

FÉLÉVES FELADAT
az Adatbázis-Használat 2. c tárgyhoz

NÉV: **Hendrich Attila**

NEPTUN KÓD: BAB03T

A DOLGOZAT CÍME:

Beadási határidő:
13. oktatási hét

Szerepjáték karakterlap

FELADATKIÍRÁS

Készítsen egy web-es adatbázis-alkalmazást az Oracle11g APEX környezetében. Az alkalmazás célja egy szerepjáték karakterlapjainak tárolása és webes felületen való szerkesztése. Az adatbázisban tárolni kell a karakterek alaptulajdonságait, a tárgyainak és képességeinek a listáját. Készítsen valamilyen C-típusú nyelven egy véletlen karaktert generálni képes programot, melyet e nyelv dinamikus könyvtárán keresztül hív meg az alkalmazás.

A dolgozatnak tartalmaznia kell az alkalmazási környezet bemutatását, az elkészített rendszer tervezését és a végleges adatmodellt, valamint a C-típusú nyelven írt program algoritmusának ismertetését, mellékletként pedig a kifejlesztett alkalmazás fejlesztői környezetének (a benne szereplő riportok, objektumok, eljárások, és az alkalmazás egyéb rendszeralkotó objektumainak bemutatását, a tesztfutások jegyzőkönyvét, valamint CD-n a dolgozat anyagát word dokumentum formátumban, az exportált alkalmazás szkriptjét, és text file-okban a rendszer adatállományát.

2013. szeptember 18.

.....
Hendrich Attila

.....
Nagy István
adjunktus

NORMALIZÁLT TÁBLA SPECIFIKÁCIÓ:

1. lépés: Tárolandó objektumok:

játekos<NÉV, FAJ, KASZT, ERŐSSÉG, VÉDELEM, GYORSASÁG, CÉLZÁS, PÉNZ>

Ez egy játékos tulajdonságait, jellemzőit tartalmazó objektum. A játék egy fantasy világban játszódik, ahol az adott játékosok különböző fajú fantasy lények egyikét irányíthatják. Ezen játékban egy *faj* (pl. ember, elf, törp stb.) fogja meghatározni a a játékos fizikai tulajdonságait. (*védelem*, *gyorsaság*, *célzás*) A *kaszt* ebben a játékban csak egy szöveges mező mely az adott játékos játékbeli szerepét határozza meg. (Pl. egy harcos feladata a társai védelme, egy tolvaj feladata a besurranás, osonás az ellenfelek mellett, míg mondjuk egy varázsló feladata többi játékosok gyógyítása.)

A pénz értelemszerűen a játékos játékban birtokolt fizetőeszközének mennyiségét jelenti.

faj<FAJ_NÉV, ERŐSSÉG, VÉDELEM, GYORSASÁG, CÉLZÁS>

Éz az objektum határozza meg az adott faj szülöttjének fizikai tulajdonságait. (Tehát egy faj névhez tartozik, hogy az adott játékos milyen erős, gyors stb. lesz.)

tárgy<TÁRGY_NÉV, TULAJDONOS_NÉV, MENNYISÉG, VISELT>

A játékban a játékosok tárgyakat birtokolhatnak, ezek lehetnek például fogyóeszközök (pl. ételek), illetve felszerelési tárgyak (pl. fegyverek, ruházat, páncélzat. Egy ilyen objektum a *tárgy nevét*, a *tulajdonosát*, az adott tulajdonos által birtokolt *mennyiséget* tárolja. Emellett felszerelési tárgy esetén azt, hogy a tulajdonos éppen magán *viseli-e*, vagy *sem*. (Értelemszerűen például egy páncél csak akkor véd, ha a tulajdonosa éppen felvette.)

képesség<KÉPESSÉG_NÉV, TULAJDONOS_NÉV, TÍPUS>

A játékban a játékosok különböző képességeket tanulhatnak meg, ezeket tárolják el ezek az objektumok. Ilyen képességek (pl. varázslatok) *névvel vannak azonosítva*. A *tulajdonos név* az adott képességet birtokló játékos neve lesz, míg végül a *típus* azt definiálja, hogy az adott képesség offenzív, defenzív vagy egyéb. Ez az információ ahhoz szükséges, hogy milyen helyzetben használható a képesség.

2. lépés: Függőségek felismerése

Játékos neve egyértelműen azonosítja a játékost

$f_{\text{játekos}} = \{\text{NÉV}\} \rightarrow \{\text{FAJ, KASZT, ERŐSSÉG, VÉDELEM, GYORSASÁG, CÉLZÁS, PÉNZ}\}$

Faj neve egyértelműen azonosítja a hozzátartozó attribútumokat

$f_{\text{faj}} = \{\text{FAJ_NÉV}\} \rightarrow \{\text{ERŐSSÉG, VÉDELEM, GYORSASÁG, CÉLZÁS}\}$

Tárgyakknál és képességeknél a név és a tulajdonos neve együtt azonosítja a tárgyat és képességet

$f_{\text{tárgy}} = \{\text{TÁRGY_NÉV, TULAJDONOS_NÉV}\} \rightarrow \{\text{MENNYISÉG, VISELT}\}$

$f_{\text{képesség}} = \{\text{KÉPESSÉG_NÉV, TULAJDONOS_NÉV}\} \rightarrow \{\text{TÍPUS}\}$

3. lépés: Normalizálás

$f_{\text{játekos}}$

1NF-ben van, mert nincs összetett attribútum

2NF-ben van, mert 1NF-ben van és minden jobb oldalon álló attribútum függ az elsődleges attribútumoktól, és nem függ az elsődleges attribútumok egy részhalmazától.

3NF-t sérti, mivel játékos tulajdonságai a fajától függenek, ezeket innét leválasztjuk, így 3NF-nek megfelel
 $f_{\text{játékos1}} = \{\text{NÉV}\} \rightarrow \{\text{FAJ, KASZT, PÉNZ}\}$
 BCNF-ben van, mert 3NF-ben van és nincs kulcstörés

$f_{\text{játékos}}$

1NF-ben van, mert nincs összetett attribútum
 2NF-ben van, mert 1NF-ben van és minden jobb oldalon álló attribútum függ az elsődleges attribútumoktól, és nem függ az elsődleges attribútumok egy részhalmazától.
 3NF-ben van, mert 2NF-ben van és a másodlagos attribútumok között nincs függés (nincs tranzitív függés).
 BCNF-ben van, mert 3NF-ben van és nincs kulcstörés

f_{faj}

1NF-ben van, mert nincs összetett attribútum
 2NF-ben van, mert 1NF-ben van és minden jobb oldalon álló attribútum függ az elsődleges attribútumoktól, és nem függ az elsődleges attribútumok egy részhalmazától.
 3NF-ben van, mert 2NF-ben van és a másodlagos attribútumok között nincs függés (nincs tranzitív függés).
 BCNF-ben van, mert 3NF-ben van és nincs kulcstörés

$f_{\text{tárgy}}$

1NF-ben van, mert nincs összetett attribútum
 2NF-ben van, mert 1NF-ben van és minden jobb oldalon álló attribútum függ az elsődleges attribútumoktól, és nem függ az elsődleges attribútumok egy részhalmazától.
 3NF-ben van, mert 2NF-ben van és a másodlagos attribútumok között nincs függés (nincs tranzitív függés).
 BCNF-ben van, mert 3NF-ben van és nincs kulcstörés

$f_{\text{képesség}}$

1NF: Mindegyik tábla 1NF-ben van, nincs benne összetett attribútum
 2NF: Mindegyik tábla 2NF-ben van, jobb oldalon lévő tulajdonságok mindenütt a bal oldalon lévő kulcstól, vagy kulcsoktól függenek
 3NF: Az $f_{\text{játékos}}$ tábla nincs 3NF-ben, mivel a játékos tulajdonsága a fajától függ
 $f_{\text{játékos1}} = \{\text{NÉV}\} \rightarrow \{\text{FAJ, KASZT, PÉNZ}\}$
 BCNF: Mindegyik tábla BCNF-ben van, mert a táblák baloldalának egyetlen valódi részhalmaza se függ egyetlen elsődleges, vagy másodlagos attribútumtól.

4. lépés: Kapott táblák összefoglaló felsorolása:

$f_{\text{játékos1}} = \{\text{NÉV}\} \rightarrow \{\text{FAJ, KASZT, PÉNZ}\}$
 $f_{\text{faj}} = \{\text{FAJ_NÉV}\} \rightarrow \{\text{ERŐSSÉG, VÉDELEM, GYORSASÁG, CÉLZÁS}\}$
 $f_{\text{tárgy}} = \{\text{TÁRGY_NÉV, TULAJDONOS_NÉV}\} \rightarrow \{\text{MENNYISÉG, VISELT}\}$
 $f_{\text{képesség}} = \{\text{KÉPESSÉG_NÉV, TULAJDONOS_NÉV}\} \rightarrow \{\text{TÍPUS}\}$

5. lépés: Idegen kulcsok:

- "tulajdonos_név" a "tárgy" és "képesség" táblában
- "faj" a "játékos" táblában

6. lépés: Táblák attribútumai típussal:

játékos:

név: varchar(30)

faj: varchar(30)

kaszt: varchar(30)

pénz: int

faj:

faj_név: varchar(30)

erősség: int

védelem: int

gyorsaság: int

célzás: int

tárgy:

tárgy_név: varchar(30)

tulajdonos_név: varchar(30)

mennyiség: int

viselt: int(1)

Megjegyzés: boolean típus hiányában

képesség:

képesség_név: varchar(30)

tulajdonos_név: varchar(30)

BEADOTT FÁJLOK

alkalmazas_export.sql Maga az alkalmazás exportja
create_tables.sql Adattáblákat létrehozó script
ora.cs Az alkalmazásban használt külső függvény könyvtár C# forrása
Oracle.dll Az alkalmazásban használt külső függvény könyvtár
function_melyikfaj.sql Az alkalmazásban használt pl/sql függvények létrehozás
function_split2.sql
function_ujfaj.sql
export_faj.txt Az adattáblák exportjai
export_jatekos.txt
export_kespesseg.txt
export_targy.txt
feladatkiiras.pdf Ez a dokumentum

ALKALMAZÁS TELEPÍTÉSE

1. Adattáblák létrehozása

1. Apexben SQL Workshop / SQL Scripts menüpont alatt töltjük fel a mellékelt *create_tables.sql* scriptet

```

1 -- Tablak létrehozása
2 create table faj(
3     faj_nev varchar(30) not null,
4     erosseg int not null,
5     vedelem int not null,
6     gyorsasag int not null,
7     celzas int not null,
8     constraint faj_primary_key primary key (faj_nev)
9 );
10 create table jatekos(
11     nev varchar(30) not null,
12     faj varchar(30) not null,
13     kaszt varchar(30) not null,
14     penz int,
15     constraint jatekos_primary_key primary key (nev),
16     constraint jatekos_foreign_key foreign key (faj) references faj (faj_nev) on delete cascade
17 );
18 create table targy(
19     targy_nev varchar(30) not null,
20     tulajdonos_nev varchar(30) not null,
21     mennyiseg int not null,
22     viselt int not null,
23     constraint targy_primary_key primary key (targy_nev,tulajdonos_nev),
24     constraint targy_foreign_key foreign key (tulajdonos_nev) references jatekos (nev) on delete cascade
25 );
26 create table kepesseg(
27     kepesseg_nev varchar(30) not null,
28     tulajdonos_nev varchar(30) not null,
29     tipus varchar(30) not null,
30     constraint kepesseg_primary_key primary key (kepesseg_nev,tulajdonos_nev),
31     constraint kepesseg_foreign_key foreign key (tulajdonos_nev) references jatekos (nev) on delete cascade
32 );
    
```

2. Futtasuk a scriptet

2. Adattáblák feltöltése

1. APEX-ben SQL Workshop, Data Workshop menüpontja
2. Data Load blokk Text Load linkje
3. Existing table választása
4. Elsőre válasszuk ki a faj táblát
5. Tallózzuk ki az *export_faj.txt* fájlt és importáljuk
6. Ismételjük a maradék három táblára a második ponttól. (*export_jatekos.txt*, *export_kespesseg.txt*, *export_targy.txt*)

3. Külső függvény könyvtár importálása

1. Adjuk meg a jogosultságot a rendszer felhasználónak, hogy külső könyvtárakat tudjon használni, a tantárgy segéd anyagának alapján sqlplus-ban, vagy sql-developerben

```

1 -- 03_1APELDA.sql
2
3 -- SQLPlus-ban futtatni!
4 -- (sys as sysdba/OraclE-123)
5 -- vagy az SQL DEveloperben beállítani
6 -- az alábbi jogosultságokat
7
8 GRANT CREATE ANY LIBRARY,
9     EXECUTE ANY TYPE,
10    EXECUTE ANY PROCEDURE,
11    ALTER ANY LIBRARY,
12    DROP ANY LIBRARY,
13    DROP ANY TYPE,
14    DROP ANY PROCEDURE
15 TO Sema1;
16
    
```

2. Másoljuk a mellékelt Oracle.dll-t egy az APEX számára elérhető munka könyvtárba
3. APEX SQL Workshop menüpontja alatt az SQL commands menü alatt futtassuk az alábbi utasítást:

```
CREATE OR REPLACE LIBRARY FAJOKLIB AS  
'C:\Oracle\11gr2\app\Administrator\product\11.2.0\dbhome_1\BIN\Oracle.dll';
```

(Értelemszerűen az elérésiútban a bemásolt dll elérési útját adjuk meg.)

4. Függvények létrehozása

1. Az APEX SQL Workshop SQL commands menüpontja
2. Másoljuk ide a *function_melyikfaj.sql* tartalmát és futtassuk le
3. Ismételjük ezt a másik két fájl tartalmára is (*function_split2.sql*, *function_ujfaj.sql*)

5. Hozzuk létre a szükséges felhasználókat

1. APEX Administration menüpontja alatt a Task-ok között Create User
2. Hozzuk létre a "user" felhasználónevű felhasználót
3. Ha nem lenne "scott" felhasználó, akkor őt is hozzuk létre

6. Importáljuk az alkalmazást

1. APEX Application Workshop Import/Export pontja alatt
2. Import application képernyőn tallózzuk ki az *alkalmazas_export.sql* fájlt, és alapértelmezett beállításokkal importáljuk
3. Ekkor megjelenik a "karakterlapok" alkalmazás az Application Workshopban. Most már futtatható.

AZ ALKALMAZÁS BEMUTATÁSA

1. Játékosok karakterlapjainak böngészése

Az alkalmazás első lapja, a Játékos oldal listázza szűrhető módon a már felvett játékosok karaktereinek alap adatait. Ez a lap be nem jelentkezett felhasználók számára is elérhető.

Név	Faj	Kaszt	Penz
Adeib-i	Featherfolk	Magus	364
Aeto	Demon	Ijász	314
Alesaw	Elf	Barbar	13
Angol	Ember	Barbar	263
Ashrak	Elf	Barbar	115
Augh-ald	Ember	Techno	341
Bathnai	Elf	Barbar	378
Belny	Demon	Ijász	150

A listában a játékos nevére kattintva (*drill-down*) a játékos karakterlapja lesz látható, ahol láthatóak a részletes adatait, a birtokolt tárgyai, képességei.

A drill-down működéshez a report régió oszlop beállításainál a név oszlopból csináltam linket, ami átadja a név értékét a karakterlap egy mezőjének.

A karakterlap betöltődésekor (before header) process-ben kitöltődik a játékos neve alapján az adatbázisból kikérdezett tulajdonságait (faj, kaszt, erő stb.) egy egy rejtett mezőbe. Megjelenítéskor a bal oldali html régió ezekből emeli ki a tartalmakat. A kép elérési útja a faj neve alapján generált távoli URL.

A karakterlapon a játékos tárgyai tabuláris űrlap módon szerkeszthetők.

Megjegyzés: A karakterlap és minden további nem adminisztratív lap eléréséhez a "user", "scott" vagy "admin" felhasználónevű felhasználóval kell bejelentkezve lenni.

Karakter	Kiválasztott karakter	Adeib-i
Faja	Featherfolk	
Kasztja	Magus	
Erősség	3	
Védelem	2	
Gyorsaság	6	
Céltzás	3	
Pénz	364	

Tárgy	Név	Mennyiség	Viselt
<input type="checkbox"/>	Potion	28	<input type="radio"/> Viselt <input checked="" type="radio"/> Nem viselt

KEPESSEG_NEV	TÍPUS
Kisütes	Offenzív
Napsütes	Offenzív

2. Játékosok karaktereinek képességeire és tárgyaira való szűrés

A tárgyak és képességek fül alatt karakterekhez tartozó tárgyakat és képességeket kérdezhetünk le. Itt van lehetőség ezek szerkesztésére form on report módon.

The screenshot shows the 'Karakterlapok' application with the 'Tárgyak' (Items) tab selected. The interface includes a search bar, a table of items, and a list of abilities.

Karakterlapok

Játékos: [Angol]

Search: [Q-] Go Actions

Tárgy Név	Mennyiség	Viselt
Potion	24	0

Abilities:

- ☐ Ladata
- ☐ Lyeonim
- ☐ Norjl
- ☐ Pedid
- ☐ Rod-ton
- ☐ Roufnys-dyn
- ☐ Seris
- ☐ Shyeng
- ☐ Snehat
- ☐ Styachiss
- ☐ Tharoddyn
- ☐ Uskend
- ☐ Waldis
- ☐ Yaiq
- ☐ Loljly
- ☐ Lyzyk
- ☐ Nunez
- ☐ Rean
- ☐ Ronli
- ☐ Samdenther
- ☐ Sersul
- ☐ Shying
- ☐ Soard
- ☐ Sulach
- ☐ Tur-ny
- ☐ Uskr
- ☐ Wardenray
- ☐ Ylora
- ☒ Lothech
- ☐ Naltone
- ☐ Oomi
- ☐ Riss
- ☐ Rothor
- ☐ Seria
- ☐ Shisayad
- ☐ Skelat
- ☐ Socore
- ☐ Tanone
- ☐ Unt-Moni
- ☐ Ver-ild
- ☐ Xofir
- ☐ Yusyca

Search: [Q-] Go Actions

Képesség Név	Tulajdonos Név	Típus
Kisütes	Naltone	Offenzív
Napsütes	Naltone	Offenzív
Visszavonulás	Naltone	Defenzív

1 - 3

3. Fajok módosítása

Fajok tábla szerkesztéséhez a fajok fület kell választani. Itt szintén tabuláló form formájában lehet a táblát módosítani.

The screenshot shows the 'Karakterlapok' application with the 'Fajok' (Races) tab selected. The interface includes a table for editing race statistics.

Karakterlapok

Játékos: [Angol]

Search: [Q-] Go Actions

Faj Név	Erősség	Védelem	Gyorsaság	Céltás
Demon	6	5	2	1
Elf	2	2	4	6
Ember	4	4	3	3
Featherfolk	3	2	6	3

1 - 4

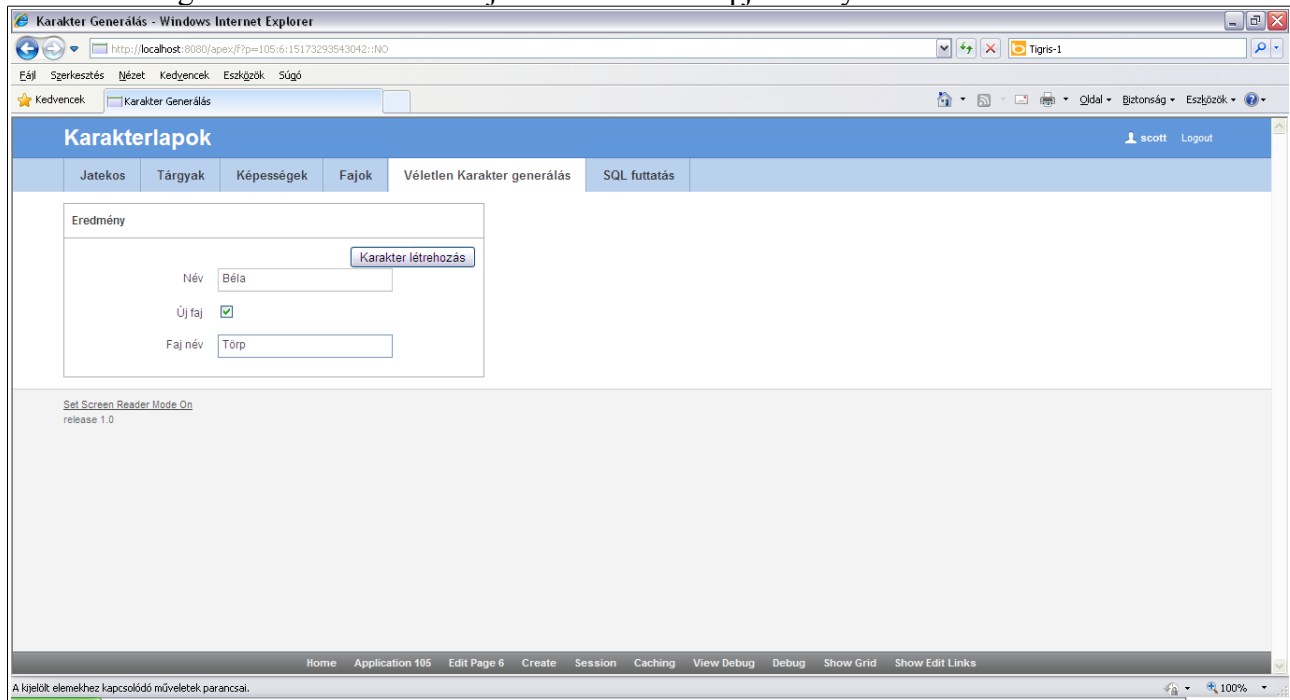
Új faj

4. Új karakter generálása

Új karakter generálásához a külső DLL két függvényét használjuk. Ez egy adminisztratív feladat, így csak "scott" és "admin" felhasználónevű felhasználó érheti el ezt a lapot.

Karakter generálásnál meg kell adni egy nevet, illetve új fajt is lehet véletlen generálni ha a betesszük az új faj pipálósdobozba a pipát. Ez esetben meg kell adni az új faj nevét.

Ha a karakter generálás sikeres volt az új karakter karakterlapjára irányít át az alkalmazás.



Használt függvények a processben:

Külső dll függvényei:

UJFAJ(): Ez visszaadja a generált faj tulajdonságait egy kettősponttal tagolt karakterláncban

MELYIKFAJ(FAJOK): Ez elvárja a fajok neveinek listáját egy kettősponttal tagolt karakterláncban, majd visszatér az egyikkel

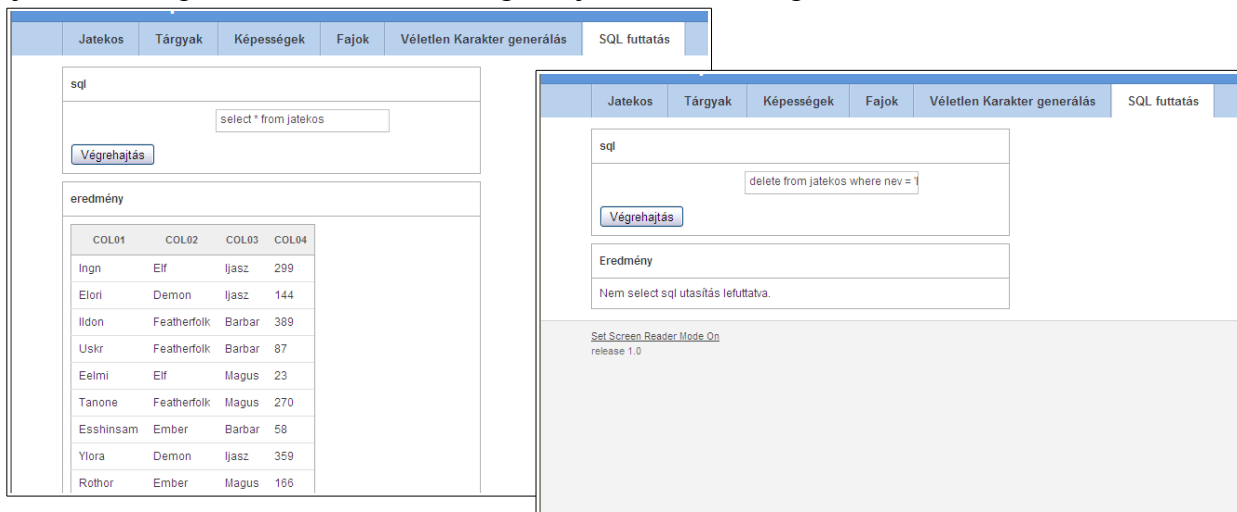
PL/SQL függvény:

SPLIT2(): Ezen segéd függvénnyel bontom szét több részre az UJFAJ() által visszaadott karakter láncot

Az átirányításért egy branch felel

5. SQL utasítások futtatása

Az utolsó fül (mely szintén adminisztratív feladat) lehetőséget biztosít SQL parancsok közvetlen futtatására. Lekérdezések esetén az úralp alatt egy reportban lesz látható az eredmény. Egyéb esetben csak tájékoztatást kap a felhasználó a feladat végrehajtásának sikerességéről.



HITELESÍTÉSI SÉMÁK

A lapok elérhetőségét két (illetve ha az alapértelmezetten elérhető "mindenki számára látható" sémát is bele vesszük három) hitelesítési séma garantálja. Az én két külső sémám:

1. user_es_admin:

Ezen sémával ellátott lapokat user, scott és admin felhasználónevű felhasználók böngészhetik

```
7
8 begin
9 if :APP_USER = 'ADMIN' or :APP_USER = 'SCOTT' or :APP_USER = 'USER'
10 then return true;
11 else return false;
12 end if;
13 end;
14
15
```

2. csak_admin

Ezen sémával ellátott lapokat csak scott és admin felhasználónevű felhasználók böngészhetik

```
1 begin
2 if :APP_USER = 'ADMIN' or :APP_USER = 'SCOTT'
3 then return true;
4 else return false;
5 end if;
6 end;
```

Sémák nem teljesülése esetén magyar nyelvű hibaüzenetet fog látni a felhasználó.

