

Universitatea Constantin Brâncuși din Târgu Jiu

Facultatea: Inginerie

*Program de conversie profesională a cadrelor didactice din învățământul preuniversitar: Informatică,
Tehnologia Informației și a Comunicațiilor*

Baze de date

Teams: TIC - Baze de date-2022/2023

Adrian Runceanu



THE **INFORMATION** COMPANY

Curs 3

Limbajul SQL

Limbajul SQL

Cereri SELECT pe o tabelă

3.1. SELECT. Sintaxa. Efect. Rezultat

3.2. Lista SELECT

3.3. Clauza WHERE

3.4. Clauza ORDER BY

SINTAXA

SELECT [**DISTINCT**] lista_de_expresii

FROM nume_tabela

WHERE conditie_linie -- clauza optionala

ORDER BY criterii_sortare_rezultat; -- clauza
optionala

EFFECT

Se parcurg rând pe rând liniile tabelii specificate în clauza **FROM**.

Din fiecare linie conținând date pentru care condiția aflată pe clauza **WHERE** este adevărată va rezulta o linie în rezultatul cererii.

În cazul în care **WHERE** lipsește, toate liniile tabelii **FROM** vor avea o linie corespondentă în rezultatul cererii.

Linia de rezultat este compusă pe baza listei de expresii aflată pe clauza **SELECT**.

EFFECT

Daca exista cuvântul cheie **DISTINCT**, din rezultat se elimina liniile duplicat.

Înainte de a trimite rezultatul, serverul îl sorteaza în functie de criteriile specificate de clauza **ORDER BY**.

În cazul în care **ORDER BY** lipseste, liniile din rezultat sunt într-o ordine independenta de continutul lor sau de ordinea în care ele au fost adaugate în tabela.

REZULTAT

Numarul coloanelor din rezultat este egal cu numarul expresiilor din lista aflata pe clauza **SELECT**.

Aceste expresii dau si numele coloanelor din rezultat.

În lipsa clauzei **DISTINCT**, numarul de linii din rezultat este egal cu numarul liniilor din tabela care îndeplinesc conditia **WHERE** sau, când clauza respectiva lipseste, cu numarul total de linii din tabela.

REZULTAT

Evaluarea valorii de adevar a conditiei din **WHERE** se face doar pe baza datelor aflate pe linia respectiva.

Deoarece parcurgerea liniilor specificata de o cerere **SELECT** se face dupa un plan de executie generat de server, folosirea clauzei **ORDER BY** este obligatorie în cazul în care se doreste un rezultat sortat dupa anumite criterii.

Limbajul SQL

Cereri SELECT pe o tabelă

3.1. SELECT. Sintaxa. Efect. Rezultat

3.2. Lista SELECT

3.3. Clauza WHERE

3.4. Clauza ORDER BY

Exemple

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

Tabela STUDENT

MATR	NUME	AN	GRUPA	DATAN	LOC	INDRUMATOR	PUNCTAJ	CODS
1234	POPA MARCEL	1	114A	12-03-87	BUC	1001	2345	1
1235	POPESCU ION	2	121B	02-04-89	TARGU-JIU	1003	1300	1
1236	AVRAM NICOLAE	1	115A	21-03-68	TARGU-JIU	1001	3000	2
1237	IONESCU MARIANA	2	116C	05-05-89	BUC	1003	1234	3
1256	POPESCU GINA	3	114A	06-09-90	TARGU-JIU	1002	3456	2

Tabelele **SPECIALIZARE** si **BURSA**

CODS	NUME	DOMENIU
1	AUTOMATICA	CALCULATOARE
2	ENERGETICA	INGINERIE ELECTRICA
3	MECANICA	INGINERIE MECANICA

Tip	Pmin	Pmax	Suma
Fara bursa	0	399	
Bursa sociala	400	899	100
Bursa de studiu	900	1799	150
Bursa de merit	1800	2499	200
Bursa de exceptie	2500	3999	300

LISTA SELECT

Nume de coloane sau *

Exemplu 1:

```
SELECT NUME, DOMENIU  
FROM SPECIALIZARE;
```

Considerăm următoarele tabele:

Exemplu 2:

```
SELECT *  
FROM STUD;
```

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS

2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU

3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

LISTA SELECT

Constante:

Exemplu 3:

```
SELECT 'Specializarea ', NUME, ' infiintata in ', 1995  
FROM SPECIALIZARE
```

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

LISTA SELECT

Expresii aritmetice:

Exemplu 4:

```
SELECT TIP, SUMA, (SUMA+20)*1.1  
FROM BURSA;
```

Considerăm următoarele tabele:

1. **STUD** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

LISTA SELECT

Expresii concatenate:

Exemplu 5:

```
SELECT 'Specializarea ' || NUME || ' are codul ', CODS  
FROM SPECIALIZARE;
```

Considerăm următoarele tabele:

1. **STUDENT** având structura MATR, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura CODS, NUME, DOMENIU
3. **BURSA** având structura TIP, PMIN, PMAX, SUMA

Exemplu 6:

Cu valori nule:

```
SELECT TIP, ' are valoarea ' || SUMA || '.Lei'  
FROM BURSA;
```


LISTA SELECT

Alias de coloana:

Nu poate fi mai lung de 30 de caractere.

Începe cu o litera, contine numai **litere, cifre, _, #** si **\$** sau e pus între ghilimele (tot max. 30 caractere între ghilimele).

Între ghilimele literele mici sunt considerate diferite de literele mari.

Nu poate fi folosit decât în cererea curenta.

Sistemul nu stocheaza în baza de date sau altundeva aceste nume alternative.

Nu poate fi folosit în alte clauze ale cererii (doar in

□ **SELECT** si **ORDER BY**).Curs - Baze de date

LISTA SELECT

Considerăm următoarele tabele:

1. **STUDENT** având structura MATR, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura CODS, NUME, DOMENIU
3. **BURSA** având structura TIP, PMIN, PMAX, SUMA

Alias de coloana:

Exemplu 7:

```
SELECT TIP AS "Tip bursa", ' are valoarea ' || SUMA ||  
' .Lei' AS Descriere  
FROM BURSA;
```

Rezultat:

Tip bursa

FARA BURSA

BURSA SOCIALA

DESCRIERE

are valoarea .Lei

are valoarea 100.Lei

LISTA SELECT

DISTINCT: Elimina liniile duplicat din rezultat:

Exemplu 8:

```
SELECT CODS  
FROM STUD;
```

Exemplu 9:

```
SELECT DISTINCT CODS  
FROM STUD;
```

Exemplu 10:

```
SELECT DISTINCT CODS, AN  
FROM STUD;
```

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS

2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU

3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

Limbajul SQL

Cereri SELECT pe o tabelă

3.1. SELECT. Sintaxa. Efect. Rezultat

3.2. Lista SELECT

3.3. Clauza WHERE

3.4. Clauza ORDER BY

CLAUZA WHERE

Sintaxa:

WHERE expresie_logica

Exemplu 11:

SELECT NUME, GRUPA, CODS

FROM STUD

WHERE AN = 4;

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

CLAUZA WHERE

Operatori de comparare

<	Mai mic
<=	Mai mic sau egal
>	Mai mare
>=	Mai mare sau egal
<>	Diferit
!=	Diferit
^=	Diferit

CLAUZA WHERE

Conditii compuse (**AND, OR, NOT**) si paranteze

AN=2 AND PUNCTAJ>500 OR CODS=11

AN=2 AND (PUNCTAJ>500 OR CODS=11)

Considerăm următoarele tabele:

1. STUDENT având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS

2. SPECIALIZARE având structura **CODS**, NUME, DOMENIU

3. BURSA având structura **TIP**, PMIN, PMAX, SUMA

CLAUZA WHERE

Operatorul BETWEEN:

Sintaxa:

expresie **BETWEEN** valoare_minima **AND**
valoare_maxima

Exemplu 12:

```
SELECT NUME, AN, PUNCTAJ  
FROM STUD  
WHERE PUNCTAJ BETWEEN 2000 AND 3999;
```

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

CLAUZA WHERE

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

BETWEEN: Alte exemple

Exemplu 13:

```
SELECT NUME, AN, PUNCTAJ
FROM STUD
```

```
WHERE PUNCTAJ + 100 BETWEEN INDRUMATOR - 2000
AND INDRUMATOR + 1000;
```

Exemplu 14:

```
SELECT NUME, LOC, DATAN
FROM STUD
```

```
WHERE LOC BETWEEN 'A' AND 'L'
```

```
AND DATAN BETWEEN '1-JAN-89' AND '31-DEC-89';
```

CLAUZA WHERE

Operatorul IN:

Sintaxa: **expresie IN (val_1, val_2, ..., val_n)**

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

Exemplu 15:

```
SELECT NUME, AN, DATAN  
FROM STUD  
WHERE INDRUMATOR IN (1001, 1234);
```

CLAUZA WHERE

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

***IN** ignora valorile nule din lista:*

Exemplu 16 (pentru a lua in considerare valorile nule):

```
SELECT NUME, AN, GRUPA, INDRUMATOR  
FROM STUD  
WHERE INDRUMATOR IN (NULL, 1001, 1234);
```

CLAUZA WHERE

NOT IN intoarce fals daca lista contine valori nule:

Exemplu 17:

```
SELECT NUME, AN, GRUPA, INDRUMATOR  
FROM STUD  
WHERE INDRUMATOR NOT IN (NULL, 1001, 1234);
```

Considerăm următoarele tabele:

1. **STUDENT** având structura MATR, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura CODS, NUME, DOMENIU
3. **BURSA** având structura TIP, PMIN, PMAX, SUMA

CLAUZA WHERE

NOT IN intoarce fals daca lista contine valori nule:

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS

2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU

3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

Exemplu 18:

```
SELECT NUME, AN, DATAN  
FROM STUD
```

```
WHERE INDRUMATOR=1001 OR INDRUMATOR=1234;
```

CLAUZA WHERE

Considerăm următoarele tabele:

1. **STUDENT** având structura **MATR**, NUME, AN, GRUPA, DATAN, LOC, INDRUMATOR, PUNCTAJ, CODS
2. **SPECIALIZARE** având structura **CODS**, NUME, DOMENIU
3. **BURSA** având structura **TIP**, PMIN, PMAX, SUMA

Operatorul IN.

Exemplu 19:

```
SELECT NUME, PUNCTAJ, CODS
FROM STUD
```

```
WHERE PUNCTAJ + 10 IN (CODS*30+70, CODS*200+700);
```

Exemplu 20:

```
SELECT NUME, LOC, DATAN
FROM STUD
```

```
WHERE LOC IN ('BUCURESTI', 'PLOIESTI')
```

```
OR DATAN IN ('02-SEP-1995', '19-APR-1994', '29-AUG-
```

```
1994');
```

Clauza WHERE

Clauza **WHERE** poate compara valori în coloană, valori literale, expresii aritmetice sau funcții.

WHERE - restricționează cererea la rândurile care îndeplinesc o *condiție*;

Condiție este alcătuită din trei elemente:

1. Un nume de coloană
2. Un operator de comparație
3. Un nume de coloană, constantă sau listă de valori

CLAUZA WHERE

A. Operatori de comparare (operatori logici)

<	Mai mic
<=	Mai mic sau egal
>	Mai mare
>=	Mai mare sau egal
<>	Diferit
!=	Diferit
^=	Diferit

Clauza WHERE

Exemplu:

Listează toți angajații care au salariul mai mic sau egal cu 1500.

```
SELECT nume, functie  
FROM angajati  
WHERE salariu<=1500
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela **DEPARTAMENTE**
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Clauza **WHERE**

B. Șirurile de caractere și date calendaristice

Pentru a putea folosi în clauza **WHERE** șiruri de caractere și date calendaristice acestea trebuie introduse între ghilimele simple (' '), singura excepție fiind constantele numerice.

Clauza WHERE

Exemplu:

Listează toți angajații care sunt pe poziția de 'CONTABIL'.

```
SELECT *
FROM angajati
WHERE functie='CONTABIL'
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
 Den_dept varchar2(20)
 Id_manager varchar2(3)
 Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
 Id_dept number(3) referinta (FK) la tabela DEPARTAMENTE
 Nume varchar2(40)
 Prenume varchar2(40)
 Functie varchar2(25)
 Salariu number(7)
 Id_manager varchar2(3)
 Data_ang date
 Comision number(5)

Clauza WHERE

Exemplu:

Listează toți
angajații care au data
de angajare 17-DEC-
1990.

```
SELECT *
FROM angajati
WHERE data_ang='17-  
DEC-1990'
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

```
Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)
```

Tabela ANGAJATI

```
Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela DEPARTAMENTE
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)
```

Clauza WHERE

Exemplu:

Listeaza toti angajatii
care sunt in
departamentul 10.

```
SELECT nume, functie,  
        salariu  
FROM angajati  
WHERE id_dept=10
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

```
Id_dept number(3) cheie primara (PK)  
Den_dept varchar2(20)  
Id_manager varchar2(3)  
Locatie varchar2(100)
```

Tabela ANGAJATI

```
Id_angajat number(3) cheie primara (PK)  
Id_dept number(3) referinta (FK) la tabela DEPARTAMENTE  
Nume varchar2(40)  
Prenume varchar2(40)  
Functie varchar2(25)  
Salariu number(7)  
Id_manager varchar2(3)  
Data_ang date  
Comision number(5)
```

Clauza WHERE

Atenție!

Contează dacă caracterele sunt scrise cu litere mari sau cu litere mici.

Datele calendaristice sunt înregistrate în baza de date într-un format numeric intern:

secol, an, luna, ziua, ora, minute, secunde

Formatul de afișare este: **DD-MON-RR**.

Acesta poate fi schimbat (vedeți în alt curs).

Clauza WHERE

C. Condiții de comparare

În **SQL** sunt 4 operatori care pot fi folosiți pentru toate tipurile de date:

Operator	Semnificație
BETWEEN...AND...	între 2 valori (inclusiv)
IN (lista)	compară cu o listă de valori
LIKE	compară cu un model de tip caracter
IS NULL	este o valoare nulă

Clauza WHERE

1. BETWEEN...AND...

Condiția **BETWEEN** poate fi folosită pentru a selecta rânduri pe baza unui interval de valori (conținut în condiție).

Intervalul este inclusiv, are o limită inferioară și o limită superioară și neapărat prima specificată trebuie să fie limita inferioară.

Clauza WHERE

Exemplu:

Listează toți angajații
care au salariul între 1000 și
2000.

```
SELECT *  
FROM angajati  
WHERE salariu BETWEEN  
1000 AND 2000
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela **DEPARTAMENTE**
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Clauza WHERE

2. IN

Condiția **IN** testează valorile dintr-o listă specificată.

Pot fi folosite orice tipuri de date, cu precizarea că pentru șirurile de caractere și date calendaristice trebuie folosite ghilimelele simple (' ').

Clauza WHERE

Exemplu:

Listează toți angajații
care au salariul în lista
(3631, 1432, 4000, 5000).

```
SELECT nume, functie, salariu  
FROM angajati  
WHERE salariu IN (3631,  
1432, 4000, 5000)
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela **DEPARTAMENTE**
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Clauza WHERE

Exemplu:

Listează toți angajații care au numele în lista ('POPA', 'IONESCU', 'POPESCU').

```
SELECT *  
FROM angajati  
WHERE nume IN ('POPA',  
                  'IONESCU', 'POPESCU')
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela DEPARTAMENTE
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Clauza WHERE

3. LIKE

Dacă nu se cunoaște valoarea exactă căutată, cu ajutorul condiției **LIKE** putem să selectăm rândurile care se potrivesc cu un model specificat de caractere.

Operația de căutare după un model poate fi asemănată cu o căutare "**wildcard**".

Pentru construirea modelului șirului căutat pot fi folosite 2 simboluri:

- %** : orice secvență de 0 sau mai multe caractere
- _** : un singur caracter (oarecare)

Aceste simboluri pot fi folosite în orice combinație de caractere literale.

Clauza WHERE

Exemplu:

Listează toți angajații
al căror nume începe cu
litera S.

```
SELECT *  
FROM angajati  
WHERE nume LIKE 'S%'
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela **DEPARTAMENTE**
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Clauza WHERE

Exemplu:

Listează toți angajații care au numele de 4 caractere.

```
SELECT nume
FROM angajati
WHERE nume LIKE '____'
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
 Den_dept varchar2(20)
 Id_manager varchar2(3)
 Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
 Id_dept number(3) referinta (FK) la tabela DEPARTAMENTE
 Nume varchar2(40)
 Prenume varchar2(40)
 Functie varchar2(25)
 Salariu number(7)
 Id_manager varchar2(3)
 Data_ang date
 Comision number(5)

Clauza WHERE

Exemplu

Listeaza toți angajații care al doilea caracter din nume 'o'.

```
SELECT nume, functie,
      data_ang
FROM angajati
WHERE nume LIKE '_o%'
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

```
Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)
```

Tabela ANGAJATI

```
Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela DEPARTAMENTE
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)
```


Clauza WHERE

ESCAPE - când sunt căutate chiar caracterele % sau _. Acesta specifică caracterul care este "sărit".

Sintaxa:

SELECT coloana

FROM tabela

WHERE coloana **LIKE** '%sa_%' **ESCAPE** '\\';

ESCAPE identifică '\\ ' drept caracterul care trebuie "sărit".

El precede caracterul '_' astfel încât acesta va fi considerat drept literal.

Clauza WHERE

4. Conditia IS NULL

Pentru a verifica valorile de tip **NULL** există condiția **IS NULL** sau negarea acesteia **IS NOT NULL**.

O valoare nulă este o valoare care este sau incorectă, sau necunoscută, sau inaplicabilă de aceea nu poate fi testată cu "=".

O valoare nulă nu este la fel cu "zero" care este un număr.

Dacă valoarea **NULL** este utilizată într-o comparație, atunci operatorul de comparație trebuie să fie **IS NULL** sau **IS NOT NULL** altfel rezultatul este întotdeauna FALSE.

Clauza WHERE

Exemplu:

Listează toți angajații
care nu au comision.

```
SELECT *  
FROM angajati  
WHERE comision IS NULL
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela **DEPARTAMENTE**
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Clauza WHERE

5.Negarea expresiilor

Operator	Semnificatie
!=	diferit de (pt anumite S.O.)
^=	diferit de (pt anumite S.O.)
<>	diferit de
NOT BETWEEN	nu se afla intre 2 valori date
NOT IN	nu se afla intr-o lista
NOT LIKE	diferit de sirul
IS NOT NULL	nu este o valoare nula

Clauza WHERE

D. Prioritatea de execuție

Ordinea de execuție	Operator
1.	Operatorii aritmetici
2.	Operatorii de concatenare
3.	Condițiile de comparare
4.	IS [NOT] NULL, LIKE, [NOT] IN
5.	[NOT] BETWEEN
6.	Operatorul logic NOT
7.	Operatorul logic AND

Clauza WHERE

Exemplu:

```
SELECT *  
FROM angajati  
WHERE functie = 'DIRECTOR'  
OR functie = 'CONTABIL'  
AND salariu < 1500
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela DEPARTAMENTE
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Clauza WHERE

Dar ordinea se poate modifica dacă se folosesc paranteze.

```
SELECT *  
FROM angajati  
WHERE (functie = 'DIRECTOR'  
OR functie = 'CONTABIL')  
AND salariu < 1500
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela DEPARTAMENTE
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Limbajul SQL

Cereri SELECT pe o tabelă

3.1. SELECT. Sintaxa. Efect. Rezultat

3.2. Lista SELECT

3.3. Clauza WHERE

3.4. Clauza ORDER BY

Clauza ORDER BY

În mod normal (fără clauza **ORDER BY**) rândurile sunt returnate într-o ordine convenită de Oracle însă ea fiind consistentă de la cerere la cerere.

Cu ajutorul clauzei **ORDER BY** rândurile vor fi afișate în ordinea solicitată (cu toate acestea nu se va modifica ordinea internă a rândurilor din baza de date).

Clauza **ORDER BY** trebuie să fie ultima clauză din cerere (interogare).

Clauza ORDER BY

Sintaxa:

SELECT coloana

FROM tabela

[**WHERE** conditie]

[**ORDER BY** {coloana, expresie}

[ASC/DESC]];

unde:

ORDER BY - Specifică ordinea în care sunt ordonate rândurile

ASC - Ordonează rândurile crescător

DESC - Ordonează rândurile descrescător

Clauza ORDER BY

Exemplu:

Listați toți angajații ordonându-i după data de angajare (în ordine crescătoare).

```
SELECT *  
FROM angajati  
ORDER BY data_ang
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
Den_dept varchar2(20)
Id_manager varchar2(3)
Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
Id_dept number(3) referinta (FK) la tabela **DEPARTAMENTE**
Nume varchar2(40)
Prenume varchar2(40)
Functie varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Clauza ORDER BY

Exemplu:

Listați toți angajații ordonându-i după data de angajare (în ordine descrescătoare).

```
SELECT *
FROM angajati
ORDER BY data_ang DESC
```

Sa se creeze urmatoarele tabele:

Tabela DEPARTAMENTE

Id_dept number(3) cheie primara (PK)
 Den_dept varchar2(20)
 Id_manager varchar2(3)
 Locatie varchar2(100)

Tabela ANGAJATI

Id_angajat number(3) cheie primara (PK)
 Id_dept number(3) referinta (FK) la tabela **DEPARTAMENTE**
 Nume varchar2(40)
 Prenume varchar2(40)
 Functie varchar2(25)
 Salariu number(7)
 Id_manager varchar2(3)
 Data_ang date
 Comision number(5)

Clauza ORDER BY

Mai putem ordona randurile returnate cu ajutorul alias-urilor.

Exemplu:

Listati toti angajatii ordonandu-i dupa salariul anual: sal*12 salannual (in ordine crescatoare).

```
SELECT nume, functie, salariu*12 salariuanual  
FROM angajati  
ORDER BY salariuanual
```

Clauza ORDER BY

Rândurile mai pot fi ordonate și după mai multe coloane.

Numărul de coloane după care se pot ordona rândurile întoarse este numărul maxim de coloane existente în tabelă.

Exemplu

Listați toți angajații ordonându-i după data de angajare, salariu și comision (în ordine crescătoare).

```
SELECT *  
FROM angajati  
ORDER BY data_ang, salariu, comision
```

Clauza ORDER BY

În clauza **ORDER BY** se pot folosi coloane care nu sunt conținute în **SELECT**.

Exemplu:

Listați toți angajații ordonându-i după salariu (în ordine crescătoare).

```
SELECT nume, functie  
FROM angajati  
ORDER BY salariu
```

Exercitii

1. Listați toți angajații care au salariul între 1500 și 3000.
2. Listați numele angajaților ordonându-i în ordine alfabetică.
3. Listați toți angajații ai căror nume conțin NE sau LL în interior.
4. Listați toți angajații care nu au manager.

Exercitii

5. Listați toți angajații care au fost angajați în anul 2010.
6. Afișați numele, salariul anual și comisionul pentru toți vânzătorii ai căror salariu lunar este mai mare decât comisionul lor.

leșirea va fi ordonată după salariu, cele mai mari primele.

Dacă doi sau mai mulți angajați au același salariu trebuie sortați după nume în ordinea celor mai mari salarii.

Întrebări?