

LABORATOR 2: CERERI MULTITABEL FUNCTII (INTRODUCERE)

Recapitulare: SELECT

```
SELECT {{{DISTINCT|UNIQUE}|ALL}|lista_campuri|*}  
FROM [nume_schema.]nume_obiect]  
[, [nume_schema.]nume_obiect ...]  
[WHERE conditie_clauza_where]  
[START WITH conditie_clauza_start_with  
CONNECT BY conditie_clauza_connect_by]  
[GROUP BY expresie[, expresie ...]  
[HAVING conditie_clauza_having]]  
[ORDER BY {expresie|pozitie}[, {expresie|pozitie} ...]]  
[FOR UPDATE  
[OF [[nume_schema.]nume_obiect.]nume_coloana  
[, [[nume_schema.]nume_obiect.]nume_coloana] ...]  
[NOWAIT | WAIT numar_intreg]];
```

Clauza **ORDER BY** determină ordonarea liniilor selectate de o cerere, după valorile expresiilor precizate. Expresiile după care se face ordonarea, separate prin virgulă, se evaluează de la stânga la dreapta astfel:

se ordonează liniile rezultate după prima expresie;

liniile pentru care calculul acelei expresii conduc la același rezultat se ordonează după a doua expresie etc.

În această clauză pot fi specificate expresii sau numere întregi reprezentând pozițiile unor expresii în lista **SELECT**. Astfel, ordonarea poate avea loc după valorile expresiilor care apar în clauză sau după valorile expresiilor din lista **SELECT** afișate pe poziția specificată în clauză. Cuvintele cheie **ASC** și **DESC** indică modul în care se face ordonarea, după valorile crescătoare, respectiv descrescătoare, ale expresiei specificate. Implicit, ordinea este cea crescătoare. Dacă se omite clauza **ORDER BY**, atunci ordinea în care sunt regăsite înregistrările poate să difere de la o execuție la alta.

Exemplul 1: Să se afișeze angajații care lucrează în departamentul 30 în ordinea alfabetică.

```
SELECT last_name, first_name, department_id  
FROM employees  
WHERE department_id = 30  
ORDER BY last_name, first_name;
```

Exercițiul 2: Să se afișeze angajații în ordinea descrescătoare a salariului. Angajații care au același salariu vor fi afișați în ordinea descrescătoare a comisionului.

Exercițiul 3: Să se stabilească dacă valorile null ale unei expresii "exp" sunt plasate la începutul sau la sfârșitul unei liste ordonate după valoarea "exp". Pentru a modifica modul implicit de ordonare a valorilor null se poate folosi clauza NULLS FIRST sau NULLS LAST.

Exercițiul 4: Să se afișeze numele, titlul jobului, salariul și comisionul pentru angajații care câștigă un comision mai mare de 20% din salariu. Să se ordoneze rezultatul după valoarea venitului ($\text{commission_pct} * \text{salary} + \text{salary}$).

Exercițiul 5: Să se afișeze angajații în ordinea crescătoare a vechimii.

CERERI MULTITABEL, JOIN

Join-ul este operația 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.

Condiția de join se scrie în clauza WHERE a instrucțiunii SELECT. Într-o instrucțiune SELECT care unește tabele prin operația de join, se recomandă ca numele coloanelor să fie precedate de numele sau alias-urile tabelelor pentru claritate și pentru îmbunătățirea timpului de acces la baza de date. Dacă același 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 puțin $n-1$ condiții de join.

Exemplul 6: Să se afișeze numele salariatului, codul și numele departamentului pentru toți angajații.

```
SELECT employee_id, last_name, department_name
FROM employees, departments
WHERE employees.department_id = departments.department_id;
```

```
SELECT employee_id, last_name, department_name
FROM employees e, departments d
WHERE e.department_id = d.department_id
```

Numele sau alias-urile tabelelor sunt obligatorii în dreptul coloanelor care au același nume în mai multe tabele. Altfel, nu sunt necesare dar este recomandată utilizarea lor pentru o mai bună claritate a cererii.

Exercițiul 7: Să se afișeze pentru fiecare angajat numele, codul și denumirea departamentului și titlul funcției actuale.

Exercițiul 8: Să se listeze titlurile job-urilor atribuite angajaților care lucrează în departamentul 30.

Exercițiul 9: Să se afișeze numele angajatului, numele departamentului și orașul pentru toți angajații care câștigă comision.

Exercițiul 10: Să se afișeze numele salariatului și numele departamentului pentru toți salariații care au litera A inclusă în nume.

Exercițiul 11: Să se afișeze numele și denumirea jobului actual pentru angajații care au avut funcția de 'Stock Clerk'.

Exercițiul 12: Să se afișeze pentru fiecare departament denumirea, numele șefului și titlul jobului acestuia.

FUNCTII SQL (single-row)

Funcțiile pentru formatarea datei sunt TO_DATE(data, format), TO_CHAR(data, format) unde formatul, inclus între '', poate fi alcătuit dintr-o combinație a următoarelor elemente separate de caracterele '-', '!', ':' etc. :

Element	Semnificație
D	Numărul zilei din săptămâna (duminica=1, luni=2,..., sâmbătă=6)
DD	Numărul zilei din lună.
DDD	Numărul zilei din an.
DY	Numele zilei din săptămână, printr-o abreviere de 3 litere (MON, THU etc.)
DAY	Numele zilei din săptămână, scris în întregime.
MM	Numărul lunii din an.
MON	Numele lunii din an, printr-o abreviere de 3 litere (JAN, FEB etc.)
MONTH	Numele lunii din an, scris în întregime.
Y	Ultima cifră din an
YY, YYYY, YYYY	Ultimele 2, 3, respectiv 4 cifre din an.
YEAR	Anul, scris în litere (ex: two thousand four).
HH12, HH24	Orele din zi, între 0-12, respectiv 0-24.
MI	Minutele din oră.
SS	Secundele din minut.
SSSSS	Secundele trecute de la miezul nopții.

Exemplul 13: Să se afișeze numele și data angajării pentru fiecare salariat care a fost angajat în 1987.

```
SELECT first_name, last_name, TO_CHAR(hire_date,'dd-mon-yy')
FROM employees
WHERE hire_date LIKE ('%87%');
```

```
SELECT first_name, last_name, TO_CHAR(hire_date,'dd-mon-yy')
FROM employees
WHERE TO_CHAR(hire_date,'yyyy') = '1987';
```

În exemplul anterior este necesară includerea anului între apostrof-uri?

Obs: Există două tipuri de conversii:

- implicite, realizate de sistem atunci când este necesar;
- explicite, indicate de utilizator prin intermediul funcțiilor de conversie.
-

Conversiile implicite asigurate de server-ul Oracle sunt:

- de la VARCHAR2 sau CHAR la NUMBER;
- de la VARCHAR2 sau CHAR la DATE;
- de la NUMBER la VARCHAR2 sau CHAR;
- de la DATE la VARCHAR2 sau CHAR.

Funcțiile de conversie cele mai importante sunt:

Funcție	Semnificație	Exemplu
TO_CHAR (expr_number_sau_date[, format] [,nlsparameters])	Convertește o valoare de tip numeric sau dată calendaristică, la un șir de caractere conform cu formatul specificat sau cu setările naționale specificate (NLS - National Language Support). Dacă formatul sau parametrii lipsesc se utilizează formatul și parametrii impliciți. Formatul este case sensitive.	TO_CHAR('3') = '3' TO_CHAR(-12) = '-12' TO_CHAR(sysdate, 'DDMMYYYY')= '09122009'
TO_NUMBER (expr_char[, format] [,nlsparameters])	Convertește o valoare de tip șir de caractere la o valoare numerică conform cu formatul specificat. Dacă formatul sau parametrii lipsesc se utilizează formatul și parametrii impliciți.	TO_NUMBER ('-12.22', 'S99.99') = -12.22
TO_DATE (expr_char[, format] [,nlsparameters])	Convertește o valoare de tip șir de caractere la o valoare de tip dată calendaristică în conformitate cu formatul specificat. Dacă formatul sau parametrii lipsesc se utilizează formatul și parametrii impliciți.	TO_DATE ('15-feb-2006','dd-mon-yyyy')

Exercițiul 14: Să se testeze funcțiile prezentate utilizând comenzi de tipul :

```
SELECT apel_funcție FROM dual;
```

```
SELECT TO_NUMBER('100.00', '9G999D99') from dual;
```

```
SELECT TO_NUMBER('1,300.23', '9G999D999' ) FROM DUAL;
```

```
SELECT TO_CHAR (13598.98, '099G999D999' ) FROM DUAL;
```

```
SELECT TO_CHAR (13598.98, '099G999D999' , ' NLS_NUMERIC_CHARACTERS = ",." ' )  
FROM DUAL;
```

Exercițiul 15: Să se afișeze denumirea și orașul departamentelor în care lucrează salariați care au fost angajați după 03-Jan-1990.

Exercițiul 16: Să se afișeze data curentă în formatul ”zi – lună – an ora:minut”.

Exercițiul 17: Să se afișeze numele , job-ul și data la care au început lucrul salariații angajați între 20 Februarie 1987 și 1 Mai 1989.

Observați formatul implicit al datei. Pentru a schimba acest format se poate utiliza comanda:

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY MM DD';
```

```
ALTER SESSION SET NLS_DATE_FORMAT = 'HH24:MI:SS';
```

```
SELECT sysdate FROM dual;
```

Optional

DATE, TIMESTAMP, INTERVAL

```
SELECT TO_CHAR(sysdate, 'hh:mi:ss PM') FROM dual;
```

```
SELECT SYSTIMESTAMP FROM dual; --TIMESTAMP
```

```
SELECT CAST (SYSDATE AS TIMESTAMP) FROM dual;
```

```
SELECT CURRENT_TIMESTAMP FROM dual; --TIMESTAMP WITH TIME ZONE
```

```
SELECT CAST (SYSDATE AS TIMESTAMP WITH TIME ZONE) FROM dual;
```

```
SELECT CAST (SYSDATE AS TIMESTAMP WITH LOCAL TIME ZONE) FROM dual;
```

```
SELECT TO_TIMESTAMP('03-02-2014 10:30:50.45', 'dd-mm-yyyy hh24:mi:ss.ff') from dual;
```

```
SELECT TO_TIMESTAMP_TZ('03-02-2014 10:30:50.45 03:30', 'dd-mm-yyyy hh24:mi:ss.ff  
TZh:TzM') from dual;
```

```
SELECT  
FROM_TZ(TO_TIMESTAMP('03-02-2014 10:30:50.45', 'dd-mm-yyyy hh24:mi:ss.ff'),'2:00')  
from dual;
```

```
SELECT 'x'  
FROM DUAL  
WHERE FROM_TZ(TO_TIMESTAMP('03-02-2014 10:30:50.45', 'dd-mm-yyyy  
hh24:mi:ss.ff'),'2:00') =  
TO_TIMESTAMP_TZ('03-02-2014 09:30:50.45 01:00', 'dd-mm-yyyy hh24:mi:ss.ff TZh:TzM');
```

```
SELECT sysdate + INTERVAL '3-2' YEAR TO MONTH from dual;
```

```
SELECT TO_CHAR(sysdate + INTERVAL '1' YEAR - INTERVAL '1 1:05' DAY TO MINUTE,  
'dd/mm/yyyy hh24:mi')  
FROM DUAL;
```

```
SELECT systimestamp + INTERVAL '1 1:05:30.40' DAY TO SECOND FROM DUAL;
```