után elvégezni annak tesztelését, vagyis az alábbi pontok mindegyike után egy (ismételt) `Apply Changes` érvényesítést elvégezni, majd a próbafuttatást.

- Jelöljük ki szerkesztésre először a `JOB – Munkakör` oszlopot az ikonra való kattintással, majd lépjünk a `Tabular Form Element` ablakba, és a `Display as` legördülő menüből válasszuk a `Select List (query based LOV)` elemet. Ezután a `List of Values` ablakban a `Display Value` mező legyen `No`, mivel a munkakör mező nem lehet `NULL` értékű, a `Display Extra Value` mezőt szintén `No` értékre állítsuk (mivel nem akarjuk az aktuális mező tartalmával kiegészíteni a választékot), majd a `LOV QUERY` szövegdobozba írjuk be a munkakör elemeit előállító alábbi lekérdezést, végül `Apply Changes`.

```
SELECT DISTINCT job a, job b
  from sas
  order by a
```

- Jelöljük ki szerkesztésre az `MGR – Főnöknév` oszlopot, a `Tabular Form Element` ablakban végezzük el a fenti beállítást, majd a `List of Values` ablakban a `Display Value` mező legyen `Yes`, mivel megengedjük, hogy egy dolgozónak ne legyen főnöke (és így az `MGR – Főnöknév` nevű oszlop értéke valamely sorban `NULL` érték is lehessen), a `Display Extra Value` mező pedig legyen `No`, mivel nem akarjuk az aktuális mező tartalmával kiegészíteni a választékot. Ezután írjuk be a `LOV QUERY` szövegdobozba a főnöknév elemeit előállító alábbi lekérdezést (melyben tehát főnökként csak már létező főnök állítható be, vagy a `NULL` érték), végül `Apply Changes`.

```
SELECT ename, empno
  from sas
  where empno IN (SELECT DISTINCT mgr
```

```
                    from sas)
```

- Jelöljük ki szerkesztésre a DEPTNO – Telephely oszlopot, a Tabular Form Element ablakban végezzük el az JOB – Munkakör oszlopnál bemutatott beállításokat. Tehát az ott bemutatott módon állítsuk be a Tabular Form Element ablakot, a List of Values ablakban a Display Value, és a Display Extra Value mező legyen egyaránt No értékű, a LOV QUERY szövegdobozba pedig írjuk be a telephely elemeit előállító alábbi lekérdezést, majd Apply Changes

```
SELECT loc, deptno
  from pal
  order by loc
```

- A fizetés és a jutalék oszlopokat vegyük a megjelenített listán keskenyebbre. Ennek érdekében a jelöljük ki szerkesztésre először a SAL – Fizetés oszlopot, és a Tabular Form Element ablakban az Element width mező értékét állítsuk kisebbre (6), majd Apply Changes. Ugyanezt végezzük el a COMM – Jutalék oszlopban is.

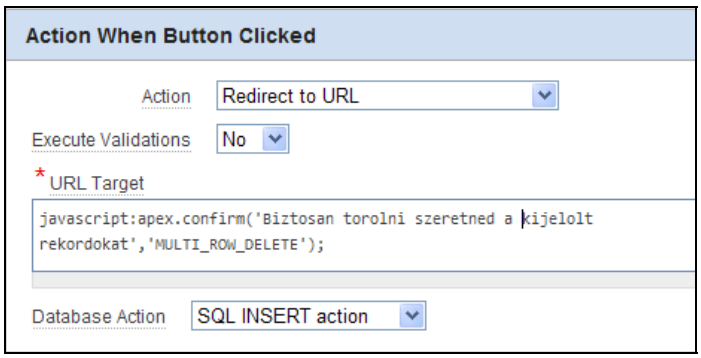
Végezetül a törlő gomb által megjelenő Windows ablak kérdését írjuk át magyarrá.

Megoldás

Létezik egy javasScript deleteConfirm(msg) függvény a Oracle rendszerben, ha ezt meghívjuk és az üzenet magyar nyelvű lesz, akkor megoldottuk a feladatot.
Megjegyezzük: az Oracle rendszerben igenn sok már megírt java program található. Ezeket illetve általunk írt bármilyen javaScript függvényt bárhová beilleszthetünk
A törlő javaszkript példáját most itt bemutatjuk:

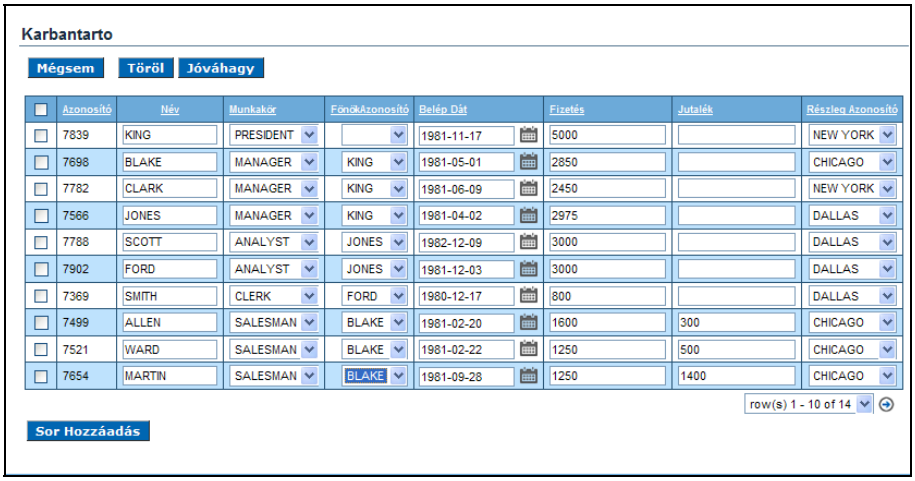
```
<script type="text/javascript">
  function deleteConfirm(msg)
  {
var confDel = msg;
if (confDel == null)
  confDel = confirm("Would you like to perform this delete action?");
else
  confDel = confirm(msg);
if (confDel == TRUE)
    doSubmit('Delete'
  }
</script>
```

- Edit Page 3,
- Majd a Gomb (Button) ablakban kattintsunk Töröl hiperhivatkozásra, menjünk lefelé az Optional URL Redirect ablakba, azon belül az URL Target szövegdobozba, és itt írjuk át a zárójelben levő üzenetet magyar nyelvűre (11.33. ábra).



11.33. ábra. *Törlő hibaüzenet magyarítása*

- A beállításokkal elkészültünk, egy újabb `Apply Changes` után futtassuk le a `Karbantarto` lapot (11.34.ábra):

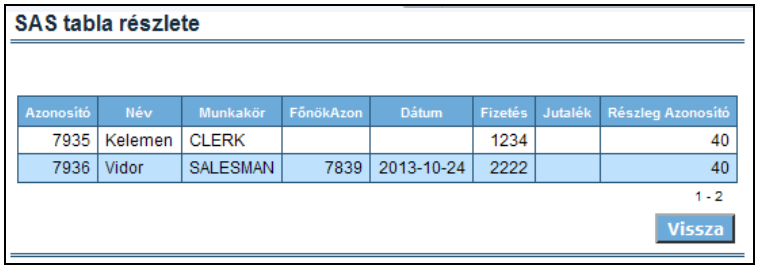


Azonosító	Név	Munkakör	FőnökAzonosító	Belép Dát	Fizetés	Jutalék	Részleg Azonosító
7839	KING	PRESIDENT		1981-11-17	5000		NEW YORK
7698	BLAKE	MANAGER	KING	1981-05-01	2850		CHICAGO
7782	CLARK	MANAGER	KING	1981-06-09	2450		NEW YORK
7566	JONES	MANAGER	KING	1981-04-02	2975		DALLAS
7788	SCOTT	ANALYST	JONES	1982-12-09	3000		DALLAS
7902	FORD	ANALYST	JONES	1981-12-03	3000		DALLAS
7369	SMITH	CLERK	FORD	1980-12-17	800		DALLAS
7499	ALLEN	SALESMAN	BLAKE	1981-02-20	1600	300	CHICAGO
7521	WARD	SALESMAN	BLAKE	1981-02-22	1250	500	CHICAGO
7654	MARTIN	SALESMAN	BLAKE	1981-09-28	1250	1400	CHICAGO

11.34. ábra. *Törlő hibaüzenet magyarítása*

- Próbáljuk meg az adatbevitelt, módosítást, végül a törlést is.

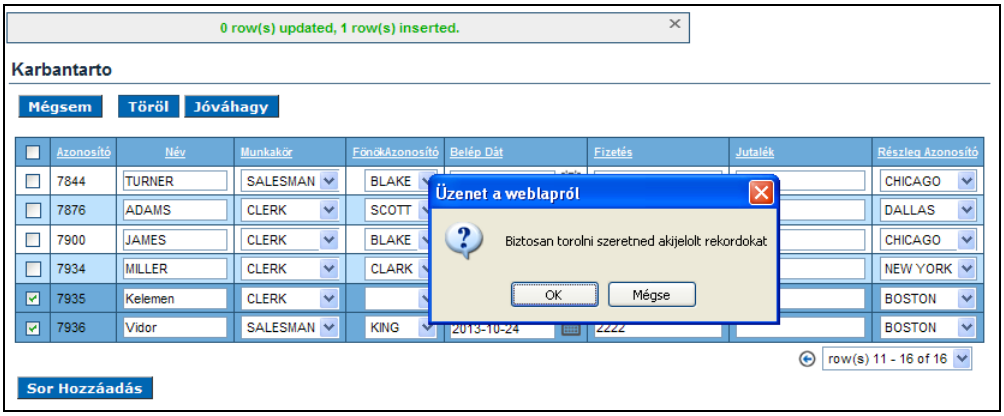
Figyeljük meg, hogy felvettük a `Sor Hozzáadás` gomb,majd a `Jóváhagyás` gomb segítségével `Kelemen`, és `Vidor` nevű alkalmazottakat. Kelemennek nincs főnöke, és Bostonban fog dolgozni, Vidor alkalmazottnak van főnöke és szintén Bostoni. Mielőtt törölnénk a kijelölt alkalmazottakat, menjünk vissza az első lapra,és kattintsunk a 40-es részlegre. Láthatjuk, hogy a felvett dolgozók megmaradtak akkor is, amikor az első lap betöltődött.



Azonosító	Név	Munkakör	FőnökAzon	Dátum	Fizetés	Jutalék	Részleg Azonosító
7935	Kelemen	CLERK			1234		40
7936	Vidor	SALESMAN	7839	2013-10-24	2222		40

11.35. ábra *Bostoni alkalmazottak*

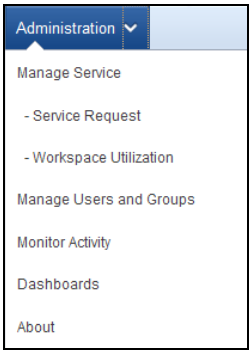
- Töröljük ki, `Kelemen` és `Vidor` alkalmazottakat



11.36. ábra. Adatok sikeres bevitele, és a weblap magyar üzenete

Felhasználói csoport, Felhasználó létrehozása

- Kattintsunk a Fejlesztői Eszközsoron az Application nnn menüpontra
- 1.) Egyik megoldás :Kattintsunk felül az Administration Fülre, jobb oldali Tevékenységek (`Tasks`) ablakban válasszuk a `Create Users` menüpontot.
  2.) Másik megoldás: Kattintsunk a Administration fül nyilára , és a megjelenő ablakban kattintsunk a Manage Users and Groups (.



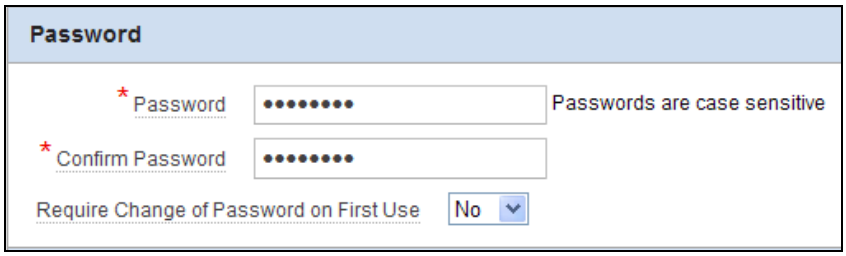
11.37. ábra. *Az elvégezhető feladatok*

- Hozzuk létre a `kezelo` csoportot, és a `karbantarto` csoportot.
- Válasszuk a Csoport (`Group`) Fület
- Kattintsunk a felhasználói Csoport Létrehozása (`Create User Group`) gombra
- Írjuk be a Csoport nevét , `Create Group` gomb, és létrejött a két csoport (11.38. ábra)

11.38. ábra. *A kezelő felhasználói csoport*

Hozzunk létre felhasználókat is és rendeljük a csoportokhoz.
Legyen `Varga` karbantartó és `Balla` kezelő.A létrehozás lépései.

- Lépjünk be az Alkalmazás Varázslóba. Legfelül az Adminisztrációs fül lnyilára kattintva a megjelenő ablakban válasszuk a `Manage Users and Groups` menüpontot.
- Válasszuk a `Users` Fület és kattintsunk a `Create User` gombra.
- `User Name` : `Varga`; adjuk meg az e-mail cimét,
- Haladjunk lejjebb, válasszuk Vargát fejlesztőnek (`User is a developer`), kattintsunk a `Yes` gombra
- Adjuk meg a jelszót legyen `Tigris-2`, Ne legyen szükséges a jelszó megváltoztatása az első belépéskor Legyen `No` (11.39.ábra)

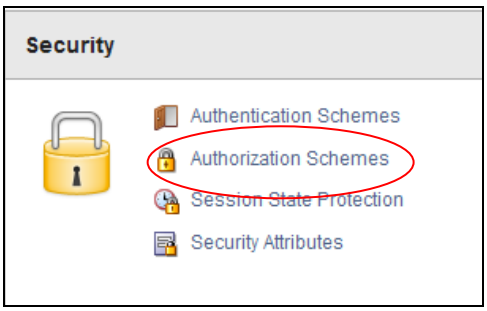


11.39. ábra. *A jelszó ablak*

- Menjünk le és jelöljük ki a karbantaro csoport tagjaként.
- Menjünk fel és kattintsunk a `Create and Create Another` gomb
- Ugyanígy hozzuk létre `Balla /Tigris-3` jelszóval csak ő kezelő legyen.

A biztonság beállítása

Menjünk az alkalmazásunkra (`Edit Application`) válasszuk a `Shared Components` ikont. Lépjünk bele és a Biztonsági osztály (`Security`) részben válasszuk az Engedélyezési Minta (`Authorization Schemes`) menüpontot (11.40.ábra):



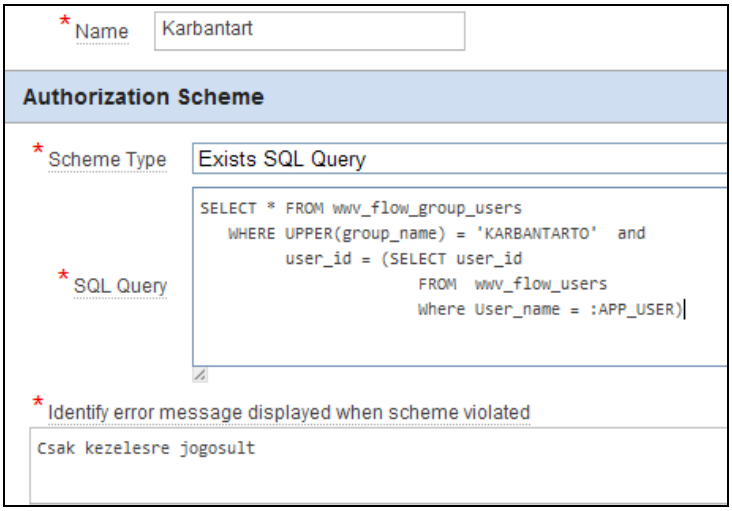
11.40. ábra. *Biztonsági osztályok*

Megjegyzés

A csoport_név lekérdezhető a `SELECT * FROM wwv_flow_group_users` táblából az APEX SQL Commands objektumában, például
A felhasználók lekérdezhetők: a `www_flow_users` táblából

Tehát

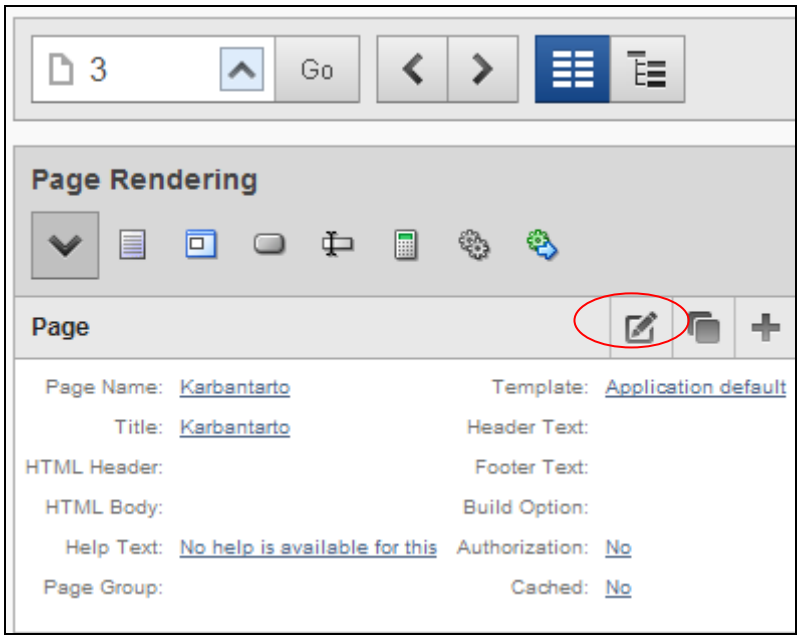
- Hozzunk létre egy újat. `Create`,
- `From Scratch` ,`Next`,
- A Minta típusa legy `Exists SQL QUERY`
  A fenti két adatbázis-táblából lekérdezhető a belépő felhasználó létezése és csoportja. Irjuk meg a lekérdezést (a táblákban a harmadik "v" az szimpla!) (11.41. ábra). Ha létezik a `Karbantarto` felhasználói csoport és a felhasználó ennek csoportnak a tagja, akkor beléphet.



11.41. ábra. *Az új Karbantarto engedélyezési minta*

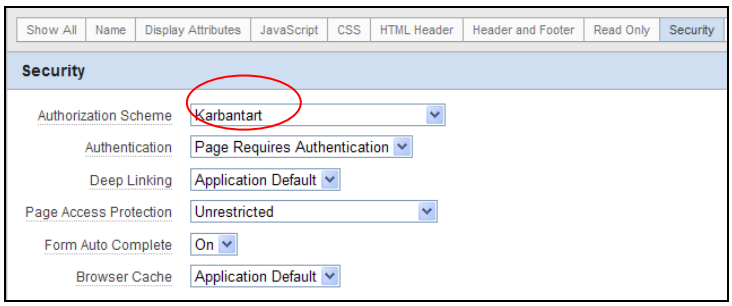
- `Create Athorization Scheme`

- Most menjünk a harmadik oldalra `Edit Page 3`, és kattintsunk az lap jellemzőinek szerkesztése ikonra (`Edit`)11.42. ábra).



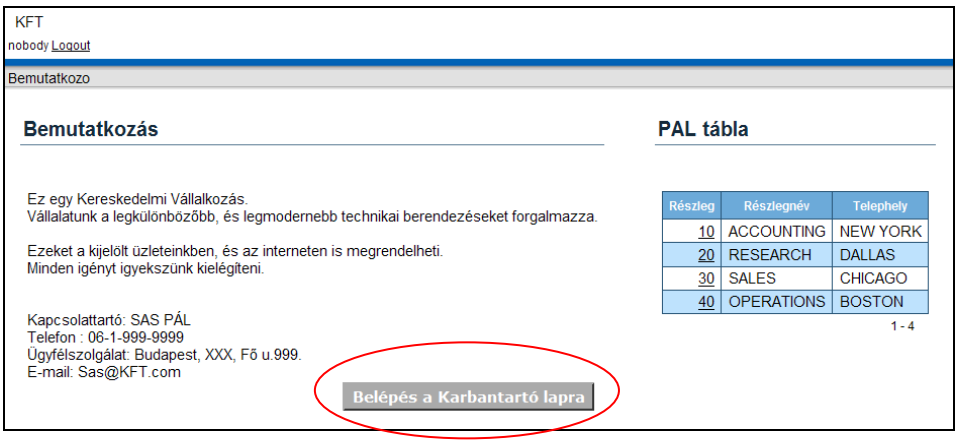
11.42. ábra. *A Lap Szerkesztése*

- Kattintsunk az Biztonság(`Security`) menüpontra, illetve haladjunk lefelé a lapon az biztonság (`Security`) részig, és állítsuk be a `Karbantart` titkosítást (11.43. ábra).



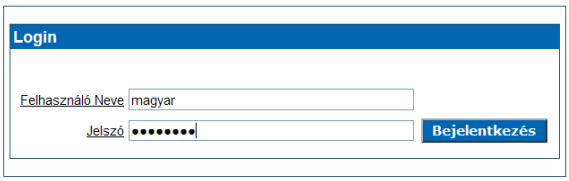
11.43. ábra. *Engedélyezési minta kiválasztása a gördülő listából*

- Menjünk fel a lap tetejére, és érvényesítsük a változtatást. (`Apply Changes`)
- Lépjünk ki és futtassuk le az alkalmazást.



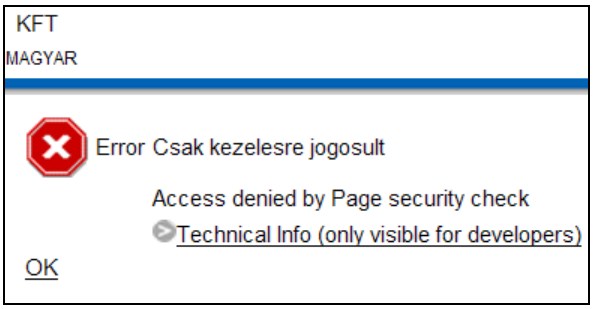
11.44. ábra. *A bejelentkező első pulikus lap lap*

Kattintsunka `Belépés a Karbantartó lapra` gombra:
Megjelenik a jelszókérő bejelentkező lap, amit mi hoztunk már létre (11.45.ábra).



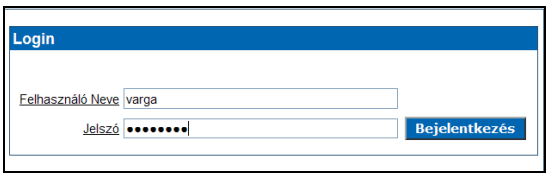
11.45 ábra. *Az általunk létrehozott bejelentkező lap*

Nézzük meg a sszokásos bejelentkezéssel: `magyar/Tigris-1` jelszópárossal. Tudjuk, hogy magyar az adminisztrátor.
Láthatjuk, hogy ez a felhasználó nem tud belépni a karbantartásra (11.46.ábra).



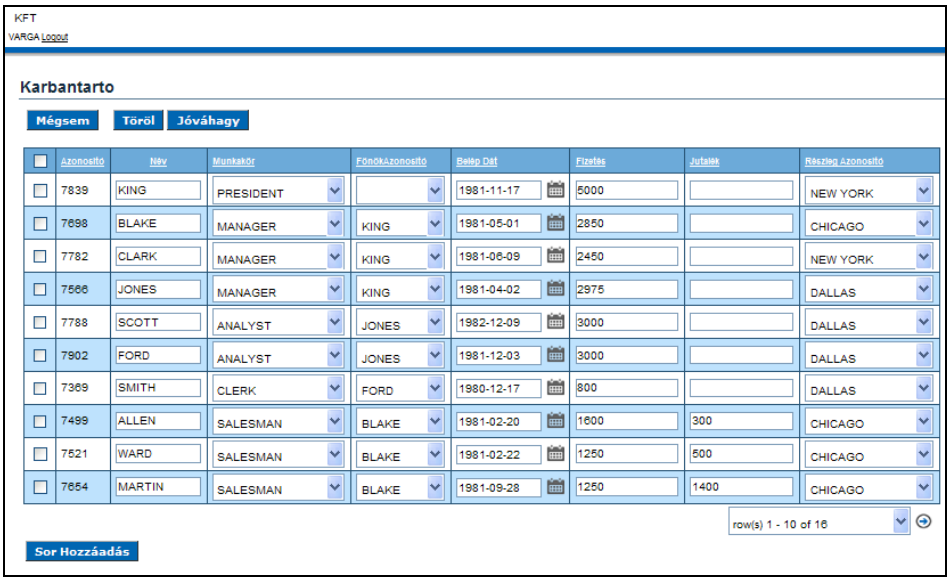
11.46. ábra. *Magyar nem tud belépni a karbantartó lapra*

Kattintsunk az `Ok` gombra, és adjuk meg egy karbantartó belépését (`Varga/Tigris-2`). (11.47. ábra), aki a `Bejelentkezés` gomb lenyomására el tudja indítani a `Karbantartó` lapot.



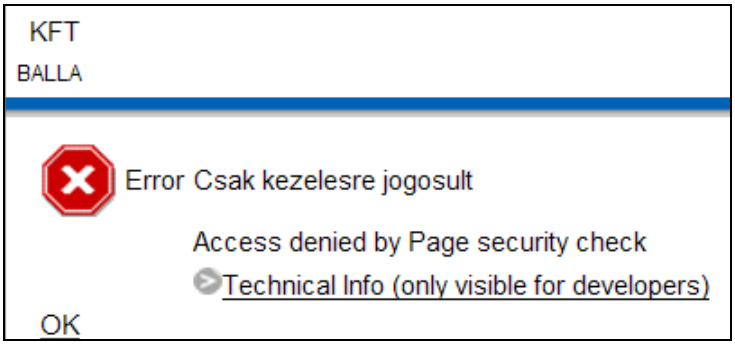
11.47. ábra. *A karbantaro csoport tagja lép be*

Belépés sikerült (11.48. ábra).



11. 48. ábra. *A karbantartó lap*

Végül lépjünk be `Balla/Tigris-3` kezelőként



11. 49. ábra Kezelő sem tud belépni.