Baze de date-Anul 1
Laborator 5 SQL

Cereri multi-relaţie

I. [Join]

Join-ul este operaţia de regăsire a datelor din două sau mai multe tabele, pe baza valorilor comune ale unor coloane. De obicei, aceste coloane reprezintă cheia primară, respectiv cheia externă a tabelelor.

Condiţia de *join* se poate scrie şi în clauza *WHERE* a instrucţiunii *SELECT*. Într-o instrucţiune *SELECT* care uneşte tabele prin operaţia de *join*, se recomandă ca numele coloanelor să fie precedate de numele sau alias-urile tabelelor pentru claritate şi pentru îmbunătăţirea timpului de acces la baza de date. Dacă acelaşi nume de coloană apare în mai mult de două tabele, atunci numele coloanei se prefixează **obligatoriu** cu numele sau alias-ul tabelului corespunzător. Pentru a realiza un *join* între ***n* tabele**, va fi nevoie de cel puţin ***n* – 1 condiţii de *join***.

Tipuri de join :

- ***Inner join (equijoin, join simplu)*** – corespunde situaţiei în care valorile de pe coloanele ce apar în condiţia de *join* trebuie să fie egale.
- ***Nonequijoin -*** condiţia de *join* conţine alţi operatori decât operatorul egalitate.
- ***Left | Right | Full Outer join*** – un *outer join* este utilizat pentru a obţine în rezultat şi înregistrările care nu satisfac condiţia de *join*. Operatorul pentru *outer join* este semnul plus inclus între paranteze **(+)**, care se plasează în acea parte a condiţiei de *join* care este **deficitară în informaţie**. Efectul acestui operator este de a uni liniile tabelului care nu este deficient în informaţie şi cărora nu le corespunde nici o linie în celălalt tabel cu o linie cu valori *null*. Operatorul (+) poate fi plasat în orice parte a condiţiei de *join*, dar nu în ambele părţi.

  Full outer join – left outer join + right outer join.

Obs: O condiţie care presupune un ***outer join*** nu poate utiliza operatorul *IN* şi nu poate fi legată de altă condiţie prin operatorul *OR*.

Obs: Un caz special al operaţiei de join este ***self join*** – join-ul unui tabel cu el însuşi.

Obs : *Alias*-urile pot fi utilizate oriunde, ca o notaţie mai scurtă, în locul denumirii tabelului. Ele pot avea lungimea de maxim 30 de caractere, dar este recomandat să fie scurte şi sugestive. Dacă este atribuit un alias unui tabel din clauza FROM, atunci el trebuie să înlocuiască apariţiile numelui tabelului în instrucţiunea SELECT.

Un alias poate fi utilizat pentru a califica denumirea unei coloane. Calificarea unei coloane cu numele sau alias-ul tabelului se poate face :

- opţional, pentru claritate şi pentru îmbunătăţirea timpului de acces la baza de date;
- obligatoriu, ori de câte ori există o ambiguitate privind sursa coloanei. Ambiguitatea constă, de obicei, în existenţa unor coloane cu acelaşi nume în mai mult de două tabele.

***Join* în standardul SQL3 (SQL:1999):**

Pentru *join*, sistemul *Oracle* oferă şi o sintaxă specifică, introdusă de către standardul *SQL3* (*SQL:1999*). Această sintaxă nu aduce beneficii în privinţa performanţei faţă de *join*-urile care se specifică în clauza *WHERE*. Tipurile de *join* conforme cu *SQL3* sunt definite prin cuvintele cheie *CROSS JOIN* (pentru produs cartezian), *NATURAL JOIN*, *FULL OUTER JOIN*, clauzele *USING* şi *ON*.

Sintaxa corespunzătoare standardului *SQL3* este următoarea:

SELECT *tabel_1.nume_coloană*, *tabel_2.nume_coloană*

FROM *tabel_1*
[***CROSS JOIN*** *tabel_2*]
| [***NATURAL JOIN*** *tabel_2*]
| [***JOIN*** *tabel_2* ***USING*** (*nume_coloană*)]
| [***JOIN*** *tabel_2* ***ON*** (*tabel_1.nume_coloană = tabel_2.nume_coloană*)]
| [***LEFT | RIGHT | FULL OUTER JOIN*** *tabel_2*
ON (*tabel_1.nume_coloană = tabel_2.nume_coloană*)];

- *NATURAL JOIN* presupune existenţa unor coloane având acelaşi nume în ambele tabele. Clauza determină selectarea liniilor din cele două tabele, care au valori egale în aceste coloane. Dacă tipurile de date ale coloanelor cu nume identice sunt diferite, va fi returnată o eroare.

  Coloanele având acelaşi nume în cele două tabele trebuie să nu fie precedate de numele sau *alias*-ul tabelului corespunzător.

- *JOIN tabel_2 USING nume_coloană* efectuează un *equijoin* pe baza coloanei cu numele specificat în sintaxă. Această clauză este utilă dacă există coloane având acelaşi nume, dar tipuri de date diferite. Coloanele referite în clauza ***USING*** trebuie să nu conţină calificatori (să nu fie precedate de nume de tabele sau *alias*-uri) în nici o apariţie a lor în instrucţiunea *SQL*. Clauzele *NATURAL JOIN* şi *USING* **nu pot coexista** în aceeaşi instrucţiune *SQL*.

- ***JOIN tabel_2 ON tabel_1.nume_coloană = tabel_2.nume_coloană*** efectuează un *equijoin* pe baza condiţiei exprimate în clauza *ON*. Această clauză permite specificarea separată a condiţiilor de *join*, respectiv a celor de căutare sau filtrare (din clauza *WHERE*).

- *LEFT*, *RIGHT* şi *FULL OUTER JOIN tabel_2 ON* (*tabel_1.nume_coloană = tabel_2.nume_coloană*) efectuează *outer join* la stânga, dreapta, respectiv în ambele părţi pe baza condiţiei exprimate în clauza *ON*.

Un *join* care returnează rezultatele unui *inner join*, dar şi cele ale *outer join*-urilor la stânga şi la dreapta se numeşte *full outer join*.

II. [Exerciţii]

1. Să se listeze job-urile angajaţilor care lucrează în departamentul 30.
2. Să se afişeze numele salariatului şi numele departamentului pentru toţi salariaţii care au litera A inclusă în nume.
3. Să se afişeze numele, job-ul, codul şi numele departamentului pentru toţi angajaţii care lucrează în Oxford.
4. Sa se determine codurile angajatilor si numele job-urilor al caror salariu este mai mare decat 3000 sau este egal cu media dintre salariul minim si salariul maxim aferente job-ului respectiv.
5. Sa se afişeze codul departamentului, numele departamentului, numele şi job-ul tuturor angajaţilor din departamentele al căror nume conţine şirul ‘ti’. Rezultatul se va ordona alfabetic după numele departamentului, şi în cadrul acestuia, după numele angajaţilor.
6. Să se afişeze codul angajatului şi numele acestuia, împreună cu numele şi codul şefului său direct. Se vor eticheta coloanele Ang#, Angajat, Mgr#, Manager.
7. Să se afişeze numele şi data angajării pentru salariaţii care au fost angajaţi după *Gates*.
8. Sa se afişeze codul şi numele angajaţilor care lucrează în acelaşi departament cu cel puţin un alt angajat al cărui nume conţine litera “t”. Se vor afişa, de asemenea, codul şi numele departamentului respectiv. Rezultatul va fi ordonat alfabetic după nume.
9. Sa se afişeze numele, salariul, titlul job-ului, oraşul şi ţara în care lucrează angajatii condusi direct de **King**.

Operatori pe mulţimi.

I. [Operatori pe mulţimi]

Operatorii pe mulţimi combină rezultatele obţinute din două sau mai multe interogări. Cererile care conţin operatori pe mulţimi se numesc ***cereri compuse***. Există patru operatori pe mulţimi: *UNION*, *UNION ALL*, *INTERSECT* şi *MINUS*.

Toţi operatorii pe mulţimi au aceeaşi precedenţă. Dacă o instrucţiune *SQL* conţine mai mulţi operatori pe mulţimi, *server*-ul *Oracle* evaluează cererea de la stânga la dreapta (sau de sus în jos). Pentru a schimba această ordine de evaluare, se pot **utiliza paranteze**.

- ***Operatorul UNION*** returnează toate liniile selectate de două cereri, eliminând duplicatele. Acest operator nu ignoră valorile *null* şi are precedenţă mai mică decât operatorul *IN*.
- Operatorul ***UNION ALL*** returnează toate liniile selectate de două cereri, fără a elimina duplicatele. Precizările făcute asupra operatorului *UNION* sunt valabile şi în cazul operatorului *UNION ALL*. În cererile asupra cărora se aplică *UNION ALL* nu poate fi utilizat cuvântul cheie ***DISTINCT***.
- Operatorul ***INTERSECT*** returnează toate liniile comune cererilor asupra cărora se aplică. Acest operator nu ignoră valorile *null*.
- Operatorul ***MINUS*** determină liniile returnate de prima cerere care nu apar în rezultatul celei de-a doua cereri.

Observaţii:

- În mod implicit, pentru toţi operatorii cu excepţia lui *UNION ALL*, **rezultatul este ordonat crescător** după valorile primei coloane din clauza *SELECT*.
- Pentru o cerere care utilizează operatori pe mulţimi, cu excepţia lui *UNION ALL*, *server*-ul *Oracle* elimină liniile duplicat.
- În instrucţiunile *SELECT* asupra cărora se aplică operatori pe mulţimi, coloanele selectate trebuie să corespundă ca număr şi tip de date. Nu este necesar ca numele coloanelor să fie identice. Numele coloanelor din rezultat sunt determinate de numele care apar în clauza *SELECT* a primei cereri.

1. Se cer codurile departamentelor al căror nume conţine şirul “re” sau în care lucrează angajaţi având codul job-ului “SA_REP”.

2. Sa se obtina codurile departamentelor in care nu lucreaza nimeni (nu este introdus nici un salariat in tabelul *employees*). Se cer două soluţii.

```
SELECT department_id
FROM departments
WHERE department_id NOT IN (SELECT department_id
                            FROM employees);
```

? Este corecta aceasta varianta? De ce ?