

BAZE DE DATE - 3

Autor

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Carti

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbnb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

Limbajul SQL

Cereri SELECT pe o tabela

1. **Funcții**
2. Funcții referitoare la o singură înregistrare
3. Funcții referitoare la mai multe înregistrări
 1. Clauza **GROUP BY**
 2. Excluderea grupurilor (clauza **HAVING**)
 3. Imbricarea funcțiilor de grup

Tabele Angajati si Departamente

Pentru exemplele din cursuri vom folosi tabelele *Angajati* si *Departamente*.

DEPARTAMENTE

Id_dept NUMBER(3) CHEIE PRIMARA (PK)

Den_dept VARCHAR2(20)

Id_manager VARCHAR2(3)

Locatie VARCHAR2(100)

Tabele Angajati si Departamente

ANGAJATI

Id_angajat **NUMBER(3)** CHEIE PRIMARA (**PK**)

Id_dept **NUMBER(3)** REFERINTA (**FK**) LA TABELA

DEPARTAMENTE

Nume **VARCHAR2(40)**

Prenume **VARCHAR2(40)**

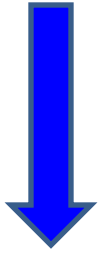
Functie **VARCHAR2(25)**

Salariu **NUMBER(7)**

Id_manager **VARCHAR2(3)**

Data_ang **DATE**

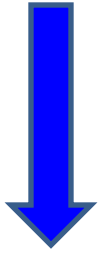
Comision **NUMBER(5)**



Funcții

Funcțiile sunt o caracteristică importantă a SQL și sunt utilizate pentru:

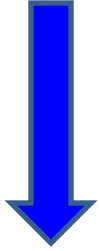
- 1. a realiza calcule asupra datelor**
- 2. a modifica date**
- 3. a manipula grupuri de înregistrări**
- 4. a schimba formatul datelor**
- 5. sau pentru a converti diferite tipuri de date**



Funcții

Funcțiile se clasifică în două tipuri:

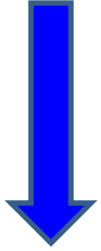
1. **Funcții referitoare la o singură înregistrare (single-row functions)**
2. **Funcții referitoare la mai multe înregistrări (multiple-row functions)**



Funcții

1. Funcții referitoare la o singură înregistrare (single-row functions):

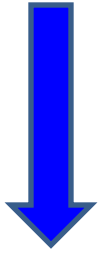
1. funcții caracter
2. funcții numerice
3. funcții pentru data calendaristică și oră
4. funcții de conversie
5. funcții diverse



Funcții

2. Funcții referitoare la mai multe înregistrări (multiple-row functions):

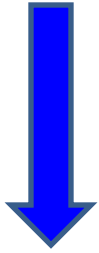
- funcții totalizatoare sau funcții de grup



Funcții

Diferența dintre cele două tipuri de funcții este numărul de înregistrări pe care acționează:

- *Funcțiile referitoare la o singură înregistrare returnează un singur rezultat pentru fiecare rând al tablei,*
- *pe când funcțiile referitoare la mai multe înregistrări returnează un singur rezultat pentru fiecare grup de înregistrări din tabela.*

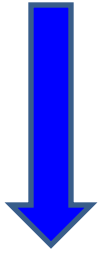


Funcții

O observație importantă este faptul că dacă se apelează o funcție **SQL** ce are un argument (parametru) egal cu valoarea **Null**, atunci în mod automat rezultatul va avea valoarea **Null**.

Singurele funcții care nu respectă această regulă sunt:

- **CONCAT**
- **DECODE**
- **DUMP**
- **NVL**
- **REPLACE**

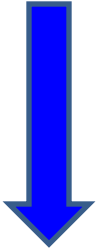


Limbajul SQL

Cereri SELECT pe o tabela

1. Funcții
2. Funcții referitoare la o singură înregistrare
3. Funcții referitoare la mai multe înregistrări
 - Clauza **GROUP BY**
 - Excluderea grupurilor (clauza **HAVING**)
 - Imbricarea funcțiilor de grup

Funcții referitoare la o singură înregistrare



- *Sunt funcții utilizate pentru manipularea datelor individuale.*
- Ele pot avea unul sau mai multe argumente și returnează o valoare pentru fiecare rând rezultat în urma interogării.

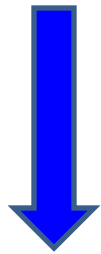
↓ Funcții referitoare la o singură înregistrare

Sunt mai multe tipuri de funcții pe un singur rând.

Sintaxa:

function_name [(arg1,arg2,...)]

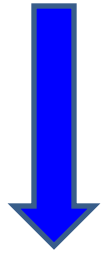
În sintaxa generală ***function_name*** este numele funcției și ***arg1,arg2*** sunt argumentele funcției care pot fi date de **numele unei coloane** sau de o expresie



Funcții referitoare la o singură înregistrare

Funcțiile pe un singur rând cuprind următoarele tipuri de funcții:

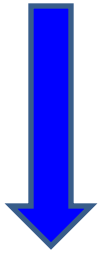
1. funcții de tip caracter
2. funcții de tip numeric
3. funcții de tip data
4. funcții de conversie
5. funcții generale: **NVL, NVL2, NULLIF, COALESCE, CASE, DECODE**



Funcții referitoare la o singură înregistrare

1. Funcții de tip caracter

Aceste funcții au ca argumente date de tip caracter și returnează date de tip **VARCHAR2**, **CHAR** sau **NUMBER**.



Funcții referitoare la o singură înregistrare

Cele mai importante funcții caracter sunt:

Funcție	Descriere
LOWER(column expression)	converteste alfa caracterele din caractere mari in caractere mici
UPPER(column expression)	converteste alfa caracterele din caractere mici in caractere mari
INITCAP(column expression)	converteste prima litera a fiecarui cuvânt in caractere mari si restul cuvântului in caractere mici
CONCAT(column1 expression1, column2 expression2)	funcția este echivalentul operatorului de concatenare ()
SUBSTR(column expression, m [, n])	returnează un șir de <i>n</i> caractere începând cu caracterul aflat pe poziția <i>m</i>
LENGTH(column expression)	returnează numărul de caractere dintr-o expresie
INSTR(column expression, 'string', [m], [n])	returnează poziția unui anumit șir, opțional se poate începe căutarea cu poziția <i>m</i> sau cu a <i>n</i> -a apariție a șirului. <i>m</i> și <i>n</i> sunt prin definiție 1
REPLACE(text, search_string, replacement_string)	caută un anumit text într-un șir de caractere și dacă îl găsește îl înlocuiește

Funcții referitoare la o singură înregistrare

Exemplu de utilizare a funcției **LENGTH**:

```
SELECT LENGTH(ume)  
FROM angajati;
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

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)

id_autor	nume
1	P
2	D
3	V
4	L
5	I
6	G
7	C
8	N
9	V

Funcții referitoare la o singură înregistrare

Exemplu:

```
SELECT 'Numele functiei pentru  
|| UPPER(nume) ||'este '|| LOWER(funcție) AS  
"DETALII ANGAJAT"  
FROM angajati;
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbbb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

id_autor	nume
1	P
2	D
3	V
4	L
5	I
6	G
7	C
8	N
9	V

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)
Funcție varchar2(25)
Salariu number(7)
Id_manager varchar2(3)
Data_ang date
Comision number(5)

Funcții referitoare la o singură înregistrare

Exemplu:

```
SELECT id_angajat, UPPER(nume), functie, id_dept  
FROM angajati  
WHERE nume = 'popa'
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbnb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

id_autor
1
2
3
4
5
6
7
8
9

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)

Funcții referitoare la o singură înregistrare

Clauza **WHERE** a acestei cereri **SQL** compară numele din tabela Angajați cu 'Popa'.

Pentru comparație numele sunt convertite în litere mici și din această cauză se obține un rezultat.

Exemplu:

```
SELECT id_ang, UPPER(nume), functie, id_dept  
FROM angajati  
WHERE INITCAP(nume) = 'Popa'
```

idcarte	autor	titlu	pret	cantitatea
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950.00	23
2-2222-222-13	Vieru	MAMA	2000.00	200
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3

Funcții referitoare la o singură înregistrare

Exemplu:

Pentru afișarea numelui cu majuscule de folosește funcția **UPPER**.

```
SELECT id_ang, CONCAT(nume, functie), nume,  
UPPER(nume)  
FROM angajati;
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

id_autor	nume
1	Petru
2	Dium
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Cole
8	Nicu
9	Valea

Funcții referitoare la o singură înregistrare

Spre deosebire de alte funcții, funcțiile caracter pot fi imbricate până la orice adâncime.

Dacă funcțiile sunt imbricate, atunci ele sunt ***evaluate din interior spre exterior.***

Pentru a determina, de exemplu, de câte ori apare caracterul 'A' în câmpul nume vom folosi interogarea:

Funcții referitoare la o singură înregistrare

```
SELECT nume, LENGTH (nume) - LENGTH (TRANSLATE(nume,'DA','D'))  
FROM angajati;
```

Sa se creeze urmatoarele tabele:

NUME	'A'
GHEORGHIU	0
MARIN	1
GEORGESCU	0
IONESCU	0
ALBU	1
VOINEA	1
STANESCU	1

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

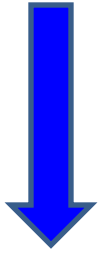
Comision number(5)

Funcții referitoare la o singură înregistrare

Notă:

În exemplul anterior, funcția **TRANSLATE** (nume, 'DA', 'D') va căuta în coloana “nume” primul caracter (caracterul 'D') din cel de-al doilea argument al funcției (șirul de caractere 'DA') și îl va înlocui cu primul caracter (adică tot cu caracterul 'D') din cel de-al treilea argument al funcției (șirul de caractere 'D'), apoi va căuta cel de-al doilea caracter, adică caracterul 'A', și îl va șterge din câmpul nume deoarece acesta nu are caracter corespondent în cel de-al treilea argument al funcției.

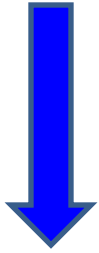
Am folosit acest artificiu deoarece șirul de caractere vid este echivalent cu valoarea **Null**, deci funcția **TRANSLATE** (nume, 'A', ' ') ar fi înlocuit toate valorile câmpului “nume” cu valoarea **Null**.



Limbajul SQL

Cereri SELECT pe o tabela

1. Funcții
2. Funcții referitoare la o singură înregistrare
3. Funcții referitoare la mai multe înregistrări
 1. Clauza **GROUP BY**
 2. Excluderea grupurilor (clauza **HAVING**)
 3. Imbricarea funcțiilor de grup



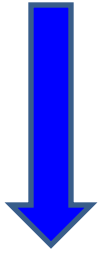
Funcții de grup

Funcțiile de grup sunt funcții care operează pe un set de rânduri pentru a da un rezultat pe întreg setul.

Parametrii și descrierea funcțiilor de grup.

Funcțiile de grup sunt:

1. **AVG**
2. **COUNT**
3. **MAX**
4. **MIN**
5. **STDDEV**
6. **SUM**
7. **VARIANCE**



Funcții de grup

Fiecare dintre aceste funcții acceptă anumiți parametri:

Funcția	Descriere
AVG([DISTINCT ALL]n)	Valoarea medie pentru grup , ignorand valorile nule
COUNT({* [DISTINCT ALL]expr})	Numarul de randuri unde expr evalueaza altceva in afara de null (folosind * sunt numarate toate randurile, incluzand duplicatele si pe cele cu valoare nula)
MAX([DISTINCT ALL]expr)	Valoarea maxima a expr , ignorand valorile nule
MIN([DISTINCT ALL]expr)	Valoarea minima a expr , ignorand valorile nule
STDDEV([DISTINCT ALL]x)	Deviatia standard pentru grup, ignorand valorile nule
SUM([DISTINCT ALL]x)	Suma valorilor pentru grup, ignorand valorile nule
VARIANCE([DISTINCT ALL]x)	Variatia pentru grup, ignorand valorile nule



Funcții de grup

DISTINCT face ca funcția să ignore valorile duplicat.


ALL face ca funcția să afișeze și valorile duplicat.

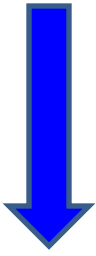
Valoarea implicită este **ALL**, deci nu este necesar să fie specificată.

Tipul de dată returnat de funcția *expr* poate fi **CHAR**, **VARCHAR2**, **NUMBER** sau **DATE**.

Toate funcțiile de grup ignoră valorile nule.

Pentru a lua în considerare și valorile nule se folosesc funcțiile **NVL**, **NVL2** sau **COALESCE**.





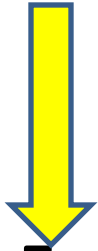
Funcții de grup

Sintaxa funcțiilor de grup:

```
SELECT [coloana,] functie_de_grup(coloana),  
...  
FROM tabel  
[WHERE conditie]  
[GROUP BY coloana]  
[HAVING conditie_de_grupare]  
[ORDER BY coloana];
```

*Rezultatele sunt sortate **implicit crescător**.* Pentru o ordonare descrescătoare se va folosi clauza **DESC** după **ORDER BY**.

Funcții de grup



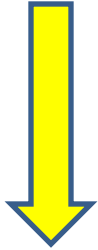
Exemplul 1:

id_autor	nume_a
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Afișarea salariului mediu, maxim, minim și suma tuturor salariilor angajaților cu funcție VANZATOR.

```
SELECT AVG(salariu), MAX(salariu), MIN(salariu),  
       SUM(salariu)  
FROM angajati  
WHERE functie='VANZATOR';
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2



Exemplul 2

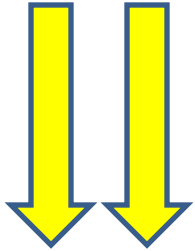
Funcții de grup

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Datele la care s-au făcut prima și ultima angajare.

```
SELECT MIN(data_ang), MAX(data_ang)  
FROM angajati;
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2



Exemplul 3

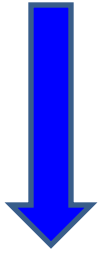
Funcții de grup

id_autor	nume_a
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Primul și ultimul nume de angajat în ordine alfabetică.

```
SELECT MIN(nume), MAX(nume)
FROM angajati;
```

idcarte	autor	titlu	pret	cantitatea
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbbb	950.00	23
2-2222-222-13	Vieru	MAMA	2000.00	200
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3



Funcții de grup

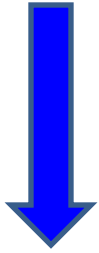
Funcția **COUNT**

Funcția **COUNT** are 3 formate:

COUNT(*)

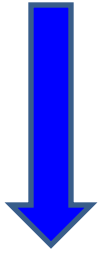
COUNT(expr)

COUNT(DISTINCT expr)



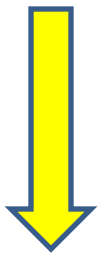
Funcții de grup

- **COUNT(*)** *întoarce numărul de rânduri dintr-o tabela care satisface criteriul de selecție, incluzând rândurile duplicat și rândurile conținând valori nule.*
- Dacă clauza **WHERE** este introdusă, atunci **COUNT(*)** returnează numărul de rânduri care satisfac condiția din clauza **WHERE**.



Funcții de grup

- În contrast, funcția **COUNT(expr)** întoarce **numărul de valori nenule** din coloana specificată de **expr**.
- **COUNT(DISTINCT expr)** returnează **numărul de valori distincte, nenule** din coloana specificată de **expr**.



Funcții de grup

Exemplul 4

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Numărul angajaților din departamentul cu id-ul 30.

```
SELECT COUNT(*)  
FROM angajati  
WHERE id_dept = 30;
```

titlu	pret	cantitatea	id_autor
GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
MAMA	2000.00	200	3
cu Ghid Php	1000.00	3	2
cu Ghid Php	1000.00	3	2
Cei trei muschetari Php	930.00	40	4
cu Ghid Php	1000.00	3	6
cu Ghid Php	1000.00	3	2

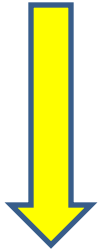
Sa se creeze urmatoarele tabele:

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)



Funcții de grup

Exemplul 5

Numărul angajaților care iau comision din departamentul 30.

```
SELECT COUNT(comision)
FROM angajati
WHERE id_dept = 30;
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-0	Eminescu	Ghid Php	1000.00	3	2

id_autor	nume_au
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

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)

1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Funcții de grup

Exemplul 6

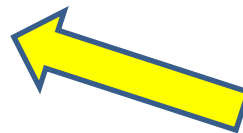
Numărul de departamente din firma
(**varianta incorectă** și **varianta corectă**).

idcarte	autor	titlu
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb
2-2222-222-13	Vieru	MAMA
2-2222-2222-22	Eminescu	Ghid Php
2-2222-2222-23	Eminescu	Ghid Php
2-2222-2222-6	Vica	Cei trei muschetari Php
2-2222-2222-8	Eminescu	Ghid Php
2-2222-2222-9	Eminescu	Ghid Php

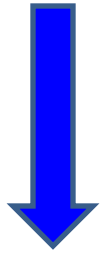
SELECT COUNT(id_dept), COUNT(DISTINCT id_dept)

FROM angajati;

**SELECT COUNT(id_autor), COUNT(DISTINCT id_autor)
FROM carti;**



id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea



Funcții de grup

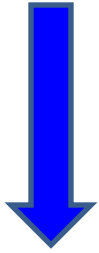
Exemplul 7

Comisionul mediu în departamentul 30 (ignorând sau nu valorile nule).

```
SELECT AVG(comision), AVG(NVL(comision, 0))
FROM angajati
WHERE id_dept = 30;
```



idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

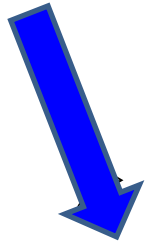


Limbajul SQL

Cereri SELECT pe o tabela

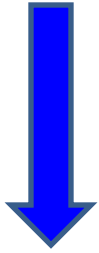
1. Funcții
2. Funcții referitoare la o singură înregistrare
3. Funcții referitoare la mai multe înregistrări
 1. Clauza **GROUP BY**
 2. Excluderea grupurilor (clauza **HAVING**)
 3. Imbricarea funcțiilor de grup

Clauza GROUP BY




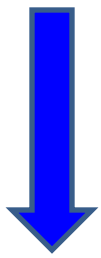
Până acum toate funcțiile de grup au fost aplicate *întregii tabele*.

- Pentru **a putea împărți tabela în grupuri** mai mici se folosește clauza **GROUP BY**.
- Folosirea acesteia returnează *informații sumare despre fiecare grup*.



Clauza GROUP BY

- Folosind **GROUP BY** *nu se pot extrage și coloane individuale*, ci doar *coloane ce rămân identice în tot grupul*.
- Folosind **WHERE** *se pot exclude rânduri, înaintea împărțirii lor în grupuri*. 
- *Nu pot fi folosite* aliasuri de coloane în clauza **GROUP BY**.
- Implicit, rândurile sunt sortate crescător după coloana (coloanele) specificate în **GROUP BY**.
- Acest lucru poate fi schimbat folosind **ORDER BY**.



Clauza GROUP BY

id_autor	nume_au
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Exemplul 8

Salariul mediu pe fiecare departament.



```

SELECT id_dept, AVG(salariu)
FROM angajati
GROUP BY id_dept;

```

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)

```

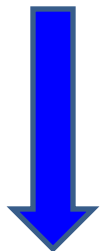
Tabela ANGAJATI

```

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)

```

titlu	pret	cantitatea	id_autor
GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
MAMA	2000.00	200	3
cu Ghid Php	1000.00	3	2
cu Ghid Php	1000.00	3	2
Cei trei muschetari Php	930.00	40	4
cu Ghid Php	1000.00	3	6
cu Ghid Php	1000.00	3	2



Clauza GROUP BY

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

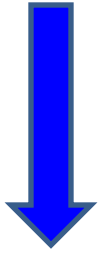
Exemplul 9

Salariul mediu pe fiecare departament, iar rezultatele ordonate după salariul mediu pe departament.



```
SELECT id_dept, AVG(salariu)
FROM angajati
GROUP BY id_dept
ORDER BY AVG(salariu);
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

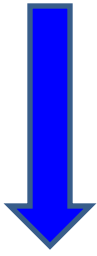


Clauza GROUP BY

Gruparea după mai multe coloane.

Câteodată este necesară obținerea de rezultate pentru grupuri în alte grupuri.

Atunci în dreptul clauzei **GROUP BY** vom întâlni mai multe coloane.

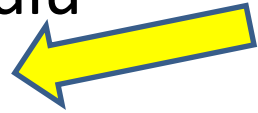


Clauza GROUP BY

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

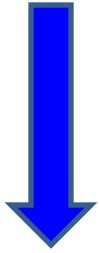
Exemplul 10

Salariul total pe fiecare departament si pe fiecare functie, iar rezultatele ordonate după salariul mediu pe departament.



```
SELECT id_dept, functie, SUM(salariu)
FROM angajati
GROUP BY id_dept, functie
ORDER BY AVG(salariu);
```

idcarte	autor	titlu	pre
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.0
2-2222-222-13	Vieru	MAMA	2000.0
2-2222-2222-22	Eminescu	Ghid Php	1000.0
2-2222-2222-23	Eminescu	Ghid Php	1000.0
2-2222-2222-6	Vica	Cei trei muschetari Php	930.0
2-2222-2222-8	Eminescu	Ghid Php	1000.0
2-2222-2222-9	Eminescu	Ghid Php	1000.0



Limbajul SQL

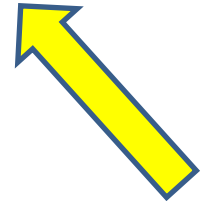
Cereri SELECT pe o tabela

1. Funcții
2. Funcții referitoare la o singură înregistrare
3. Funcții referitoare la mai multe înregistrări
 1. Clauza **GROUP BY**
 2. Excluderea grupurilor (clauza **HAVING**)
 3. Imbricarea funcțiilor de grup



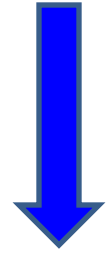
Excluderea grupurilor (clauza **HAVING**)

Clauza **HAVING** funcționează în mare ca și clauza **WHERE**, diferența fiind că **HAVING** *este folosit pentru a exclude anumite grupuri din rezultat*, nu rânduri cum făcea **WHERE**.



Clauza **HAVING** poate fi folosită înainte de **GROUP BY**, însă este mai logic să fie folosită după.

Ordinea execuției va rămâne aceeași.



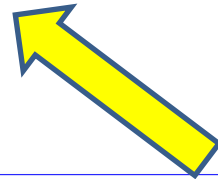
Excluderea grupurilor (clauza HAVING)

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Exemplul 11

Salariul mediu pe fiecare departament unde acesta depășește 3000\$.

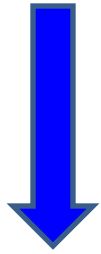
```
SELECT id_dept, AVG(salariu)
FROM angajati
HAVING AVG(salariu) > 3000
GROUP BY id_dept;
```



idcarte	autor	titlu	pret	cantitatea	id_autor		
2-2222-222-10	Petru	GFGF	Laborator	Mysql-Php	vbvbbb	950.00	23
2-2222-222-13	Vieru	MAMA				2000.00	200
2-2222-2222-22	Eminescu	Ghid	Php			1000.00	3
2-2222-2222-23	Eminescu	Ghid	Php			1000.00	3
2-2222-2222-6	Vica	Cei trei muschetari	Php			930.00	40
2-2222-2222-8	Eminescu	Ghid	Php			1000.00	3
2-2222-2222-9	Eminescu	Ghid	Php			1000.00	3

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Excluderea grupurilor (clauza HAVING)



Exemplul 12



Salariul maxim pe fiecare departament unde acesta depășește 3000\$.

```
SELECT id_dept, MAX(salariu)
FROM angajati
HAVING MAX(salariu) > 3000
GROUP BY id_dept;
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

Excluderea grupurilor (clauza **HAVING**)

idcarte	autor	titlu
2-2222-222-10	Petru GFGF	Laborator Mysq-
2-2222-222-13	Vieru	MAMA
2-2222-2222-22	Eminescu	Ghid Php
2-2222-2222-23	Eminescu	Ghid Php
2-2222-2222-6	Vica	Cei trei muscheta
2-2222-2222-8	Eminescu	Ghid Php
2-2222-2222-9	Eminescu	Ghid Php

Exemplul 13

Salariul total pe fiecare funcție, fără a lua în calcul MANAGERII, excluzând funcțiile cu suma salariilor sub 6000\$ cu ordonare după total.

SELECT functie, **SUM**(salariu)

FROM angajati

WHERE functie!='MANAGER'

GROUP BY functie

HAVING **SUM**(salariu) < 6000

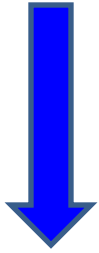
ORDER BY **SUM**(salariu);

← Salariul total pe fiecare funcție

← fără a lua în calcul **MANAGERII**

← excluzând funcțiile cu suma salariilor sub 6000\$

← cu ordonare după total

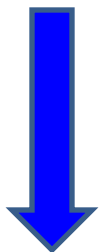


Limbajul SQL

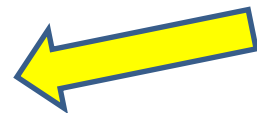
Cereri SELECT pe o tabela

1. Funcții
2. Funcții referitoare la o singură înregistrare
3. Funcții referitoare la mai multe înregistrări
 1. Clauza **GROUP BY**
 2. Excluderea grupurilor (clauza **HAVING**)
 3. **Imbricarea funcțiilor de grup**

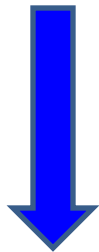
Ordinea de executie a functiilor de grup





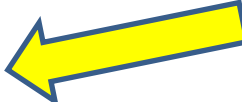
Serverul **Oracle/MYSQL** execută funcțiile de grup într-o anumită ordine:

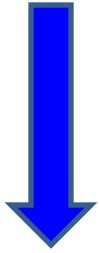


1. Selecția rândurilor ce respectă clauza **WHERE**
2. Gruparea rândurilor obținute, respectând clauza **GROUP BY**
3. Calcularea rezultatelor funcțiilor de grup pentru fiecare grup în parte
4. Eliminarea grupurilor ce nu respectă clauza **HAVING**
5. Ordonarea rezultatelor respectând clauza **GROUP BY**.

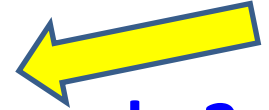


Ordinea de executie a functiilor de grup

- *Ordinea de execuție are o importanță foarte mare, deoarece are un impact direct asupra vitezei.* 
- *Cu cât mai multe înregistrări pot fi eliminate utilizând clauza **WHERE**, cu atât mai puțin va dura gruparea și operațiile ce urmează.* 
- *Dacă o cerere **SQL** este concepută să elimine înregistrări/grupuri doar folosind clauza **HAVING**, atunci ar fi bine de încercat dacă este posibil și prin clauza **WHERE**. De obicei, totuși, această rescriere nu va fi posibilă.* 



Imbricarea functiilor de grup



Funcțiile de grup pot fi imbricate cu o **adâncime de 2.**

Exemplul 14

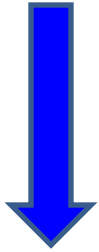
Salariul mediu maxim.

```
SELECT MAX(AVG(salariu))  
FROM angajati  
GROUP BY id_dept;
```

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

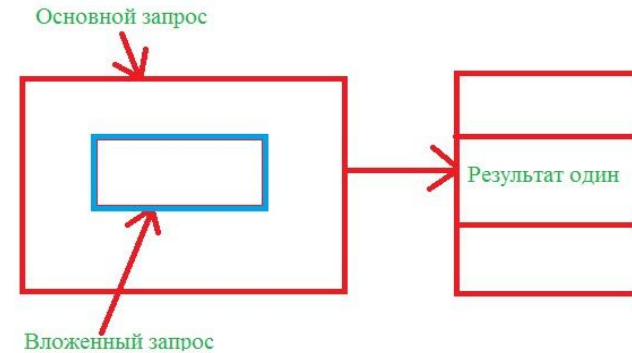
idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

Limbajul SQL

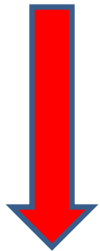
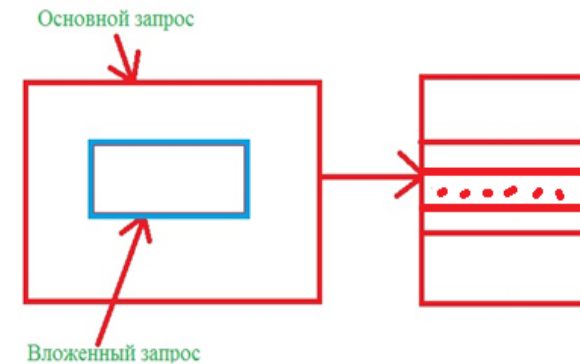


5. Subinterogări (Subqueries)

1. SINGLE ROW SUBQUERIES



2. MULTIPLE ROW SUBQUERIES

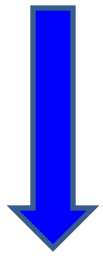


EN

<https://www.w3resource.com/sql/subqueries/understanding-sql-subqueries.php>

RU

<https://www.internet-technologies.ru/articles/podzaprosy-sql.html>

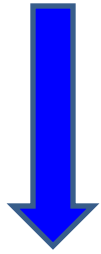


SUBQUERIES (Subinterogari)

În **SQL**, subinterogările ne permit să aflăm o **informație care ne este necesară pentru a obține informația** pe care o vrem.

- O **subinterogare (subquery)** este o instrucțiune **SELECT** care este inclusă în clauza unei alte instrucțiuni **SELECT**.

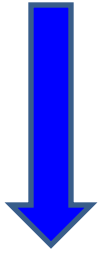




SUBQUERIES (Subinterogari)

- Subinterogarea poate fi plasata în una din următoarele clauze:
 - WHERE
 - HAVING
 - FROM
- **Subinterogarea** se execută prima dată, iar rezultatul este folosit pentru obținerea rezultatului de către interogarea principală (**outer query**).

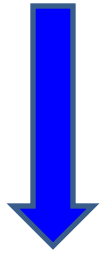




SUBQUERIES (Subinterogari)

Sintaxa generală:

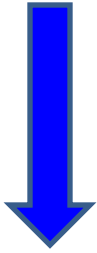
```
SELECT select_list  
FROM table  
WHERE expression operator  
                (SELECT select_list  
                FROM table);
```



SUBQUERIES (Subinterogari)

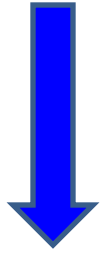
Reguli de folosire a subinterogarilor

- O **subinterogare** se pune între paranteze rotunde
- O **subinterogare** este plasata în partea dreaptă a unei condiții de comparare
- Interogarea exterioară și **subinterogarea**-ul pot prelua date din tabele diferite



SUBQUERIES (Subinterogari)

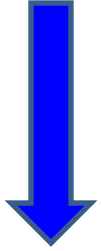
- Într-o instrucțiune **SELECT** se poate folosi o singură clauză **ORDER BY** și, dacă se folosește, trebuie să fie ultima clauza a interogării principale.
- Un subquery **nu poate avea** propria clauză **ORDER BY**.
- Singura limită a numărului de interogări este dimensiunea buffer-ului folosit de interogare.
- Dacă subinterogarea returnează **null** sau nu returnează nici o linie, atunci interogarea exterioară nu va returna nimic.



SUBQUERIES (Subinterogari)

Sunt două tipuri de subinterogări(subqueries):

- 1) **single-row subqueries** – care folosesc operatorii single-row: **>, =, >=, <, <=** și dau ca rezultat o **singură linie**.
- 2) **multiple-row subqueries** – care folosesc operatorii multiple-row: **IN, ANY, ALL** și dau ca rezultat **mai multe linii**.



Limbajul SQL

5. SUBQUERIES (Subinterogări)

1. SINGLE ROW SUBQUERIES

2. MULTIPLE ROW SUBQUERIES

SINGLE ROW SUBQUERIES

Single row-subquery

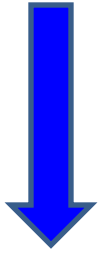
```
SELECT prenume  
FROM angajati  
WHERE salariu >
```

Aflati prenumele angajatilor care au salariul mai mare decat angajatul care se numeste Costica.

	titlu	pret	cantitatea	id_autor
GFGF	Laborator Mysql-Php vbvbnb	950.00	23	1
	MAMA	2000.00	200	3
cu	Ghid Php	1000.00	3	2
cu	Ghid Php	1000.00	3	2
	Cei trei muschetari Php	930.00	40	4
cu	Ghid Php	1000.00	3	6
cu	Ghid Php	1000.00	3	2

```
( SELECT salariu  
FROM angajati  
WHERE prenume='Costica' );
```

```
SELECT AUTOR  
FROM carti  
WHERE PRET >  
( SELECT AVG(PRET)
```

SINGLE ROW SUBQUERIES

Subcereri din mai multe tabele

Subcererile (subinterogările) nu sunt limitate la o singură interogare (cerere).

Așa cum se poate observa în exemplul următor, pot fi mai mult de o singură interogare.

De asemenea se pot face interogări din tabele diferite.

Exemplul urmator afiseaza angajatii a caror functie este acelasi cu cel al angajatului cu numarul 7369 si a caror salariu este mai mare decat cel al angajatului 7875.

Executarea unei subinterogari single-row

```
SELECT nume, functie
FROM angajati
WHERE functie =
      (SELECT functie
       FROM angajati
       WHERE id_angajat = 7369)
AND salariu >
      (SELECT salariu
       FROM angajati
       WHERE id_angajat = 7876);
```

NUME	FUNCTIE
-----	-----
MILLER	CLERK

SINGLE ROW SUBQUERIES

- Exemplul este format din 3 blocuri de cereri:
 - o cerere exterioara
 - doua cereri interne
- Blocurile de cereri interne sunt primele executate, producand rezultatele cererii: FUNCTIONAR (CLERK), respectiv 1100.
- Blocul exterior de cereri este apoi procesat si foloseste valorile returnate de catre cererile interne pentru a finaliza propriile conditii de cautare.
- Ambele cereri interne returneaza valori singulare (FUNCTIONAR si 1100), astfel ca aceasta instructiune SQL este denumita o subinterogare single-row.

SINGLE ROW SUBQUERIE

```
SELECT nume, id_functie, salariu, id_dept  
FROM angajati  
WHERE id_functie =
```

```
(SELECT id_functie  
FROM angajati  
WHERE id_angajat=141)
```

```
AND id_dept =
```

```
(SELECT id_dept  
FROM departamente  
WHERE id_locatie=1500);
```

id_autor	nume
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

titlu	pret	cantitatea	id_autor
GFGF Laborator Mysql-Php vbvbnb	950.00	23	1
MAMA	2000.00	200	3
cu Ghid Php	1000.00	3	2
cu Ghid Php	1000.00	3	2
Cei trei muschetari Php	930.00	40	4
cu Ghid Php	1000.00	3	6
cu Ghid Php	1000.00	3	2

SINGLE ROW SUBQUERIES

Se pot folosi funcțiile de grup în subinterogări.

O funcție de grup utilizată în subquery fără clauza **GROUP BY**, returnează o singură linie.

```
SELECT nume, prenume, salariu  
FROM angajati  
WHERE salariu <
```

	titlu	pret	cantitatea	id_autor
GFGF	Laborator Mysql-Php vbvbvb	950.00	23	1
	MAMA	2000.00	200	3
cu	Ghid Php	1000.00	3	2
cu	Ghid Php	1000.00	3	2
	Cei trei muschetari Php	930.00	40	4
cu	Ghid Php	1000.00	3	6
cu	Ghid Php	1000.00	3	2

```
(SELECT MAX(salariu)  
FROM angajati);
```

id_autor	nume
1	Petru
2	Dium
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

SINGLE ROW SUBQUERIES

- Subinterogările pot fi plasate și în clauza **HAVING**.
- Deoarece clauza **HAVING** are întotdeauna o condiție de grup, și subinterogarea va avea aproape întotdeauna o condiție de grup.

```
SELECT id_dept, MIN(salariu)  
FROM angajati  
GROUP BY id_dept  
HAVING MIN(salariu) >
```

idcarte	autor	titlu	pret	cantitatea	id_aut
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbnb	950.00	23	
2-2222-222-13	Vieru	MAMA	2000.00	200	
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	

```
(SELECT MIN(salariu)  
FROM angajati  
WHERE id_dept = 50);
```

SINGLE ROW SUBQUERIES

APLICAȚII

- 1) Care este numele membrilor din personalul de la firma "COSTICA S.R.L.", al căror salariu este mai mare decât angajatul cu ID-ul 12?
- 2) Care dintre angajații **Oracle** au același id al departamentului ca și cel corespunzător cu departamentul IT?

SINGLE ROW SUBQUERIES

1) Care este numele membrilor din personalul de la firma “COSTICA S.R.L.”, al căror salariu este mai mare decât angajatul cu ID-ul 12?

```
SELECT nume  
FROM angajati  
WHERE salariu >
```

	titlu	pret	cantitatea	id_autor
GFGF	Laborator Mysql-Php vbvbnb	950.00	23	1
	MAMA	2000.00	200	3
cu	Ghid Php	1000.00	3	2
cu	Ghid Php	1000.00	3	2
	Cei trei muschetari Php	930.00	40	4
cu	Ghid Php	1000.00	3	6
cu	Ghid Php	1000.00	3	2

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

```
(SELECT salariu  
FROM angajati  
WHERE id_angajat = 12);
```


SINGLE ROW SUBQUERIES

2) Care dintre angajatii **Oracle** au acelasi id al departamentului ca si cel corespunzator cu departamentul IT?

SELECT nume, prenume
FROM angajati
WHERE id_dept =

	pret	cantitatea	id_autor
Mysql-Php vbvbbb	950.00	23	1
	2000.00	200	3
	1000.00	3	2
	1000.00	3	2
uschetari Php	930.00	40	4
	1000.00	3	6
	1000.00	3	2

(**SELECT** id_dept
FROM departamente
WHERE nume_department = 'IT');

id_autor	nume_au
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

Limbajul SQL

5. SUBQUERIES (Subinterogări)

1. SINGLE ROW SUBQUERIES

2. MULTIPLE ROW SUBQUERIES

MULTIPLE ROW SUBQUERIES

Sunt acele subinterogări care dau ca rezultat mai multe valori.

Folosesc operatorii **multiple row**:

1. **IN**
2. **ANY**
3. **ALL**

Operatorul **NOT** poate fi folosit în combinație cu oricare dintre aceștia.

MULTIPLE ROW SUBQUERIES

id_autor
1
2
3
4
5
6
7
8
9

1. Operatorul IN

Operatorul **IN** este folosit dacă în interogarea exterioară clauza **WHERE** este folosită pentru a selecta acele valori care sunt egale cu una dintre valorile din lista returnată de subinterogare (**inner query**).

```
SELECT nume, salariu, id_dept
FROM angajati
WHERE salariu IN
    ( SELECT MIN(salariu)
      FROM angajati
      GROUP BY id_dept );
```

idcarte	autor	titlu	pret	c
2-2222-222-10	Petru	GFGF Laborator	950.00	Mysql-Php vbvbvb
2-2222-222-13	Vieru	MAMA	2000.00	
2-2222-2222-22	Eminescu	Ghid Php	1000.00	
2-2222-2222-23	Eminescu	Ghid Php	1000.00	
2-2222-2222-6	Vica	Cei trei muschetari	930.00	Php
2-2222-2222-8	Eminescu	Ghid Php	1000.00	
2-2222-2222-9	Eminescu	Ghid Php	1000.00	

MULTIPLE ROW SUBQUERIES

2. Operatorul ANY

Acest operator este folosit atunci când dorim ca interogarea exterioară să selecteze valori egale, mai mici sau mai mari decât cel puțin o valoare dintre cele extrase de subquery.

SELECT titlu, producator
FROM albume
WHERE an < **ANY**

(**SELECT** an
FROM albume

WHERE producator = 'Ice T');

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

MULTIPLE ROW SUBQUERIES

3. Operatorul ALL

Acest operator este folosit atunci când dorim ca interogarea exterioară să selecteze valori egale, mai mici sau mai mari decât toate valorile extrase de subquery.

```
SELECT titlu, producator  
FROM albume  
WHERE an > ALL
```

```
( SELECT an  
  FROM albume  
  WHERE producator = 'Ice T' );
```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

MULTIPLE ROW SUBQUERIES

VALORI NULL

Dacă una dintre valorile returnate de subinterogarea **multiple row** este **null**, dar celelalte valori nu sunt **null**, atunci:

- Dacă sunt folosiți operatorii **IN** sau **ANY**, interogarea exterioară va returna liniile care se potrivesc cu valorile **non-null**.
- Dacă este folosit operatorul **ALL**, interogarea exterioară nu va returna nimic.

MULTIPLE ROW SUBQUERIES

Clauzele **GROUP BY** și **HAVING**

- Pot fi folosite cu subinterogările de tip **multiple row**.

```
SELECT id_dept, MIN(salariu)
```

```
FROM angajati
```

```
GROUP BY id_dept
```

```
HAVING MIN(salariu) < ANY
```

```
( SELECT salariu
```

```
FROM angajati
```

```
WHERE id_dept IN (10,20) );
```

idcarte	autor	titlu	pret
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950.00
2-2222-222-13	Vieru	MAMA	2000.00
2-2222-2222-22	Eminescu	Ghid Php	1000.00
2-2222-2222-23	Eminescu	Ghid Php	1000.00
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00
2-2222-2222-8	Eminescu	Ghid Php	1000.00
2-2222-2222-9	Eminescu	Ghid Php	1000.00

MULTIPLE ROW SUBQUERIES

Clauzele **GROUP BY** si **HAVING**

De asemenea, se poate folosi clauza **GROUP BY** intr-o subinterogare

```
SELECT id_dept, MIN(salariu)
FROM angajati
GROUP BY id_dept
HAVING MIN(salariu) > ALL
  (SELECT MIN(salariu)
   FROM angajati
   WHERE id_dept < 50
   GROUP BY id_dept);
```

idcarte	autor	titlu	pre
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950.0
2-2222-222-13	Vieru	MAMA	2000.0
2-2222-2222-22	Eminescu	Ghid Php	1000.0
2-2222-2222-23	Eminescu	Ghid Php	1000.0
2-2222-2222-6	Vica	Cei trei muschetari Php	930.0
2-2222-2222-8	Eminescu	Ghid Php	1000.0
2-2222-2222-9	Eminescu	Ghid Php	1000.0

MULTIPLE ROW SUBQUERIES

1
2
3
4
5
6
7
8
9

APLICATII

- 1) Găsiți numele pentru toți angajații ale căror salarii sunt aceleași cu salariul minim din oricare (**any**) departament.

SELECT nume

FROM angajati

WHERE salariu = **ANY**

(SELECT MIN(salariu)

FROM angajati

GROUP BY id_dept);

idcarte	autor	titlu	p
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950
2-2222-222-13	Vieru	MAMA	2000
2-2222-2222-22	Eminescu	Ghid Php	1000
2-2222-2222-23	Eminescu	Ghid Php	1000
2-2222-2222-6	Vica	Cei trei muschetari Php	930
2-2222-2222-8	Eminescu	Ghid Php	1000
2-2222-2222-9	Eminescu	Ghid Php	1000

MULTIPLE ROW SUBQUERIES

2) Scopul interogării următoare este de a afișa salariul minim pentru fiecare departament al cărui salariu minim este mai mic decât cel mai mic salariu al angajaților din departamentul 50.

Oricum, subinterogarea nu se execută deoarece are 5 erori.

Găsiți erorile și corectați-le.

```
SELECT id_dept
FROM angajati
WHERE MIN(salariu)
HAVING MIN(salariu) >
GROUP BY id_dept
SELECT MIN(salariu)
WHERE id_dept < 50;
```

The diagram consists of five blue circles containing the numbers 1 through 5. Arrows point from each circle to a specific part of the SQL query:

- Circle 1 points to the `WHERE` clause.
- Circle 2 points to the `MIN(salariu)` in the `WHERE` clause.
- Circle 3 points to the `HAVING` clause.
- Circle 4 points to the `SELECT` clause of the subquery.
- Circle 5 points to the `WHERE` clause of the subquery.

idcarte	autor
2-2222-222-10	Petru GF
2-2222-222-13	Vieru
2-2222-2222-22	Eminescu
2-2222-2222-23	Eminescu
2-2222-2222-6	Vica
2-2222-2222-8	Eminescu
2-2222-2222-9	Eminescu

MULTIPLE ROW SUBQUERIES

Soluția corectă este următoarea:

```
SELECT id_dept, MIN(salariu)
FROM angajati
GROUP BY id_dept
HAVING MIN(salariu) <
```

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

idcarte	autor	titlu	pret	cantitatea	id_autor	
2-2222-222-10	Petru	GFGF	Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA		2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php		1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php		1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php		930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php		1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php		1000.00	3	2

```
(SELECT MIN(salariu)
FROM angajati
WHERE id_dept = 50);
```

Subcereri multilinie

- Subcererile multilinie returneaza mai mult decat o linie.
- Cu astfel de subcereri trebuie folositi operatori multilinie care pot prelucra una sau mai multe valori.

Operatorii utilizati sunt:

1. **IN** - egal cu oricare dintre membrii unei liste
2. **ANY/SOME** - compara o valoare cu fiecare (vreo) valoare returnata de subcerere
3. **ALL** - compara o valoare cu oricare (toate) din valorile returnate de subcerere

Exemplu **IN**

- Aflati angajatii care au salariul egal cu salariul cel mai mare din fiecare departament
- `SELECT nume, id_dept, salariu FROM angajati`
- `WHERE salariu IN`
 - `(SELECT MAX(salariu) FROM angajati GROUP BY id_dept)`
- Subcererea ofera salariile maxime din fiecare departament si prin cererea principala se afla angajatii cu aceste salarii.

Exemplu **ANY**

Aflati angajatii care au salariul mai mare decat vreun angajat al departamentului 20 si nu fac parte din acest departament.

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

SELECT nume, id_dept, salariu
FROM angajati
WHERE salariu > ANY
(SELECT salariu
FROM angajati
WHERE id_dept=20)
AND id_dept<>20

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

- Operatorul **ANY** (sinonim operatorului **SOME**) compara o valoare cu fiecare valoare din cele returnate de subcerere.

Astfel,

< ANY inseamna mai mic decat maximul
ANY inseamna mai mare decat minimul
= ANY este echivalent cu **IN**

Exemplu ALL

Gasiti angajatii care au salariul mai mic decat oricare (toti) angajatii de la departamentul 5.

id_autor
1
2
3
4
5
6
7
8
9

```
SELECT nume, id_dept, salariu  
FROM angajati  
WHERE salariu < ALL  
    ( SELECT salariu  
      FROM angajati  
      WHERE id_dept=5 )  
AND id_dept<>5
```

idcarte	autor	titlu	pre
2-2222-222-10	Petru GFGF	Laborator Mysql-Php vbvbvb	950.0
2-2222-222-13	Vieru	MAMA	2000.0
2-2222-2222-22	Eminescu	Ghid Php	1000.0
2-2222-2222-23	Eminescu	Ghid Php	1000.0
2-2222-2222-6	Vica	Cei trei muschetari Php	930.0
2-2222-2222-8	Eminescu	Ghid Php	1000.0
2-2222-2222-9	Eminescu	Ghid Php	1000.0

- Operatorul **ALL** din cererea principala compara o valoare cu oricare valoare returnata de subcerere.

Astfel:

> **ALL** inseamna mai mare decat maximul

< **ALL** inseamna mai mic decat minimul

Imbricarea subcererilor

Subcererile pot fi folosite si in interiorul altor subcereri.

Exemplu

Gasiti numele, functia, data angajarii si salariul angajatilor al caror salariu este superior celui mai mare salariu al vreunei persoane angajate dupa data de 05-JUN-1982

```

SELECT nume, functie, data_ang, salariu
      FROM angajati
     WHERE salariu >
      (SELECT MAX(salariu)
      FROM angajati
     WHERE data_ang IN
      (SELECT data_ang
      FROM angajati
     WHERE data_ang > '05-JUN-1982'))

```

id_autor	nume_au
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

idcarte	autor
2-2222-222-10	Petru GFG
2-2222-222-13	Vieru
2-2222-2222-22	Eminescu
2-2222-2222-23	Eminescu
2-2222-2222-6	Vica
2-2222-2222-8	Eminescu
2-2222-2222-9	Eminescu

Numarul maxim de imbricari pentru o subcerere este de 255.

Subcereri corelate

- O subcerere corelata este o subcerere care se executa o data pentru fiecare linie considerata de cererea principala si care la executie foloseste o valoare dintr-o coloana din cererea exterioara.
- Ea se poate identifica prin folosirea unei coloane a cererii exterioare in clauza operatorului cererii interioare.

Exemplu

Gasiti angajatii care au un salariu superior salariului mediu al departamentului lor.

```
SELECT nume, salariu, id_dept  
FROM angajati A  
WHERE salariu >  
      (SELECT AVG(salariu)  
        FROM angajati  
        WHERE (id_dept=A.id_dept))  
ORDER BY id_dept
```

id_autor	nume
1	P
2	D
3	V
4	L
5	I
6	G
7	C
8	N
9	V

idcarte	autor	titlu	pret
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00
2-2222-222-13	Vieru	MAMA	2000.00
2-2222-2222-22	Eminescu	Ghid Php	1000.00
2-2222-2222-23	Eminescu	Ghid Php	1000.00
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00
2-2222-2222-8	Eminescu	Ghid Php	1000.00
2-2222-2222-9	Eminescu	Ghid Php	1000.00

Valori de NULL intr-o subcerere

- In cazul in care subcererea returneaza vreuna din valori NULL si cererea principala are operator NOT IN, atunci cererea principala nu va returna niciun rand.
- Motivul este ca *o comparatie cu NULL conduce la un rezultat NULL.*

Exemplu

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

Gasiti angajatii care nu au subordonati.

```
SELECT nume  
FROM angajati  
WHERE id_angajat NOT IN  
        ( SELECT manager  
          FROM angajati );
```

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

- Astfel ori de cate ori valoarea **NULL** face parte din raspunsurile subcererii nu trebuie folosit operatorul **NOT IN**.
- De fapt operatorul **NOT IN** este echivalent cu **<> ALL**.
- Returnarea de valori **NULL** de catre subcerere nu prezinta nici o problema in cazul operatorului **IN** in cererea principala (in echivalent cu **= ALL**).

Exemplu

Gasiti angajatii care au subordonati.

SELECT nume

FROM angajati

WHERE id_angajat IN

**(SELECT manager
FROM angajati);**

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

idcarte	autor	titlu	pret	cantita
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbvb	950.00	
2-2222-222-13	Vieru	MAMA	2000.00	
2-2222-2222-22	Eminescu	Ghid Php	1000.00	
2-2222-2222-23	Eminescu	Ghid Php	1000.00	
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	
2-2222-2222-8	Eminescu	Ghid Php	1000.00	
2-2222-2222-9	Eminescu	Ghid Php	1000.00	

- In cazul utilizarii operatorului **NOT IN** in cererea principala trebuie avut grija sa se excluda valorile **NULL** din raspunsurile subcererii.

Exemplu

Gasiti angajatii care nu au subordonati.

id_autor	nume_autor
1	Petru
2	Diuma
3	Vieru
4	Lena
5	Ioana
6	Ghita
7	Colea
8	Nicu
9	Valea

```

SELECT nume
FROM angajati
WHERE id_angajat NOT IN
      ( SELECT manager
        FROM angajati
        WHERE manager IS NOT NULL );

```

idcarte	autor	titlu	pret	cantitatea	id_autor
2-2222-222-10	Petru	GFGF Laborator Mysql-Php vbvbbb	950.00	23	1
2-2222-222-13	Vieru	MAMA	2000.00	200	3
2-2222-2222-22	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-23	Eminescu	Ghid Php	1000.00	3	2
2-2222-2222-6	Vica	Cei trei muschetari Php	930.00	40	4
2-2222-2222-8	Eminescu	Ghid Php	1000.00	3	6
2-2222-2222-9	Eminescu	Ghid Php	1000.00	3	2

Sfaturi în utilizarea subinterogărilor

1. Includerea subinterogărilor în paranteze
2. Plasarea subinterogărilor în partea dreapta a operatorului de comparare
3. A nu se adauga clauza **ORDER BY** într-o subinterogare
4. Folosirea operatorilor single-row în subinterogari single-row
5. Folosirea operatorilor multiple-row în subinterogari multiple-row

Concluzii

1. O subinterogare este o instructiune **SELECT** inclusa într-o clauza a altei instructiuni **SQL**.
2. Subinterogările sunt folositoare atunci când interogarea se bazeaza pe criterii necunoscute.
3. Subinterogările au următoarele caracteristici:
 - a) Pot transmite un rand de date instructiunii principale care contine un **operator single-row**, precum: **=, <>, >, >=, <** sau **<=**;
 - b) Pot transmite rînduri multiple de date instructiunii principale care contine un **operator multiple-row**, precum: **IN, ANY** sau **ALL**;
 - c) Sunt primele procesate de catre server-ul **Oracle**, iar clauzele **WHERE** si **HAVING** folosesc rezultatele;
 - d) Pot contine functii de grup

