

INTEROGAREA TABELELOR

Instructiunea **SELECT** realizeza regasirea datelor din tabele pe baza cererilor de interogare

SELECT [DISTINCT] (*/coloana1 [alias], expresii [alias])
FROM tabela1 [alias], tabela2 [alias], ...
WHERE (conditii, precizarea legaturilor dintre tabele)
GROUP BY nume_tabela.nume_coloana
HAVING (conditii impuse valorilor de grup)
ORDER BY nume_tabela.nume_coloana ASC/DESC;

UNDE:

SELECT	specifică atributele selectate;
DISTINCT	suprimă valorile duplicate;
*	selectează toate atributele;
coloana	selectează o anumită coloana sau mai multe coloana dintr-una sau mai multe tabele;
expresie	permite construirea de expresii avand ca rezultat valori noi
alias	denumiri pentru coloanele selectate;
FROM tabele	specifică tabelele ce conțin coloanele selectate;
WHERE	clauza permite specificarea conditiilor si a criteriilor de selectie a datelor;
GROUP BY	se precizeaza campul dupa care vor fi grupate datele in cazul expresiilor si functiilor de grup (SUM(), AVG(), COUNT(), MIN(), MAX());
HAVING	in cazul functiilor de grup conditiile impuse acestora se precizeaza in clauza HAVING;
ORDER BY	precizeaza ordonarea in functie un anumite campuri ascendent (ASC) – implicit sau descendent (DESC). Numai ORDER BY permite utilizarea aliasului;

Comenzile SQL:

- Nu sunt case sensitive, de regula comenzile se scriu cu litere mari pentru a se distinge mai usor;
- Pot fi scrise pe mai multe linii;
- Cuvintele cheie nu pot fi prescurtate sau scrise pe mai multe linii;

In construirea comenzilor SQL se utilizeaza urmasorii operatori:

<, >, =, >=, <=, NOT	operatori de comparatie
BETWEEN val_min AND val_max	între două valori (inclusiv).
SELECT id_angajat, nume, salariul, id_departament FROM salariati WHERE salariul BETWEEN 2000 AND 6100 AND id_departament NOT IN (20, 30); SELECT id_angajat, nume, salariul, id_departament FROM salariati WHERE salariul >= 2000 AND salariul <=6100 AND id_departament = 50;	
IN (listă valori)	egal cu oricare valoare din lista de valori
SELECT id_angajat, nume, id_manager FROM salariati WHERE id_manager IN (120, 122, 123);	

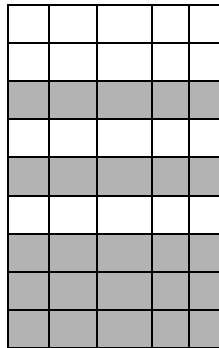
LIKE	similar cu un șablon % oricâte caractere; _ un caracter;
<pre>SELECT id_angajat, nume, id_functie FROM salariati WHERE nume LIKE 'B% ';</pre> <pre>SELECT id_angajat, nume, id_functie FROM salariati WHERE nume LIKE ' __t%';</pre>	
IS NULL	are valoarea NULL
<pre>SELECT id_angajat, nume, comision FROM salariati WHERE comision IS NULL;</pre>	

Coloanele se specifică în ordinea în care se doresc a fi afișate, nu obligatoriu în ordinea în care apar în descrierea tabelului.

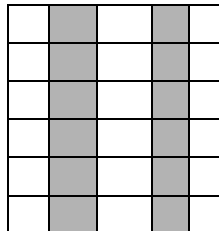
Valoarea NULL, ca și în alte limbaje, semnifică valoare ne-disponibilă, ne-alocată, și nu este același lucru cu „blank” sau „zero”.

Cu ajutorul comenzii select se pot extrage informații din baza de date. Comanda SELECT permite implementarea operatorilor algebrei relationale:

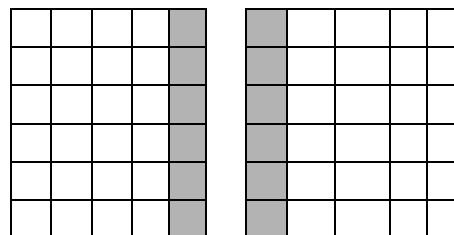
Selecția: Permite obținerea unei noi relații care conține toate atributele relației inițiale și un număr redus de tupluri. Reducerea se face după o condiție numită condiție de selecție.



Proiecția: Permite obținerea unei noi relații care conține un număr redus de atribute față de relația inițială și toate valorile sau combinațiile distincte de valori ale acestor atribute.



Joncțiunea: Operator binar, caz particular fiind produsul cartezian. Relația rezultat va conține tuplurile corespunzătoare atributelor de joncțiune care se află într-o anumită relație. Cel mai utilizat caz de joncțiune este acela în care atributele de joncțiune sunt cheie primară într-o relație și cheie externă în cealaltă relație.



Exercitii propuse:

Sa se selecteze toti clientii din tabela clienti.

`SELECT * FROM clienti;`

Sa se selecteze coloanele id_client, prenume_client nume_client si email din tabela clienti.

`SELECT id_client, prenume_client, nume_client, email_client FROM clienti;`

1. Sa se selecteze numai clientii care au limita de credit mai mare de 2500.
2. Sa se selecteze comenzile incheiate de salariatul cu id_angajat = 161.
3. Sa se selecteze toate comenzile care au fost lansate online dupa 1 ianuarie 2000.
4. Sa se afiseze salariatii al caror nume incepe cu litera s.
5. Sa se selecteze id_angajat, nume, prenume, id_manager, id_departament din tabela angajati si denumire_departament din tabela departamente si sa se realizeze jonctiunea dintre cele doua tabele.
6. Sa se selecteze comenzile incheiate de angajatul Greene numai in luna noiembrie.
7. Sa se calculeze valoarea fiecarui produs (valoare = cantitate * pret) comandat si sa se afiseze denumirea produsului, pretul, cantitatea si valoarea.
8. Sa se selecteze numai produsele comandate cu valoarea cuprinsa intre 1000 si 2000 si sa se ordoneze descrescator dupa valoare.

Realizarea interogarilor subordonate (se utilizeaza 2 comenzi SELECT imbricate)

Sa se selecteze angajatii care sunt in acelasi departament cu angajatul Smith.

`SELECT * FROM angajati
WHERE id_departament =
(SELECT id_departament FROM angajati WHERE upper(nume)= 'SMITH');`

Eroare "single-row subquery returns more than one row"

`SELECT nume, prenume, id_angajat, id_departament FROM angajati WHERE nume = 'Smith';`

```
SQL> select nume, prenume, id_angajat, id_departament FROM angajati  
2  WHERE nume='Smith';  
  
NUME          PRENUME          ID_ANGAJAT  ID_DEPARTAMENT  
-----  
Smith         Lindsey          159         80  
Smith         William          171         80  
  
2 rows selected.
```

```
SELECT * FROM angajati
WHERE id_departament IN
(SELECT id_departament FROM angajati WHERE nume= 'Smith');
```

9. Să se afișeze produsele care au prețul mai mare decât prețul procesorului „CPU D300”.

UTILIZAREA FUNCTIILOR PREDEFINITE IN INTEROGARI

- **Funcții de tip single-row (sau scalare).** O funcție single-row întoarce un singur rezultat pentru fiecare rând al tabelului interogate sau view
- Funcții de grup (sau agregate).** O funcție de grup întoarce un singur rezultat pentru un grup de rânduri interogate. Funcțiile de grup pot apărea în clauza HAVING

FUNCTII SINGLE-ROW

Funcții de tip caracter

Funcția LOWER() , UPPER(), INITCAP()

Syntax: UPPER(string1)

Example:

```
UPPER('Tech on the Net')
```

```
Result: 'TECH ON THE NET'
```

```
UPPER('george burns 123  ')
```

```
Result: 'GEORGE BURNS 123  '
```

10. Sa se afișeze cu litere mari denumirea departamentelor din locatia 1700:

11. Sa se afișeze toti angajatii cu numele Smith utilizand functiile INITCAP, UPPER, LOWER.

Operatorul de concatenare (||)

Să se afișeze denumirea produsului și pret_min.

```
SELECT 'Produsul: ' || INITCAP(denumire_produs) || ' are pretul_minim ' || pret_min
FROM produse;
```

```
' PRODUSUL: ' || INITCAP(DENUMIRE_PRODUS) || ' AREPRETUL_MINIM' || PRET_MIN
-----
Produsul: Plastic Stock - 0 are pretul_minim 1
Produsul: Plastic Stock - W/Hd are pretul_minim 2
288 rows selected.
```

Funcția CONCAT() , funcția LENGTH() , funcția SUBSTR()

Syntax: CONCAT(string1 , string2)

Syntax: LENGTH(string1)

Syntax: SUBSTR(string, start_position, [length])

Example:

SUBSTR('This is a test', 6, 2)

Result: 'is'

SUBSTR('This is a test', 6)

Result: 'is a test'

12. Să se afișeze id_client, numele clientilor concatenată cu sexul acestora și lungimea prenumelui, nivel_venituri numai pentru clientii cu venituri in categoria F: 110000 – 129999

Funcții de tip numeric

Funcția ROUND(), TRUNC()

Să se afișeze numărul 45,923 rotunjit la două zecimale si rotunjit la numar intreg. Sa se aplice si functia TRUNC.

```
SELECT ROUND(45.923,2), ROUND(45.923,0) FROM DUAL;  
SELECT TRUNC(45.923,2), TRUNC(45.923,0) FROM DUAL;
```

Funcții de tip dată calendaristică

Funcția SYSDATE

13. Să se afișeze perioada de timp corespunzătoare (în săptămâni) între data încheierii comenzii și data curentă:

Funcțiile MONTHS BETWEEN() , ADD MONTHS() , NEXT DAY() , LAST DAY()

Să se afișeze comenzile, data încheierii comenzilor, numărul de luni între data curentă și data încheierii, următoarea zi de vineri după data încheierii, ultima zi din luna din care face parte data încheierii, precum și data corespunzătoare după 2 luni de la data încheierii comenzii

```
SELECT nr_comanda, data,
```

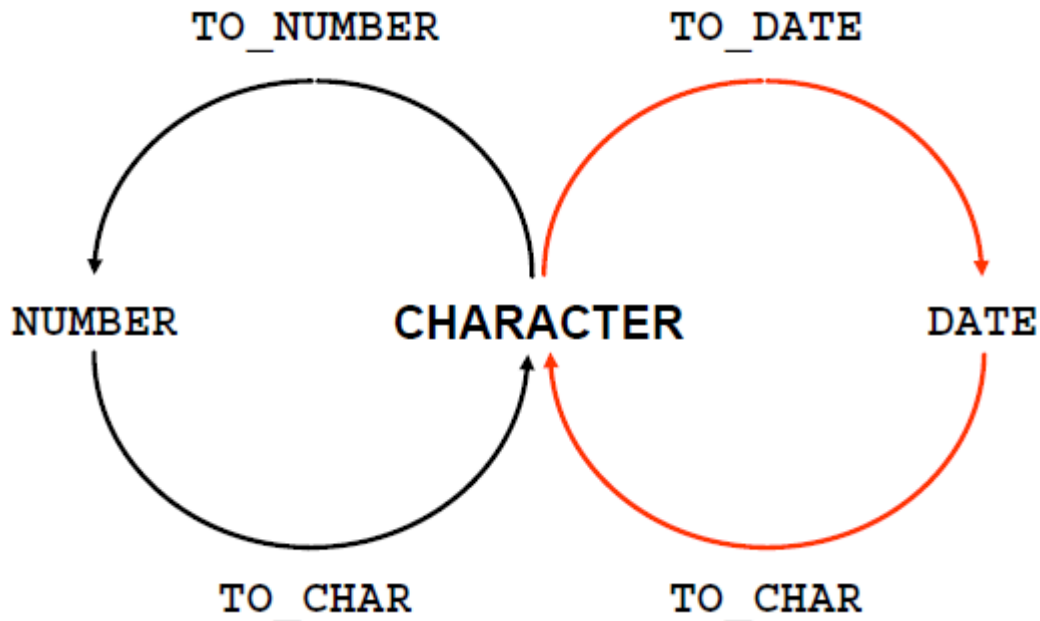
```
round(MONTHS_BETWEEN(sysdate, data)) luni,  
NEXT_DAY(data, 'FRIDAY'),  
LAST_DAY(data)  
ADD_MONTHS(data,2),  
FROM comenzi;
```

Funcția ROUND(), TRUNC()

Să se afișeze comenzile încheiate în 2000. Se va rotunji data încheierii la prima zi din luna corespunzătoare dacă data încheierii este în prima jumătate a lunii sau la prima zi din luna următoare:

```
SELECT nr_comanda, data, ROUND(data, 'MONTH') FROM comenzi  
WHERE data LIKE '%-00%';
```

Funcții de conversie



Funcția TO_CHAR

Să se afișeze comenzile și data încheierii în formatul inițial și în format “MM/YY”

```
SELECT nr_comanda, data, TO_CHAR(data, 'MM/YY') AS data_incheierii_comenzii  
FROM comenzi;
```

Funcția TO_DATE

14. Să se afișeze comenzile încheiate între 15 ianuarie și 15 decembrie 1999.

Funcția TO_NUMBER - Converteste șirul de caractere într-un număr cu un anumit format

TO_NUMBER(char[, 'format_model'])

Funcția EXTRACT()

Sa se afiseze comenzile incheiate in anii 1997 si 1998.

```
SELECT nr_comanda, data  
FROM comenzi  
WHERE EXTRACT(YEAR FROM data) IN (1997, 1998);
```

Sa se afiseze comenzile incheiate in lunile iulie si august.

```
SELECT nr_comanda, data  
FROM comenzi  
WHERE EXTRACT(MONTH FROM data) IN (7,8);
```