

Több táblás lekérdezések

Az eddigiekben olyan lekérdezéseket csináltunk, amelyben a szükséges információt 1 tábla tartalmából ki tudtuk szűrni.

Például:

Listázza ki az összes dolgozó nevét és főnökének azonosítóját.

(*select ename as "Név", mgr as "Főnök azon" from emp*)

Ezt könnyen megtehetjük, hiszen ugyanabban a táblában tároljuk a dolgozó nevét (*ename*) és főnökének azonosítóját (*mgr*).

De vannak olyan esetek, mikor a szükséges információ 2 különböző táblában van tárolva.

Például:

Listázza ki az összes dolgozó nevét, részlegének azonosítóját, részlegének nevét és a telephely címét.

Ha megnézzük az *emp* tábla felépítését könnyen beláthatjuk, hogy az egyetlen részlegre vonatkozó információt tartalmaz, ami a részleg azonosító (*deptno*). A maradék szükséges információt (részleg neve, telephely címe) a *dept* táblában találjuk. A *dept* táblában szintén találunk egy *deptno* oszlopot, amely kulcsfontosságú lesz számunkra a továbbiakban.

A SELECT utasítás egyszerűsített alakja:

SELECT SzelekciósLista

FROM tábla1 [MásodlagosTáblanév1],

tábla2 [MásodlagosTáblanév2]
[...]

[WHERE LogikaiFeltétel]

[GROUP BY ...]

[HAVING ...]

[ORDER BY...];

A SELECT utasítás egyszerűsített alakjában láthatjuk, hogy a *from* utasítás után, nem csak egy táblát adhatunk meg forrásnak, hanem vesszővel elválasztva többet is.

Például: *select * from emp, dept;*

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30	10	ACCOUNTING	NEW YORK
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20	10	ACCOUNTING	NEW YORK
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20	10	ACCOUNTING	NEW YORK
7369	SMITH	CLERK	7902	12/17/1980	800	-	20	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30	10	ACCOUNTING	NEW YORK
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30	10	ACCOUNTING	NEW YORK
7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30	10	ACCOUNTING	NEW YORK
7876	ADAMS	CLERK	7788	01/12/1983	1100	-	20	10	ACCOUNTING	NEW YORK
7900	JAMES	CLERK	7698	12/03/1981	950	-	30	10	ACCOUNTING	NEW YORK
7934	MILLER	CLERK	7782	01/23/1982	1300	-	10	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30	20	RESEARCH	DALLAS
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10	20	RESEARCH	DALLAS
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20	20	RESEARCH	DALLA

Tisztán láthatjuk, hogy az előbbi utasítás lefut ugyan, de hibás, mert nem használtunk úgynevezett táblaösszekötő feltételt (14 bejegyzés van az emp táblában, az eredménytáblánk több mint 14 bejegyzést tartalmaz, ilyenkor gyanakodhatunk, hogy a megoldásunk hibás). Ilyenkor az eredmény a 2 táblánk descartes szorzata lesz (*emp* tábla összes sorát összeköti a *dept* tábla összes sorával).

Ha megnézzük az eredményt, akkor láthatjuk, hogy a 2 tábla deptno oszlopa nem minden esetben egyezik meg. Ez számunkra elengedhetetlen lenne, hiszen ha ez a két oszlop (*emp* tábla *deptno* oszlopa és *dept* tábla *deptno* oszlopa) megegyezik, az számunkra hasznos információt ad. Eredménytáblát nézve rájöhetünk, hogy például BLAKE (második sor) részlegének azonosítója 30 (első deptno (*emp* tábla)) míg az ACCOUNTING részleg azonosítója 10 (második deptno(*dept* tábla)). Ez annyit tesz, hogy BLAKE NEM a NEW YORK-i ACCOUNTING részlegen dolgozik, csak a tábla összekötő feltételünket nem használtuk.

Most, hogy tudjuk, hogy a 2 tábla deptno oszlopának megegyezésével juthatunk helyes információhoz, ezt a feltételt megadhatjuk LogikaiFeltétel-ként.

Tehát: *select * from emp, dept where emp.deptno=dept.deptno;*

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7782	CLARK	MANAGER	7839	06/09/1981	2450	-	10	10	ACCOUNTING	NEW YORK
7934	MILLER	CLERK	7782	01/23/1982	1300	-	10	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT	-	11/17/1981	5000	-	10	10	ACCOUNTING	NEW YORK
7902	FORD	ANALYST	7566	12/03/1981	3000	-	20	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	12/09/1982	3000	-	20	20	RESEARCH	DALLAS
7566	JONES	MANAGER	7839	04/02/1981	2975	-	20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	12/17/1980	800	-	20	20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7788	01/12/1983	1100	-	20	20	RESEARCH	DALLAS
7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	1400	30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	09/08/1981	1500	0	30	30	SALES	CHICAGO
7900	JAMES	CLERK	7698	12/03/1981	950	-	30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30	30	SALES	CHICAGO
7698	BLAKE	MANAGER	7839	05/01/1981	2850	-	30	30	SALES	CHICAGO

14 rows returned in 0.03 seconds

Itt egy új dologgal szembesülhetünk: *emp.deptno=dept.deptno*

Mivel mind a kettő táblánkban van egy deptno oszlop, ezért meg kell adnunk, hogy melyik tábla deptno oszlopára gondolunk. Ezt a *táblanév.oszlopnév* formában tehetjük meg.

Amennyiben ezt nem tesszük meg (*select * from emp, dept where emp.deptno=deptno*), úgy a következő hibaüzenetet kapjuk: ORA-00918: Column ambiguously defined

Az oszlop kétértelműen van definiálva. A rendszer nem tudja, hogy melyik tábla deptno oszlopára szeretnénk hivatkozni.

Ezen ismeretekkel már meg tudjuk oldani a feljebb említett feladatot:

Listázza ki az összes dolgozó nevét, részlegének azonosítóját, részlegének nevét és a telephely címét.

```
Select ename, emp.deptno, dept.deptno, dname, loc
      from emp,dept
      where emp.deptno=dept.deptno
      order by ename desc;
```

ENAME	EMP.DEPTNO	DEPT.DEPTNO	DNAME	LOC
WARD	30	30	SALES	CHICAGO
TURNER	30	30	SALES	CHICAGO
SMITH	20	20	RESEARCH	DALLAS
SCOTT	20	20	RESEARCH	DALLAS
MILLER	10	10	ACCOUNTING	NEW YORK
MARTIN	30	30	SALES	CHICAGO
KING	10	10	ACCOUNTING	NEW YORK
JONES	20	20	RESEARCH	DALLAS
JAMES	30	30	SALES	CHICAGO
FORD	20	20	RESEARCH	DALLAS
CLARK	10	10	ACCOUNTING	NEW YORK
BLAKE	30	30	SALES	CHICAGO
ALLEN	30	30	SALES	CHICAGO
ADAMS	20	20	RESEARCH	DALLAS

Természetesen az eddig tanultak ugyanúgy működnek.

Listázzuk ki a 20-as vagy 30-as részlegen dolgozók nevét, részlegének azonosítóját, részlegének nevét és a telephely címét.

```
Select ename, emp.deptno, dept.deptno, dname, loc
      from emp, dept
      where emp.deptno=dept.deptno and (emp.deptno=20 or emp.deptno=30)
      order by ename desc
```

ENAME	DEPTNO	DEPTNO	DNAME	LOC
WARD	30	30	SALES	CHICAGO
TURNER	30	30	SALES	CHICAGO
SMITH	20	20	RESEARCH	DALLAS
SCOTT	20	20	RESEARCH	DALLAS
MARTIN	30	30	SALES	CHICAGO
JONES	20	20	RESEARCH	DALLAS
JAMES	30	30	SALES	CHICAGO
FORD	20	20	RESEARCH	DALLAS
BLAKE	30	30	SALES	CHICAGO
ALLEN	30	30	SALES	CHICAGO
ADAMS	20	20	RESEARCH	DALLAS

A zárójelezésre odafigyelni! Ha nem használnánk zárójelet rossz eredményhez jutnánk, hiszen először az AND, aztán az OR logikai művelet fog lefutni.

Zárójel nélkül a 20 as részlegazonosítójú részlegen dolgozókra jó eredményt kapnánk, míg a 30 as részlegazonosítójú részlegen dolgozókra rosszat, hiszen arra már nem szólna a táblamegszorító feltételünk.

A feladatot megoldhatnánk a következő képpen is:

```
Select ename, emp.deptno, dept.deptno, dname, loc
      from emp, dept
      where emp.deptno=dept.deptno and emp.deptno in(20,30)
      order by ename desc
```

emp.deptno in(20,30): Oszlopnév in(elem1,elem2,elem3...). Visszatérési értéke igaz lesz (kielégíti a feltételt) Amennyiben az adott oszlopban található érték felveszi a zárójelben megadott "lista" egyik elemének az értékét.

Tehát ugyanaz, mintha OR al összekapcsolnánk (*emp.deptno=20 or emp.deptno=30*).

Allekérdezések

Vannak olyan esetek, mikor a számunkra szükséges információ a rendelkezésünkre álló táblákból nem nyerhető ki. Allekérdezés használata esetén célszerű az allekérdezésünket egy összefüggő karaktersorozattal elnevezni.

Például:

Listázza ki részlegenként a legnagyobb fizetéssel rendelkező dolgozó nevét, munkakörét, fizetését, részlegének a nevét, és telephelyét.

Legmagasabb fizetésekről a tábláinkban nem tárolunk információt, ezért azt nekünk kell megszerezniünk lekérdezés útján.

Először listázzuk ki a legmagasabb fizetéseket részlegazonosító szerint csoportosítva:

select deptno, max(sal) from emp group by deptno;

Részlegazonosító Maximum fizetés	
30	2850
20	3000
10	5000

Mostmár minden részleghez megvan a legnagyobb fizetésünk. Gyakorlatilag azokra az emberekre vagyunk kíváncsiak, akiknek a fizetése megegyezik ebben az eredményben leírtakkal.

Többtáblás lekérdezéseknél nem csak kész táblát használhatunk, hanem használhatunk úgynevezett allekérdezéseket is táblaként.

Olyan, mintha a lekérdezés eredménye egy külön tábla lenne.

Ehhez a következőt kell tennünk: Az előbb elkészített maximum fizetéses lekérdezésünket a táblalistába zárójelek segítségével beillesztjük. A zárójel után pedig adunk neki egy nevet (*al*).

Az allekérdezésben szereplő oszlopokat nevezzük át, hiszen később majd ezen a néven tudunk rá hivatkozni (*maxfiz*).

*select emp.deptno, ename, job, sal, dname, loc
from emp, dept, (select deptno as reszlegazon, max(sal) as maxfiz
from emp group by deptno) al
where emp.deptno=dept.deptno and al.maxfiz=sal and emp.deptno=al.deptno;*

DEPTNO	ENAME	JOB	SAL	DNAME	LOC
10	KING	PRESIDENT	5000	ACCOUNTING	NEW YORK
30	BLAKE	MANAGER	2850	SALES	CHICAGO
20	SCOTT	ANALYST	3000	RESEARCH	DALLAS
20	FORD	ANALYST	3000	RESEARCH	DALLAS

Láthatjuk, hogy gyakorlatilag minen olyan embert megjelenítettünk, akinek a fizetése megegyezik az allekérdezésben lévő maximum fizetésekkel.

Where emp.deptno=dept.deptno and al.maxfiz=sal and emp.deptno=al.deptno

Mivel gyakorlatilag 3 táblánk van, ezért 2 összekötő feltételt kellett írunk.

Amennyiben hagyjuk az *emp.deptno=al.deptno* összekötő feltételt, eredményünk akkor is jó lesz, a tábla sajátosságai miatt. Viszont, ha mondjuk lenne egy olyan dolgozó a 10-es részlegen, aki

mondjuk szintén 3000-et keresne, akkor az *emp.deptno=al.deptno* feltétel nélkül rossz eredményt kapnánk, ugyanis őt is kilistázná. 2 emberünk lenne a 10-es részlegről: a KING, aki 5000-et keres és a másik ember, aki 3000-et keres. De mivel 5000>3000 ezért őt nem szabad kilistáznunk.

Amennyiben allekérdezéseket használunk, egy rutinra és egy jó tervre van szükség, hogy rájöjjünk, hogy az allekérdezésünkkel milyen információkhoz jussunk.

Korrelált lekérdezés

Többtáblás lekérdezéseknél nem csak különböző táblákat használhatunk, hanem használhatjuk többször ugyanazt a táblát is. Ilyenkor kötelező a táblának másodlagos nevet adni. Jó példa erre a könyvben található feladat:

Listázzuk ki az emp tábla alapján a főnökökhöz tartozó beosztottak nevét és részlegük azonosítóját célszerű módon.

```
Select f.ename as fonoknev,  
       d.ename as dolgozonev,  
       d.deptno as reszlegnev  
from emp f,  
       emp d  
where f.empno=d.mgr  
order by f.ename,  
       d.ename;
```

Ugyanez allekérdezéssel:

```
select fonoknev,  
       ename as dolgozonev,  
       deptno as reszlegnev  
from emp,  
(select empno as fonokazon,  
     ename as fonoknev  
  from emp) al  
where al.fonokazon=mgr  
order by fonoknev,  
       dolgozonev;
```

Ha az allekérdezésben szereplő oszlopot szeretnénk kilistázni, akkor simán hivatkozhatunk a szelekciós listában annak nevével. Jelen esetben: fonoknev.

Az eredmény mindkét esetben:

FONOKNEV DOLGOZONEV RESZLEGNEV		
BLAKE	ALLEN	30
BLAKE	JAMES	30
BLAKE	MARTIN	30
BLAKE	TURNER	30
BLAKE	WARD	30
CLARK	MILLER	10
FORD	SMITH	20
JONES	FORD	20
JONES	SCOTT	20
KING	BLAKE	30
KING	CLARK	10
KING	JONES	20
SCOTT	ADAMS	20