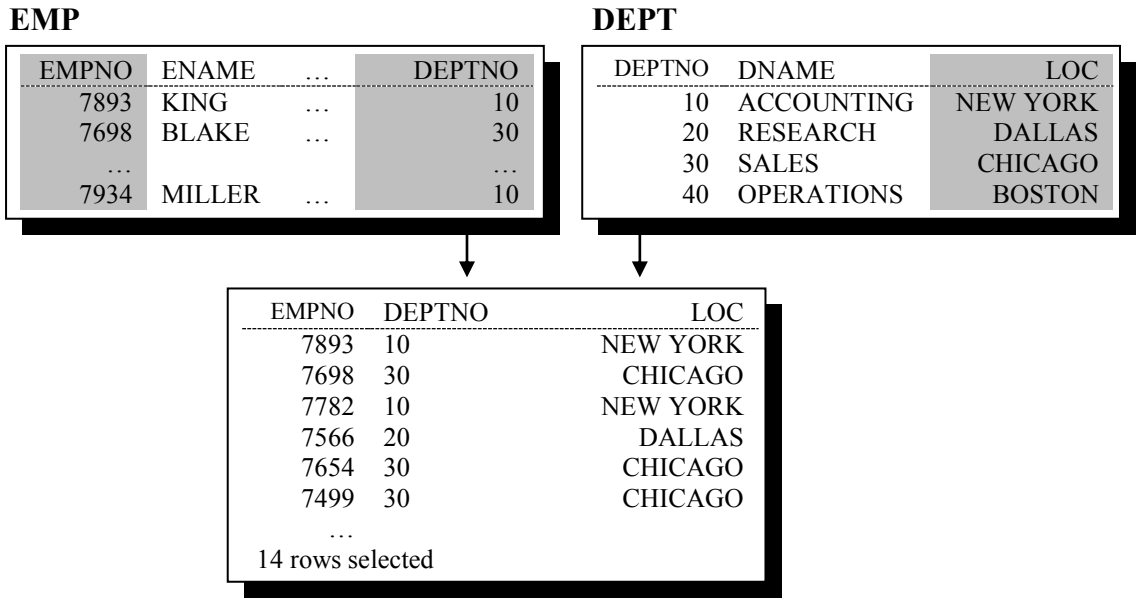
METODE DE SELECŢIE A DATELOR DIN TABELE MULTIPLE

Obiective

În acest referat sunt prezentate diferite metode utilizate pentru extragerea datelor din mai multe tabele.

După parcurgerea acestui referat, studentul va avea cunoştinţele necesare:

- scrierii unei expresii SELECT pentru a accesa date din mai multe tabele folosind legături (joncţiuni) de egalitate si nonegalitate;
- vizualizării datelor care nu îndeplinesc o condiţie de joncţiune folosind condiţii de joncţiune externă;
- efectuării unei joncţiuni între un tabel şi el însuşi.



Există situaţii când trebuie să accesăm date din mai multe tabele. În exemplul de mai sus, rezultatul afişat conţine date din două tabele separate.

- atributul `EMPNO` există în tabelul `EMP;`
- atributul DEPTNO există în ambele tabele `EMP` şi `DEPT;`
- atributul LOC există în tabelul `DEPT`.

Pentru a obţine rezultatul dorit trebuie realizată o legătură între tabelele `EMP` şi `DEPT`.

Definirea joncţiunilor

Vom folosi o condiţie de joncţiune ori de câte ori trebuie să accesăm date din mai multe tablele din baza de date. Se poate crea o corespondenţă între liniile unui tabel şi liniile altui tabel pe baza valorilor comune existente în coloanele corespondente, care sunt de obicei chei primare şi străine.

Pentru afişarea datelor din două sau mai multe tabele aflate în relaţie se va scrie o simplă condiţie de joncţiune în clauza `WHERE`:

```
SELECT  tabel1.coloana1, tabel1.coloana2, tabel2.coloana3
FROM  tabel1, tabel2
WHERE tabel1.coloana1 = tabel2.coloana3;
```

unde:

tabel.coloana	indică tabelul şi coloana de unde este extrasă data
tabel1.coloana1 =	este condiţia care leagă
tabel2.coloana2	cele două tabele

Observaţii:

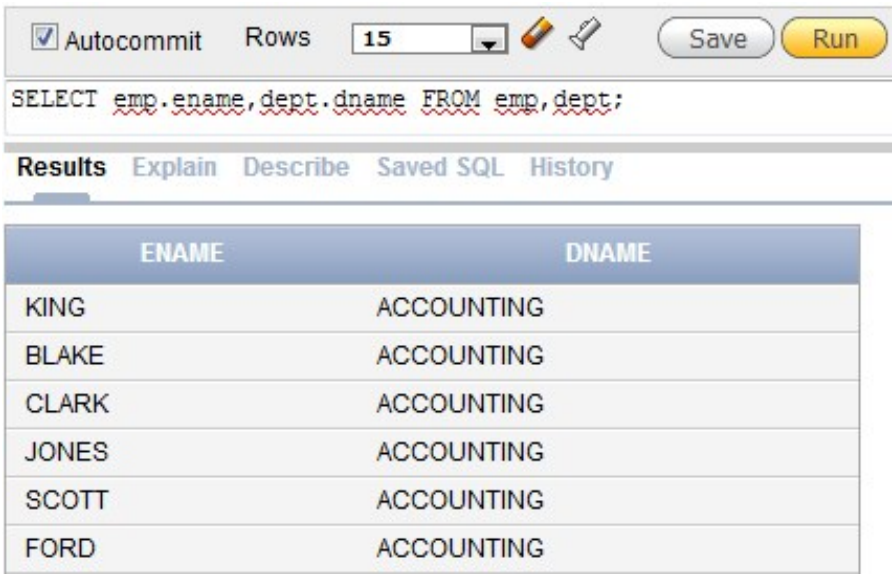
- în momentul scrierii unei expresii SELECT care conţine o condiţie de joncţiune, este indicat ca numele coloanelor să fie precedate de numele tabelului de care aparţin; în acest fel este mărită claritatea codului SQL şi se îmbunătăţeşte accesul la baza de date.
- dacă un acelaşi nume de coloană apare în mai multe tabele, numele coloanei trebuie prefixat cu numele tabelului de care aparţine.
- pentru a realiza o legătură între *n* tabele este nevoie de minim *n-1* condiţii de joncţiune (e.g. pentru a lega 4 tabele sunt necesare 3 joncţiuni). Această regula s-ar putea să nu se aplice dacă tabelul are o cheie primară formată din mai multe atribute şi astfel este necesară mai mult de o coloană pentru a identifica în mod unic fiecare linie.

Produsul Cartezian

Atunci când o condiţie de joncţiune este invalidă sau complet omisă, rezultatul este un produs cartezian în care vor fi afişate toate combinaţiile de linii din tabelele implicate. Un produs cartezian tinde să genereze un număr mare de linii, iar rezultatul său este în general nefolositor. De aceea trebuie inclusă întotdeauna o condiţie de joncţiune validă în clauza `WHERE`, cu excepţia cazului când se doreşte în mod explicit combinarea tuturor liniilor din tabele implicate în relaţie.

Exemplul următor afişează numele fiecărui angajat şi numele departamentului în care lucrează din tabelele `EMP` şi `DEPT`. Deoarece nu a fost specificată clauza `WHERE`, toate liniile (14) din tabelul `EMP` sunt combinate cu toate liniile (4) din tabelul `DEPT`, generând astfel 56 de linii în tabelul rezultat.

```
SELECT emp.ename, dept.dname FROM emp, dept;
```



ENAME	DNAME
KING	ACCOUNTING
BLAKE	ACCOUNTING
CLARK	ACCOUNTING
JONES	ACCOUNTING
SCOTT	ACCOUNTING
FORD	ACCOUNTING

Tipuri de condiţii de joncţiune

Principalele tipuri de condiţii de joncţiune sunt:
1. echi-joncţiune;
2. non-echi-joncţiune.

Pe lângă acestea mai există şi alte tipuri de condiţii de joncţiune:
3. joncţiune externă;
4. joncţiune între un tabel şi el însuşi.

1. Echi-joncţiuni

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20

...
14 rows selected.

DEPT

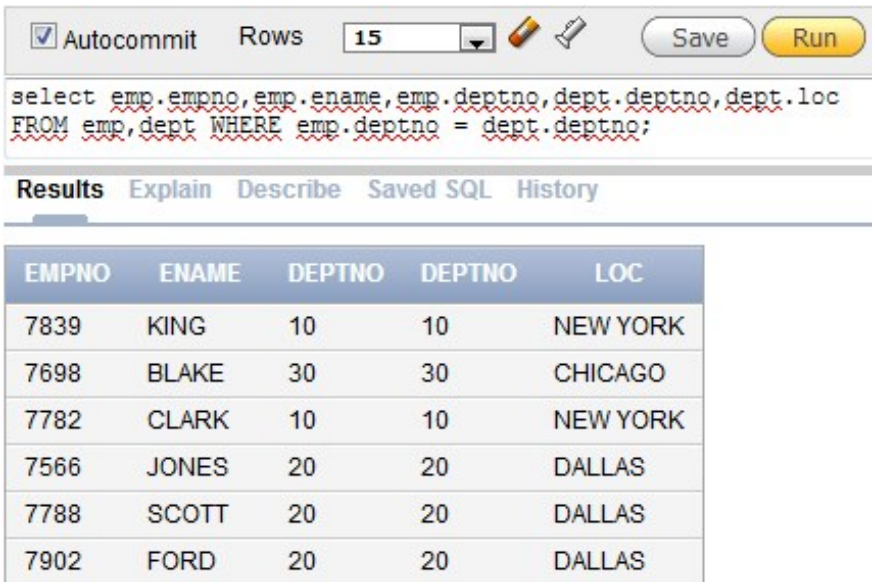
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS

...
14 rows selected.

Cheie străină **Cheie primară**

Pentru a determina numele departamentului unui angajat trebuie comparată valoarea din coloana `DEPTNO` din tabelul `EMP` cu valorile `DEPTNO` din tabelul `DEPT`. Legătura astfel creată între tabelele `EMP` şi `DEPT` este o echi-joncţiune - valorile din coloana `DEPTNO` din ambele tabele trebuie să coincidă.

```
SELECT  emp.empno, emp.ename, emp.deptno,dept.deptno, dept.loc
FROM  emp, dept WHERE emp.deptno=dept.deptno;
```



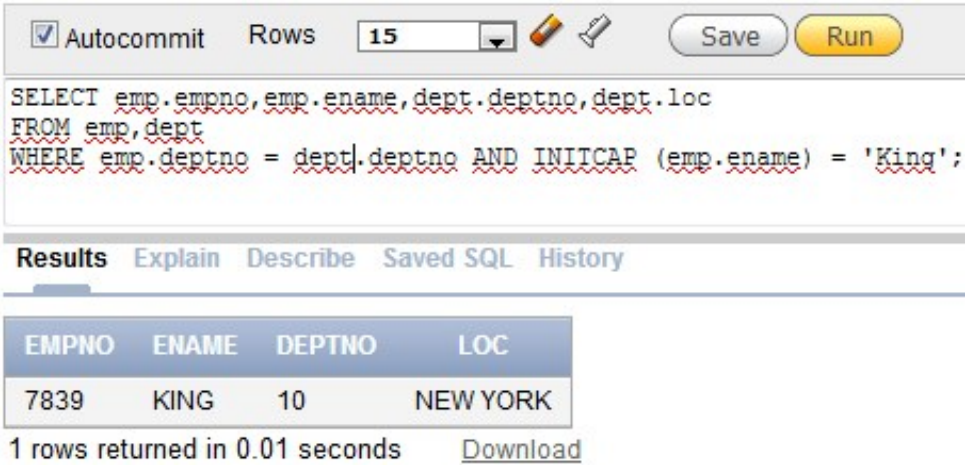
EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7566	JONES	20	20	DALLAS
7788	SCOTT	20	20	DALLAS
7902	FORD	20	20	DALLAS

În exemplul de mai sus:

- Clauza `SELECT` specifică numele coloanelor ce vor fi afişate
  - numele şi numărul angajatului şi numărul departamentului, care sunt coloane în tabelul `EMP`;
  - numărul, numele şi locaţia departamentului, sunt coloane în tabelul `DEPT`.
- Clauza `FROM` specifică cele 2 tabele ce conţin informaţiile utile:
  - tabelul `EMP`
  - tabelul `DEPT`
- Clauza `WHERE` specifică modul în care se va realiza legătura între tabele.

Suplimentar condiţiei de joncţiune, clauza `WHERE` poate conţine şi alte criterii necesare operaţiei de selecţie a datelor. De exemplu, pentru a afişa codul angajatului KING, numele, numărul şi locaţia departamentului este nevoie de o condiţie suplimentară în clauza `WHERE`.

```
SELECT  emp.empno, emp.ename, dept.deptno, dept.loc
FROM  emp, dept
WHERE emp.deptno= dept.deptno AND INITCAP(emp.ename) = 'King';
```



Alias-uri de tabele

Calificarea coloanelor cu ajutorul numelui tabelului poate consuma mult timp, mai ales în cazul în care numele tabelului este un şir lung. Pentru simplificarea codului SQL se pot folosi alias-uri de tabele în locul numelor acestora.

```
SELECT  e.empno, e.ename, e.deptno, d.deptno, d.loc
FROM  emp e, dept d
WHERE e.deptno=d.deptno;
```

Observatii:

- Alias-urile de tabel pot avea lungimea maximă de 30 caractere;
- dacă în clauza `FROM` se introduce un alias pentru tabel, atunci acel alias trebuie să înlocuiască numele tabelului în toată instrucţiunea `SELECT`;
- alias-urile de tabel ar trebui sa fie semnificative;
- alias-ul unui tabel este valid numai pentru `SELECT`-ul curent.

2. Non-echi-joncţiuni

EMP

EMPNO	ENAME	SAL
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950

...
14 rows selected.

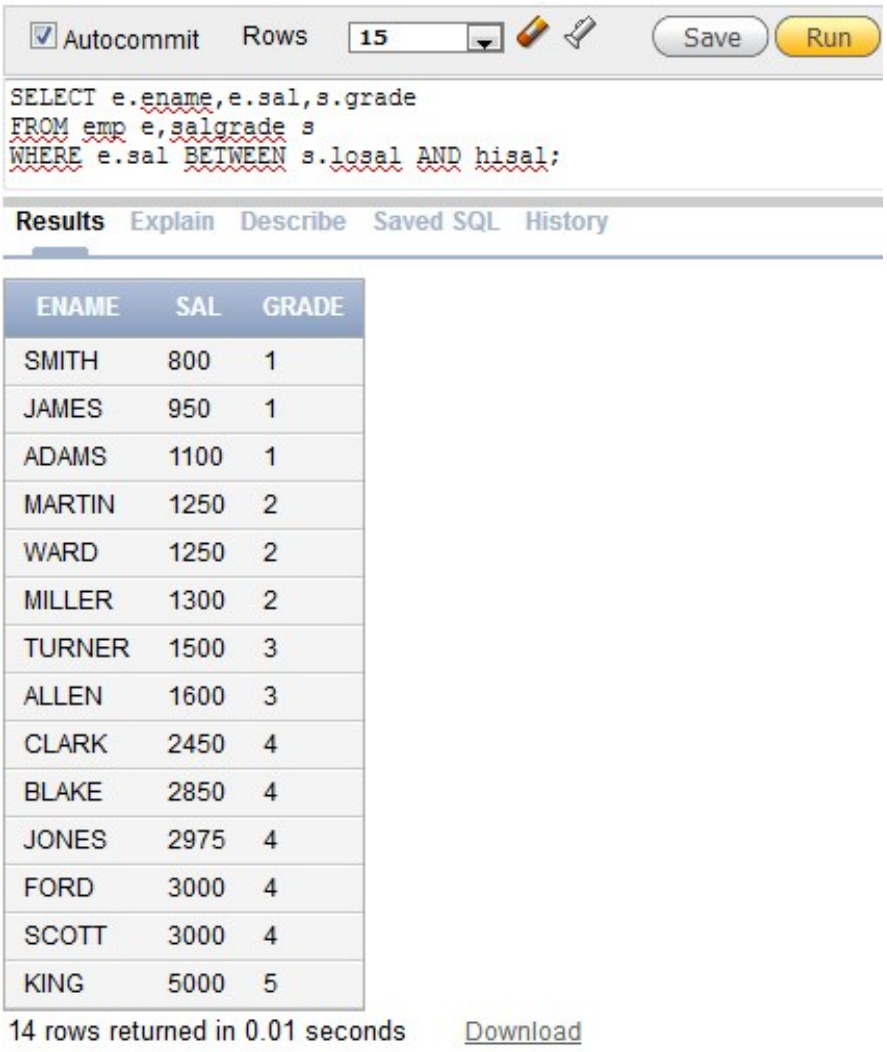
SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

"salariul din tabelul EMP este între salariul minim şi maxim din tabelul SALGRADE"

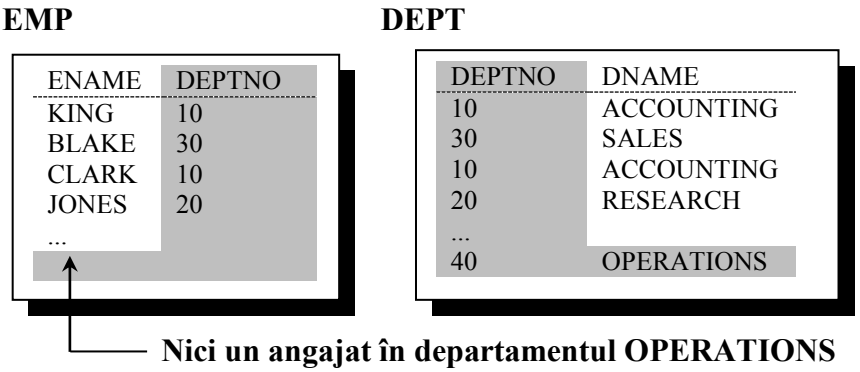
Relaţia dintre tabelul `EMP` si `SALGRADE` este o non-echi-joncţiune, în sensul că nici o coloana din tabelul `EMP` nu corespunde direct unei coloane din tabelul `SALGRADE`. Legătura dintre cele două tabele este următoarea: coloana `SAL` din `EMP` este cuprinsă între coloanele `LOSAL` şi `HISAL` din tabelul `SALGRADE`. Legătura se obţine folosind un operator, altul decât operatorul egal (‘=’).

```
SELECT  e.name, e.sal, s.grade
FROM emp e, salgrade s
WHERE e.sal BETWEEN s.losal AND s.hisal
```



Exemplul de mai sus crează o non-echi-joncţiune pentru a determina gradul de salarizare al unui angajat. Salariul trebuie să fie între (`between`) limita inferioară (LOSAL) şi cea superioară (HISAL) a unui nivel de salarizare.

3. Joncţiune externă



Dacă o linie (înregistrare) nu satisface condiţia de joncţiune, acea linie nu va apare în rezultatul interogării. De exemplu, în condiţia de echi-joncţiune a tabelelor `EMP` şi `DEPT`, departamentul OPERATIONS nu va apare pentru că nu există nici o persoană care lucrează în acel departament.

Liniile lipsă pot fi returnate dacă în condiţia de joncţiune se utilizează operatorul de joncţiune externă. Operatorul este semnul "+" între paranteze - (+) şi este plasat în acea parte a condiţiei de joncţiune corespunzătoare tabelului deficient în informaţii. Acest operator are ca efect crearea uneia sau a mai multor linii nule la care se poate adaugă una sau mai multe linii din tabelul ce conţine informaţiile neselectate.

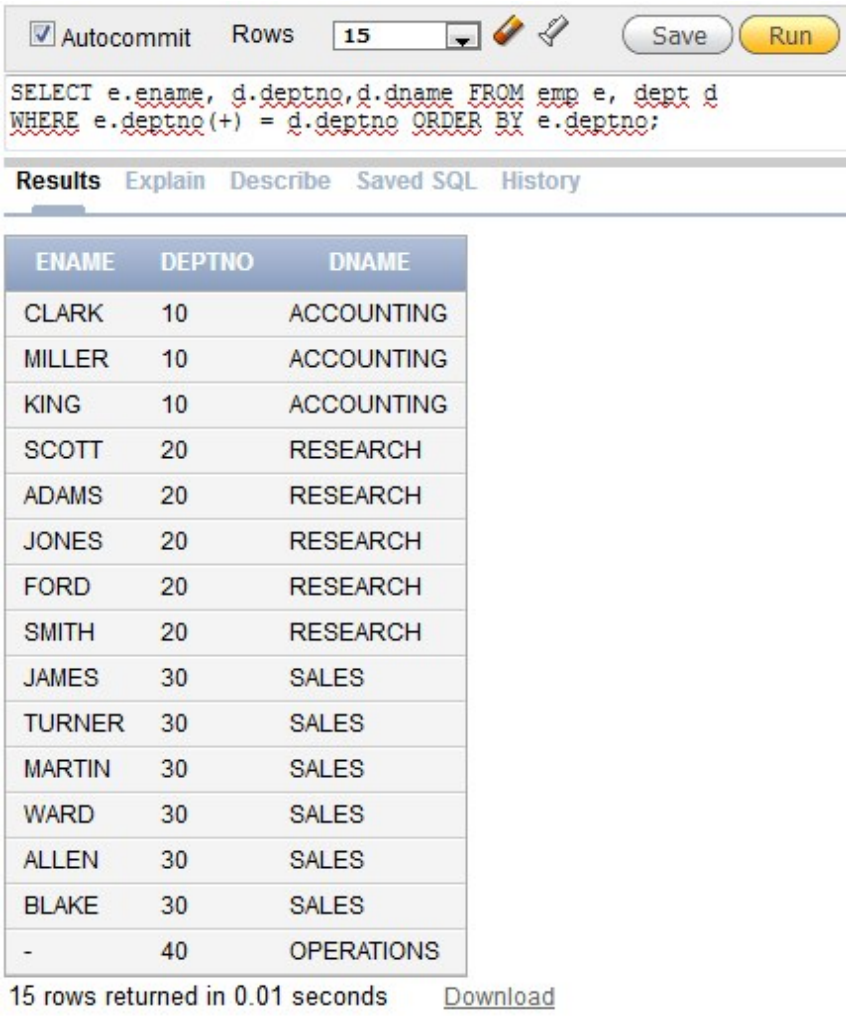
```
SELECT  tabel.coloana1, tabel.coloana2, ...
FROM  tabel1, tabel2
WHERE tabel1.coloana1 = tabel2.coloana2(+) ;
```

unde:

`tabel1.coloana1 =`	este condiţia care realizează legătura între tabele
`tabel2.coloana2(+)`	este simbolul pentru joncţiune externă; poate fi plasat in oricare parte a clauzei `WHERE`, dar nu în ambele părţi simultan. Se va plasa operatorul de joncţiune externă după numele coloanei din tabelul deficitar în informaţii.

Următorul exemplu afişează toate numerele şi numele departamentelor. Departamentul OPERATIONS, care nu are nici un angajat este de asemenea afişat.

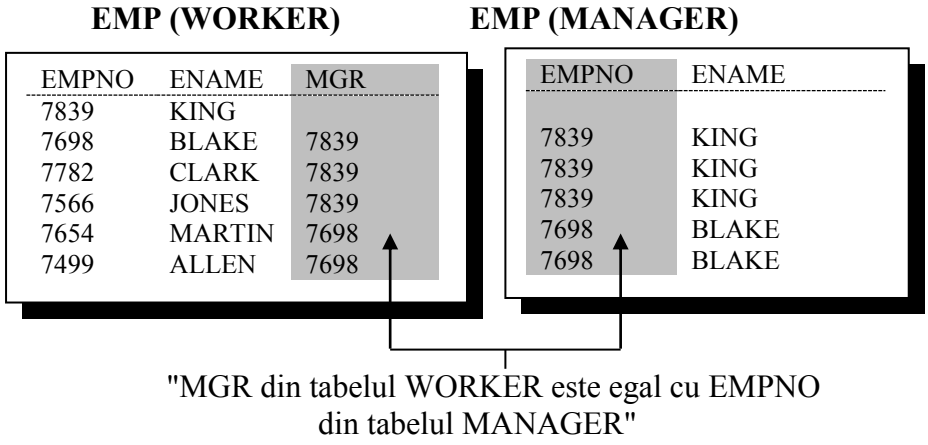
```
SELECT  e.ename, d.deptno, d.dname
FROM  emp e, dept d
WHERE e.deptno(+) = d.deptno
 ORDER BY e.deptno ;
```



Restricţii în cazul utilizării unei condiţii de joncţiune externă:

- operatorul de joncţiune externă poate apare numai într-o singură parte a unei expresii - partea care nu deţine informaţia ce nu este returnată de interogare. El returnează acele linii din tabel care nu au corespondent direct în celălalt tabel.
- o condiţie ce implică operatorul de joncţiune externă nu poate utiliza operatorul IN şi nici nu poate fi legată de o altă condiţie prin operatorul OR.

4. Condiţii de joncţiune ale unui tabel cu el însuşi



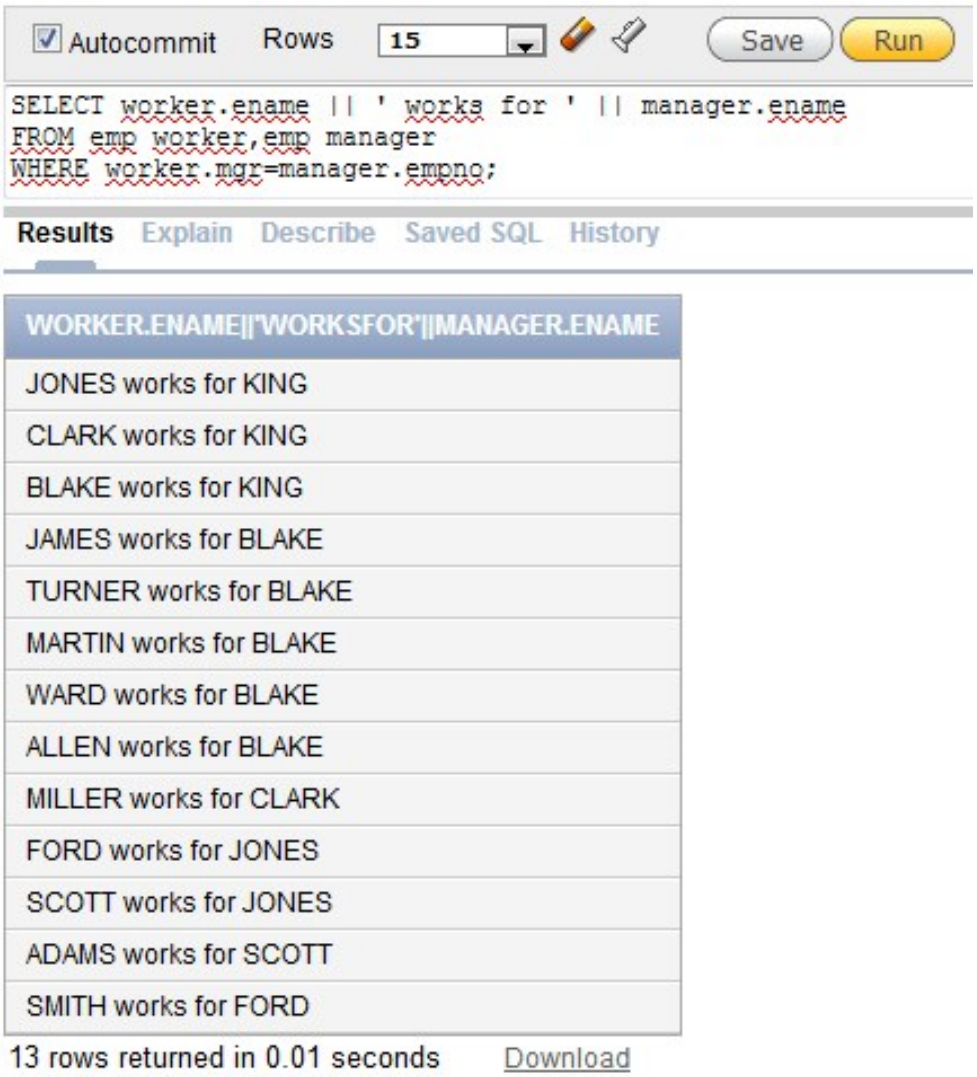
"MGR din tabelul WORKER este egal cu EMPNO din tabelul MANAGER"

Unele aplicaţii impun realizarea unei joncţiuni între un tabel şi el însuşi. De exemplu, pentru a găsi numele managerului lui Blake va trebui să:

- găsim înregistrarea corespunzătoare angajatului Blake în tabelul EMP, inspectând coloana ENAME;
- găsim codul managerului lui Blake din coloana MGR. Codul managerului lui Blake este 7839.
- găsim numele managerului având codul EMPNO egal cu 7893 în coloana ENAME. Codul angajatului King este 7839. Deci King este managerul lui Blake.

Pe parcursul acestui proces am căutat în tabel de două ori. Prima dată pentru a-l găsi pe Blake în coloana ENAME şi pentru a citi valoarea MGR - 7839. A doua oară am căutat în coloana EMPNO valoarea 7839 şi am extras din coloana ENAME valoarea King.

```
SQL> SELECT worker.ename || 'works for' || manager.ename
  2  FROM emp worker, emp manager
  3  WHERE  worker.mgr = manager.empno
```



Exemplul de mai sus crează o legătură între tabelul EMP şi el însuşi. Pentru a simula două tabele în clauza FROM, se folosesc două aliasuri, numite WORKER şi MANAGER pentru acelaşi tabel EMP. În acest exemplu, clauza WHERE conţine condiţia de joncţiune care înseamnă "pentru care codul managerului unui angajat coincide cu codul de angajat al managerului".

Probleme

1. Scrieti o interogare care să afişeze numele fiecărui angajat, precum şi numărul şi numele departamentului în care lucrează.
2. Creaţi un listing unic pentru toate funcţiile (`job`) angajaţilor din departamentul `SALES`.
3. Scrieţi o interogare care afişează numele, numele departamentului şi locaţia departamentului pentru toţi angajaţii ce au dreptul la comision.
4. Afişaţi numele angajatului şi numele departamentului pentru toţi angajaţii al căror nume conţine un caracter 'A'.
5. Scrieţi o interogare care afişează numele, funcţia, numărul departamentului şi numele departamentului pentru toţi angajaţii care lucrează în DALLAS.
6. Afişaţi numele angajaţilor şi codurile lor, împreună cu numele managerilor şi codurile acestora. Redenumiţi coloanele Employee, Emp#, Manager şi Mgr#.
7. Modificaţi interogarea anterioară pentru a afişa toţi angajatii, inclusiv pe King care nu are manager.
8. Creaţi o interogare care va afişa numele fiecărui angajat, numărul departamentului în care lucrează şi numele tuturor angajaţilor care lucrează în acelaşi departament. Redenumiţi cât mai expresiv coloanele.

DEPARTMENT	EMPLOYEE	COLLEAGUE
10	CLARK	KING
10	CLARK	MILLER
10	KING	CLARK
10	KING	MILLER
10	MILLER	CLARK
10	MILLER	KING
20	ADAMS	FORD
20	ADAMS	JONES
20	ADAMS	SCOTT
20	ADAMS	SMITH
20	FORD	ADAMS
20	FORD	JONES
20	FORD	SMITH

9. Afişaţi structura tabelului `SALGRADE`. Creaţi o interogare care va afişa numele, funcţia, numele departamentului, salariul şi gradul de salarizare pentru toţi angajaţii.
10. Creaţi o interogare care afişează numele şi data angajării pentru lucrătorii angajaţi după data de angajare a lui Blake.
11. Afişaţi toate numele angajaţilor şi datele de angajare împreună cu numele managerilor şi data lor de angajare pentru toţi cei care au fost angajaţi înaintea managerilor lor. Etichetaţi coloanele Employee, Emp, Hiredate, Manager şi Mgr Hiredate.