

Limbaajul de interogare a datelor

- Sintaxa comenzii SELECT
- Ordonare folosind ORDER BY
- Filtrarea folosind WHERE
- Numarul de inregistrari COUNT (functie agregata)
- Utilizarea JOIN

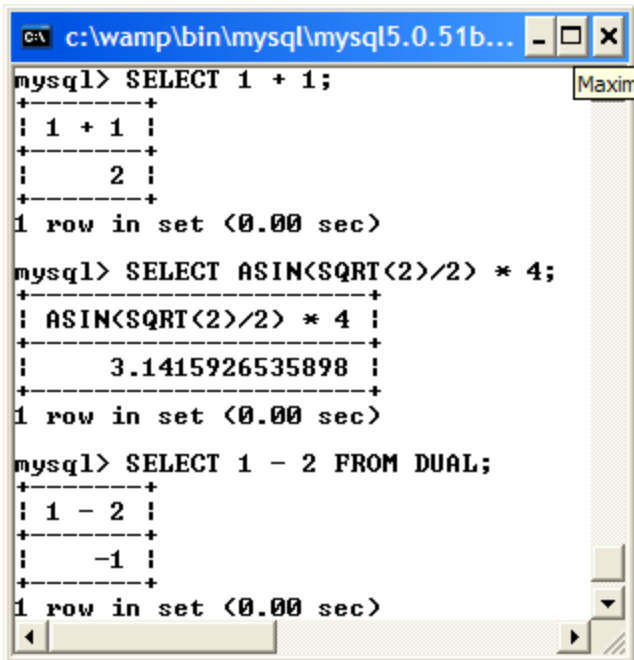
Limbaajul de interogare a datelor

SELECT

```
SELECT [ALL | DISTINCT | DISTINCTROW ]
      [HIGH_PRIORITY]
      [STRAIGHT_JOIN]
      [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
      [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
      select_expr, ...
      [FROM table_references]
      [WHERE where_condition]
      [GROUP BY {col_name | expr | position}
        [ASC | DESC], ... [WITH ROLLUP]]
      [HAVING where_condition]
      [ORDER BY {col_name | expr | position}
        [ASC | DESC], ...] [LIMIT {[offset,] row_count | row_count OFFSET offset}]
      [PROCEDURE procedure_name(argument_list)]
      [INTO OUTFILE 'file_name' export_options
        | INTO DUMPFILE 'file_name'
        | INTO var_name [, var_name]]
      [FOR UPDATE | LOCK IN SHARE MODE]]
```

```
SELECT select_expr, ...
      [FROM table_references]
      [WHERE where_condition]
```

Nici o tabela selectata:



```
mysql> SELECT 1 + 1;
+-----+
| 1 + 1 |
+-----+
|      2 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT ASIN(SQRT(2)/2) * 4;
+-----+
| ASIN(SQRT(2)/2) * 4 |
+-----+
| 3.1415926535898 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT 1 - 2 FROM DUAL;
+-----+
| 1 - 2 |
+-----+
|     -1 |
+-----+
1 row in set (0.00 sec)
```

Se poate utiliza instructiunea SELECT fara a preciza nici o referinta la vreo tabela; se pot astfel calcula expresii aritmetice; pentru comoditate, se poate utiliza sintaxa incluzand referinta la tabela DUAL (nu trebuie definita).

SQL

Exemple:

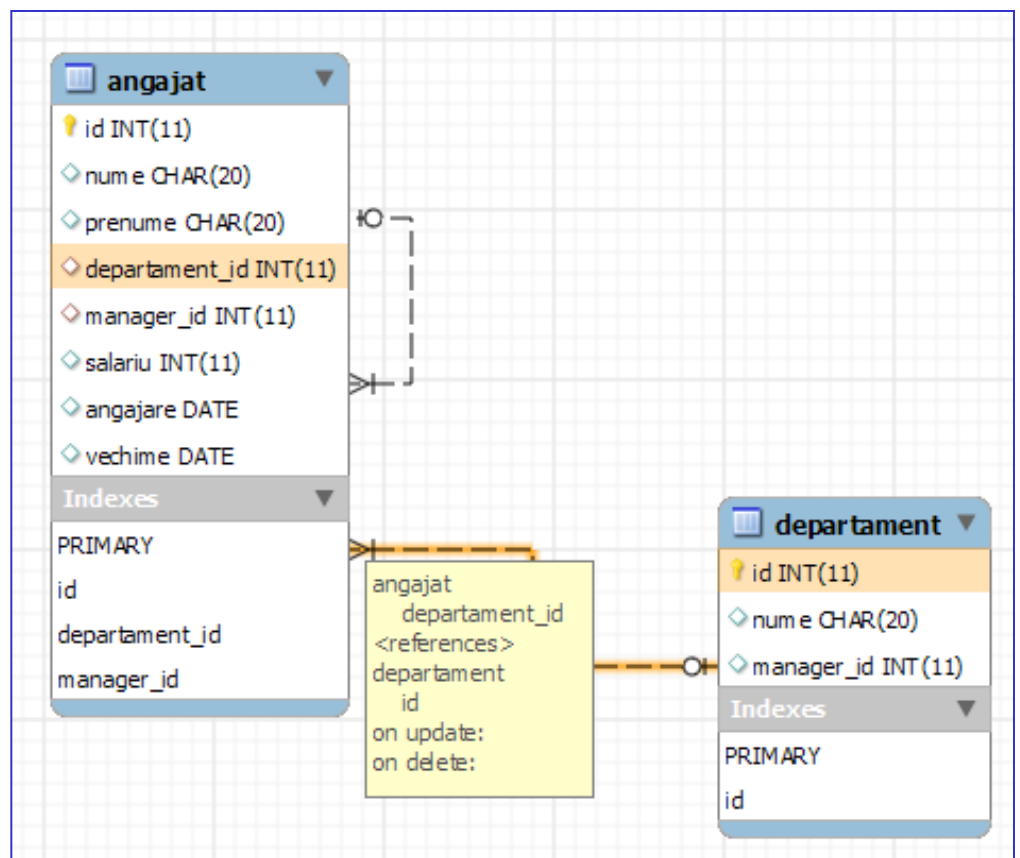
O baza de date cu doua tabele:

- angajat;
- departament;

Datele despre angajat sunt nume, prenume, salariu, data angajarii, vechimea cat si indecsii pentru managerul si departamentul sau. Pentru a modela aceste relatii, se folosesc doua chei straine:

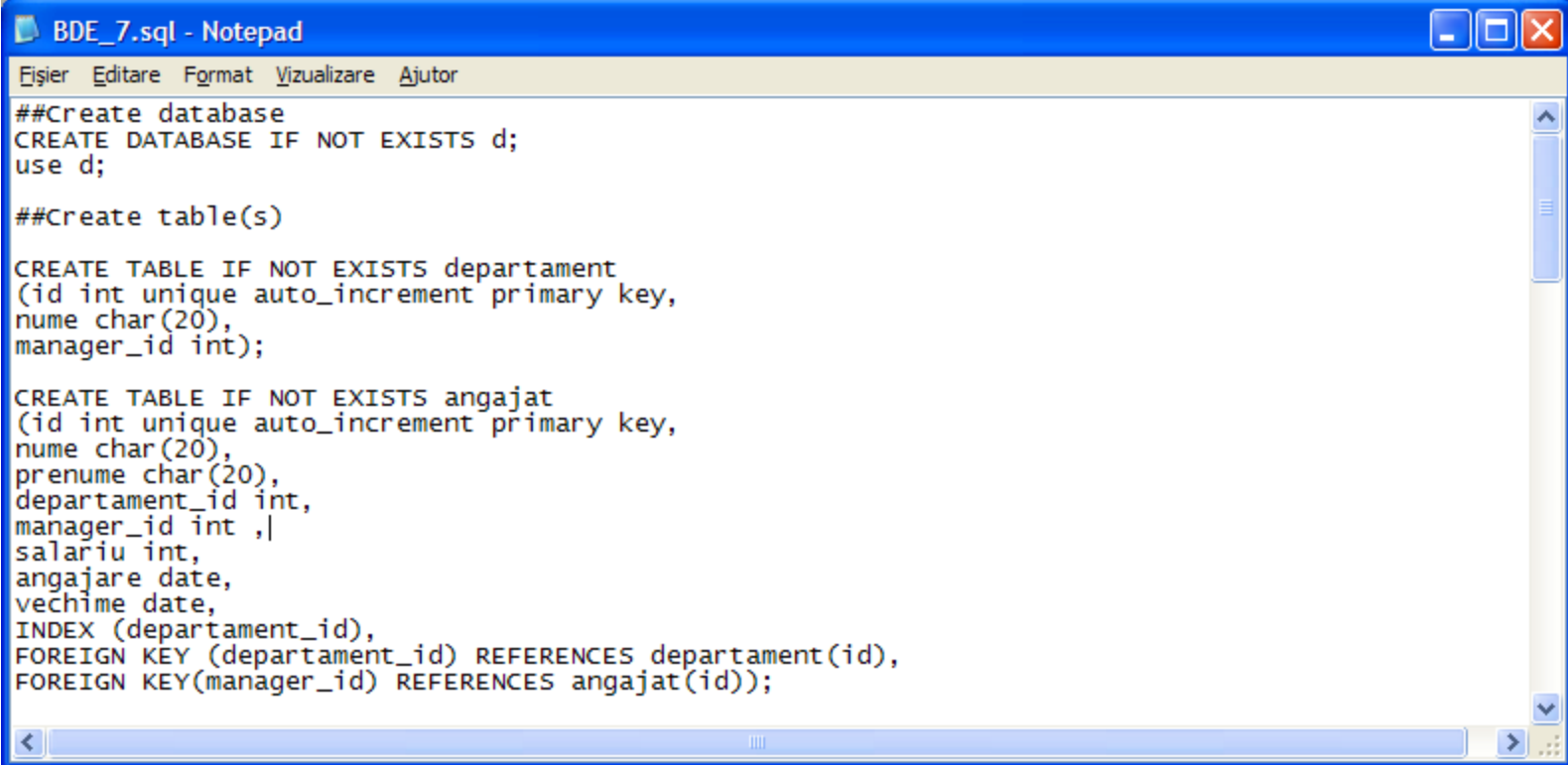
- departament_id;
- manager_id;

Actiunile referentiale nu sunt definite.



Exemple:

Crearea tabelelor angajat si departament; se creeaza mai intai tabela departament, datorita existentei unei referinte in care cheia straina este departament_id din tabela angajat iar coloana referita este cheia primara din tabela departament;



```
BDE_7.sql - Notepad
Fișier Editare Format Vizualizare Ajutor

##Create database
CREATE DATABASE IF NOT EXISTS d;
use d;

##Create table(s)

CREATE TABLE IF NOT EXISTS departament
(id int unique auto_increment primary key,
nume char(20),
manager_id int);

CREATE TABLE IF NOT EXISTS angajat
(id int unique auto_increment primary key,
nume char(20),
prenume char(20),
departament_id int,
manager_id int ,|
salariu int,
angajare date,
vechime date,
INDEX (departament_id),
FOREIGN KEY (departament_id) REFERENCES departament(id),
FOREIGN KEY(manager_id) REFERENCES angajat(id));
```

SQL

```
BDE_7.sql - Notepad
Fișier Editare Format Vizualizare Ajutor

INSERT INTO departament (nume, manager_id) VALUES
('R&D', 1), ('QA', 2), ('IT', 3), ('Backend', 4), ('HR', 5);

INSERT INTO angajat
(nume, prenume, departament_id, manager_id, salariu, angajare)
VALUES
('Popa', 'Ion', 1, NULL, 8000, '2000-1-12'),
('Popescu', 'Maria', 1, 1, 3000, '2003-5-6'),
('Marinescu', 'Vasile', 1, 1, 5000, '2004-6-3'),
('Ionescu', 'Andrei', 1, NULL, 3000, '2002-1-1'),
('Vasilescu', 'Ana', 2, NULL, 2000, '2006-3-3'),
('Dragan', 'Dinu', 2, 5, 2000, '2004-11-12'),
('Mihailescu', 'Adrian', 5, NULL, 2500, '2006-10-12'),
('Teodorescu', 'Matei', 3, NULL, 2000, '2005-1-12'),
('Popescu', 'Vasile', 3, 8, 3000, '2005-9-9'),
('Mateescu', 'Dumitru', 3, 8, 3000, '2007-2'),
('Calinescu', 'Alin', 4, NULL, 3200, '2005-8-2'),
('Popescu', 'Mihaela', 4, 12, 1500, '2005-4-8'),
('Ionescu', 'Diana', 5, NULL, 5000, '2001-1-12');
```

Inregistrarile din cele doua tabele:

	id	nume	manager_id
▶	1	R&D	1
	2	QA	2
	3	IT	3
	4	Backend	4
	5	HR	5

	id	nume	prenume	departament_id	manager_id	salariu	angajare	vechime
▶	1	Popa	Ion	1	NULL	9000	2000-01-12	NULL
	2	Popescu	Maria	1	1	4000	2003-05-06	NULL
	3	Marinescu	Vasile	1	1	4500	2004-06-03	NULL
	4	Ionescu	Andrei	1	NULL	3500	2002-01-01	NULL
	5	Vasilescu	Ana	2	NULL	2500	2006-03-03	NULL
	6	Dragan	Dinu	2	5	2000	2004-11-12	NULL
	7	Mihailescu	Adrian	5	NULL	3500	2006-10-12	NULL
	8	Teodorescu	Matei	3	NULL	3000	2005-01-12	NULL
	9	Popescu	Vasile	3	8	4000	2005-09-09	NULL
	10	Mateescu	Dumitru	3	8	3500	2007-02-05	NULL
	11	Calinescu	Alin	4	NULL	3200	2005-08-02	NULL
	12	Popescu	Mihaela	4	12	1800	2005-04-08	NULL
	13	Ionescu	Diana	5	NULL	5500	2001-01-12	NULL

SQL

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT * FROM angajat;
```

id	nume	prenume	departament_id	manager_id	salariu	angajare	vechime
1	Popa	Ion	1	NULL	8000	2000-01-12	NULL
2	Popescu	Maria	1	1	3000	2003-05-06	NULL
3	Marinescu	Uasile	1	1	5000	2004-06-03	NULL
4	Ionescu	Andrei	1	NULL	3000	2002-01-01	NULL
5	Uasilescu	Ana	2	NULL	2000	2006-03-03	NULL
6	Dragan	Dinu	2	5	2000	2004-11-12	NULL
7	Mihailescu	Adrian	5	NULL	2500	2006-10-12	NULL
8	Teodorescu	Matei	3	NULL	2000	2005-01-12	NULL
9	Popescu	Uasile	3	8	3000	2005-09-09	NULL
10	Mateescu	Dumitru	3	8	3000	0000-00-00	NULL
11	Calinescu	Alin	4	NULL	3200	2005-08-02	NULL
12	Popescu	Mihaela	4	12	1500	2005-04-08	NULL
13	Ionescu	Diana	5	NULL	5000	2001-01-12	NULL

```
13 rows in set (0.00 sec)
```

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\...
mysql> SELECT nume, prenume FROM angajat
```

nume	prenume
Popa	Ion
Popescu	Maria
Marinescu	Uasile
Ionescu	Andrei
Uasilescu	Ana
Dragan	Dinu
Mihailescu	Adrian
Teodorescu	Matei
Popescu	Uasile
Mateescu	Dumitru
Calinescu	Alin
Popescu	Mihaela
Ionescu	Diana

```
13 rows in set (0.00 sec)
```

Clauza FROM precizeaza tabela (tabelele) din care se face selectia;

Se pot preciza toate (*) coloanele sau un grup de coloane, in orice ordine; rezultatul prezentat nu este ordonat, daca nu se precizeaza aceasta;

SQL

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume, prenume FROM angajat ORDER BY nume, prenume;
```

nume	prenume
Calinescu	Alin
Dragan	Dinu
Ionescu	Andrei
Ionescu	Diana
Marinescu	Uasile
Mateescu	Dumitru
Mihailescu	Adrian
Popa	Ion
Popescu	Maria
Popescu	Mihaela
Popescu	Uasile
Teodorescu	Matei
Uasilescu	Ana

```
13 rows in set (0.00 sec)
```

Clauza ORDER BY se utilizeaza pentru a ordona (default e ascendent) dupa valoarea coloanelor precizate odata cu clauza ORDER BY. Ordonarea se face intai dupa prima coloana, apoi dupa a doua etc. Pentru a ordona descrescator, se precizeaza clauza DESC, dupa fiecare coloana. Acolo unde nu se precizeaza, se va folosi ordonarea ascendenta. Nu trebuie sa ordonam neaparat dupa coloanele selectate; Pentru ordonare, se pot folosi si numerele de ordine ale coloanelor folosite pentru criteriul de ordonare.

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume, prenume FROM angajat ORDER BY nume DESC, prenume DESC;
```

nume	prenume
Uasilescu	Ana
Teodorescu	Matei
Popescu	Uasile
Popescu	Mihaela
Popescu	Maria
Popa	Ion
Mihailescu	Adrian
Mateescu	Dumitru
Marinescu	Uasile
Ionescu	Diana
Ionescu	Andrei
Dragan	Dinu
Calinescu	Alin

```
13 rows in set (0.02 sec)
```

SQL

Selectati numele, prenumele, departamentul, salariul angajatilor ordonate dupa id-ul departamentului (descrescator) si dupa salariu;

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume AS Nume, prenume AS Prenume, departament_id AS Dept, Salariu
-> FROM angajat
-> ORDER BY departament_id DESC, salariu;
+-----+-----+-----+-----+
| Nume      | Prenume | Dept | Salariu |
+-----+-----+-----+-----+
| Mihailescu | Adrian | 5    | 2500    |
| Ionescu    | Diana  | 5    | 5000    |
| Popescu   | Mihaela | 4    | 1500    |
| Calinescu | Alin   | 4    | 3200    |
| Teodorescu | Matei  | 3    | 2000    |
| Popescu   | Uasile | 3    | 3000    |
| Mateescu  | Dumitru | 3    | 3000    |
| Uasilescu | Ana    | 2    | 2000    |
| Dragan    | Dinu   | 2    | 2000    |
| Ionescu   | Andrei | 1    | 3000    |
| Popescu   | Maria  | 1    | 3000    |
| Marinescu | Uasile | 1    | 5000    |
| Popa      | Ion    | 1    | 8000    |
+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

Selectati numele angajatilor care lucreaza la departamentul de R&D;

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mys...
mysql> SELECT nume
-> FROM angajat
-> WHERE departament_id = 1;
+-----+
| nume |
+-----+
| Popa |
| Popescu |
| Marinescu |
| Ionescu |
+-----+
4 rows in set (0.03 sec)
```

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT angajat.nume
-> FROM angajat, departament
-> WHERE angajat.departament_id = departament.id
-> AND departament.nume = 'R&D';
+-----+
| nume |
+-----+
| Popa |
| Popescu |
| Marinescu |
| Ionescu |
+-----+
4 rows in set (0.03 sec)
```

SQL

Selectati numele angajatilor care NU lucreaza la departamentul de R&D;

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT angajat.nume
-> FROM angajat, departament
-> WHERE angajat.departament_id = departament.id
-> AND departament.nume <> 'R&D';
```

nume
Uasilescu
Dragan
Mihailescu
Teodorescu
Popescu
Mateescu
Calinescu
Popescu
Ionescu

9 rows in set (0.03 sec)

Clauza WHERE se utilizeaza pentru a preciza conditiile de selectie (filtrare) pentru inregistrarile selectate; nu e obligatoriu ca atributele selectate sa fie aceleasi cu atributele care sunt utilizate in conditia de filtrare; in exemplul alaturat, se selecteaza numele si prenumele, conditia de selectie fiind apartenta la un departament anume, precizat fie prin ID-ul sau fie prin numele obtinut printr-un JOIN cu tabela de departamente.

Angajatii care au salariu mai mare de 3000 lei

```
c:\wamp\bin\mysql\mysql...
mysql> SELECT nume, prenume
-> FROM angajat
-> WHERE salariu > 3000;
```

nume	prenume
Popa	Ion
Marinescu	Uasile
Calinescu	Alin
Ionescu	Diana

4 rows in set (0.00 sec)

```
c:\wamp\bin\mysql\mysql5.0.51b\...
mysql> SELECT nume, prenume, salariu
-> FROM angajat
-> WHERE salariu > 3000;
```

nume	prenume	salariu
Popa	Ion	8000
Marinescu	Uasile	5000
Calinescu	Alin	3200
Ionescu	Diana	5000

4 rows in set (0.00 sec)

```
mysql>
```

SQL

Angajatii care au salariu de 3000 lei

```
c:\wamp\bin\mysql\mysql5.0.51b\... - _ x
mysql> SELECT nume, prenume, salariu
-> FROM angajat
-> WHERE salariu = 3000;
+-----+-----+-----+
| nume   | prenume | salariu |
+-----+-----+-----+
| Popescu | Maria   | 3000    |
| Ionescu | Andrei  | 3000    |
| Popescu | Uasile  | 3000    |
| Mateescu | Dumitru | 3000    |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> _
```

Angajatii care au salariul intre 3000 si 5000 de lei (incluzand limitele)

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe - _ x
mysql> SELECT nume, prenume, salariu
-> FROM angajat
-> WHERE salariu BETWEEN 3000 AND 5000;
+-----+-----+-----+
| nume   | prenume | salariu |
+-----+-----+-----+
| Popescu | Maria   | 3000    |
| Marinescu | Uasile | 5000    |
| Ionescu | Andrei  | 3000    |
| Popescu | Uasile  | 3000    |
| Mateescu | Dumitru | 3000    |
| Calinescu | Alin  | 3200    |
| Ionescu | Diana   | 5000    |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

Clauza BETWEEN se utilizeaza pentru a preciza limitele de variatie ale unui atribut utilizat cu clauza WHERE; limitele sunt incluse;

SQL

Angajatii care au salariul intre 3000 si 5000 de lei (incluzand limitele)

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume, prenume, salariu
-> FROM angajat
-> WHERE salariu >= 3000 AND salariu <= 5000;
+-----+-----+-----+
| nume   | prenume | salariu |
+-----+-----+-----+
| Popescu | Maria   | 3000    |
| Marinescu | Uasile | 5000    |
| Ionescu | Andrei  | 3000    |
| Popescu | Uasile  | 3000    |
| Mateescu | Dumitru | 3000    |
| Calinescu | Alin   | 3200    |
| Ionescu | Diana   | 5000    |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

Angajatii care au salariul intre 3000 si 5000 de lei (excluzand limitele)

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume, prenume, salariu
-> FROM angajat
-> WHERE salariu > 3000 AND salariu < 5000;
+-----+-----+-----+
| nume   | prenume | salariu |
+-----+-----+-----+
| Calinescu | Alin   | 3200    |
+-----+-----+-----+
1 row in set (0.00 sec)
```

SQL

Angajatii care nu au manager

```
C:\> c:\wamp\bin\mysql\mysql5.0.51... - - - X
mysql> SELECT nume, prenume
-> FROM angajat
-> WHERE manager_id IS NULL;
+-----+-----+
| nume   | prenume |
+-----+-----+
| Popa   | Ion     |
| Ionescu| Andrei  |
| Uasilescu| Ana    |
| Mihailescu| Adrian|
| Teodorescu| Matei  |
| Calinescu| Alin   |
| Ionescu| Diana   |
+-----+-----+
7 rows in set (0.00 sec)
```

Angajatii ai caror manager este Popa Ion

```
C:\> c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe - - - X
mysql> SELECT nume, prenume
-> FROM angajat
-> WHERE manager_id IN
-> <SELECT id
-> FROM angajat
-> WHERE nume='Popa' AND prenume='Ion'>;
+-----+-----+
| nume   | prenume |
+-----+-----+
| Popescu| Maria   |
| Marinescu| Uasile  |
+-----+-----+
2 rows in set (0.06 sec)
mysql> _
```

Angajatii ai caror manager are ID-ul 1

```
C:\> c:\wamp\bin\mysql\mysql5.0.51... - - - X
-+
13 rows in set (0.00 sec)
mysql> SELECT nume, prenume
-> FROM angajat
-> WHERE manager_id = 1;
+-----+-----+
| nume   | prenume |
+-----+-----+
| Popescu| Maria   |
| Marinescu| Uasile  |
+-----+-----+
2 rows in set (0.00 sec)
mysql> _
```

SQL

Angajatii ai caror manager NU este Popa Ion si care au salariu mai mare de 1000 lei

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume, prenume
-> FROM angajat
-> WHERE manager_id NOT IN
-> (SELECT id
-> FROM angajat
-> WHERE nume='Popa' AND prenume='Ion')
-> AND salariu > 1000;
```

nume	prenume
Dragan	Dinu
Popescu	Uasile
Mateescu	Dumitru
Popescu	Mihaela

4 rows in set (0.00 sec)

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume, prenume
-> FROM angajat
-> WHERE manager_id IS NULL
-> OR manager_id NOT IN
-> (SELECT id
-> FROM angajat
-> WHERE nume='Popa' AND prenume='Ion')
-> AND salariu > 1000;
```

nume	prenume
Popa	Ion
Ionescu	Andrei
Uasilescu	Ana
Dragan	Dinu
Mihailescu	Adrian
Teodorescu	Matei
Popescu	Uasile
Mateescu	Dumitru
Calinescu	Alin
Popescu	Mihaela
Ionescu	Diana

11 rows in set (0.00 sec)

```
mysql> _
```

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume, prenume, salariu, manager_id
-> FROM angajat
-> WHERE salariu > 1000;
```

nume	prenume	salariu	manager_id
Popa	Ion	8000	NULL
Popescu	Maria	3000	1
Marinescu	Uasile	5000	1
Ionescu	Andrei	3000	NULL
Uasilescu	Ana	2000	NULL
Dragan	Dinu	2000	5
Mihailescu	Adrian	2500	NULL
Teodorescu	Matei	2000	NULL
Popescu	Uasile	3000	8
Mateescu	Dumitru	3000	8
Calinescu	Alin	3200	NULL
Popescu	Mihaela	1500	12
Ionescu	Diana	5000	NULL

13 rows in set (0.01 sec)

SQL

Angajatii care au salariu mai mare de 2500 lei si lucreaza la Backend

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT angajat.nume AS Angajat, departament.nume AS Departament, salariu AS Salariu
  -> FROM angajat, departament
  -> WHERE
  -> (angajat.departament_id = departament.id
  -> AND departament.nume = 'Backend')
  -> AND salariu > 2500;
+-----+-----+-----+
| Angajat | Departament | Salariu |
+-----+-----+-----+
| Calinescu | Backend | 3200 |
+-----+-----+-----+
1 row in set (0.00 sec)
mysql> _
```


SQL

Angajatii, ordonati dupa vechime

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT nume, prenume, YEAR(CURRENT_DATE()) - YEAR(angajare) AS Uechime
-> FROM angajat
-> ORDER BY Uechime;
```

nume	prenume	Uechime
Mihailescu	Adrian	2
Uasilescu	Ana	2
Popescu	Uasile	3
Teodorescu	Matei	3
Popescu	Mihaela	3
Calinescu	Alin	3
Marinescu	Uasile	4
Dragan	Dinu	4
Popescu	Maria	5
Ionescu	Andrei	6
Ionescu	Diana	7
Popa	Ion	8
Mateescu	Dumitru	2008

```
13 rows in set (0.00 sec)
```

Join si multiple join

Angajatii, ordonati dupa departamente

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT angajat.nume, angajat.prenume, departament.nume AS Departament
-> FROM angajat, departament
-> WHERE angajat.departament_id = departament.id
-> ORDER BY Departament;
+-----+-----+-----+
| nume      | prenume | Departament |
+-----+-----+-----+
| Calinescu | Alin    | Backend     |
| Popescu   | Mihaela | Backend     |
| Mihailescu | Adrian | HR          |
| Ionescu   | Diana   | HR          |
| Teodorescu | Matei   | IT          |
| Popescu   | Uasile  | IT          |
| Mateescu  | Dumitru | IT          |
| Uasilescu | Ana     | QA          |
| Dragan    | Dinu    | QA          |
| Marinescu | Uasile  | R&D         |
| Ionescu   | Andrei  | R&D         |
| Popa      | Ion     | R&D         |
| Popescu   | Maria   | R&D         |
+-----+-----+-----+
13 rows in set (0.02 sec)
```

Cand se utilizeaza atat clauza WHERE cat si clauza ORDER BY, WHERE apare intotdeauna inaintea lui ORDER BY.

SQL

Numarul de angajati de la fiecare departament

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT departament.nume AS Departament, count(*) AS 'Numar Angajati'
-> FROM angajat, departament
-> WHERE angajat.departament_id = departament.id
-> GROUP BY Departament;
+-----+-----+
| Departament | Numar Angajati |
+-----+-----+
| Backend     | 2               |
| HR          | 2               |
| IT          | 3               |
| QA          | 2               |
| R&D         | 4               |
+-----+-----+
5 rows in set (0.00 sec)
```

Numarul de angajati de la fiecare departament, ordonat descrescator dupa numarul de angajati din fiecare departament

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> SELECT departament.nume AS Departament, count(*) AS 'Numar Angajati'
-> FROM angajat, departament
-> WHERE angajat.departament_id = departament.id
-> GROUP BY Departament
-> ORDER BY count(*) DESC;
+-----+-----+
| Departament | Numar Angajati |
+-----+-----+
| R&D         | 4               |
| IT          | 3               |
| Backend     | 2               |
| HR          | 2               |
| QA          | 2               |
+-----+-----+
5 rows in set (0.00 sec)
```

SQL

Listati persoanele care nu au manager si departamentele la care lucreaza

```
select
  a.id as ID, a.nume as Nume, a.prenume as Prenume,
  d.nume AS Dept
from angajat a
join departament d
on a.departament_id = d.id
where a.manager_id is null;
```

ID	Nume	Prenume	Dept
1	Popa	Ion	R&D
4	Ionescu	Andrei	R&D
5	Vasilescu	Ana	QA
8	Teodorescu	Matei	IT
11	Calinescu	Alin	Backend
7	Mihailescu	Adrian	HR
13	Ionescu	Diana	HR

SQL

Listati persoanele care au manageri, managerul lor si departamentul la care lucreaza

```
select
  a.id as ID, a.nume as Nume, a.prenume as Prenume,
  d.nume as Dept,
  concat(am.nume, " ", am.prenume) as Manager
from angajat a
join departament d on
  a.departament_id = d.id
join
  angajat am on
  a.manager_id = am.id;
```

ID	Nume	Prenume	Dept	Manager
2	Popescu	Maria	R&D	Popa Ion
3	Marinescu	Vasile	R&D	Popa Ion
6	Dragan	Dinu	QA	Vasilescu Ana
9	Popescu	Vasile	IT	Teodorescu Matei
10	Mateescu	Dumitru	IT	Teodorescu Matei
12	Popescu	Mihaela	Backend	Popescu Mihaela

SQL

Listati persoanele de la R&D care au manager, si managerul lor este tot la R&D

```
select
  a.id as ID, a.nume as Nume, a.prenume as Prenume,
  d.nume as Dept,
  concat(am.nume, " ", am.prenume) as Manager,
  (
    select distinct departament.nume as 'Manager Dept'
    from departament join angajat on departament.id = angajat.departament_id
    where departament.id = am.departament_id
  ) as 'Manager Dept'
from angajat a
join departament d on
  a.departament_id = d.id
join
  angajat am on
  a.manager_id = am.id
where am.id in
  (
    select distinct aa.manager_id
    from angajat aa
    join departament dd
    on aa.departament_id = dd.id
    where dd.nume = 'R&D'
  );
```

ID	Nume	Prenume	Dept	Manager	Manager Dept
2	Popescu	Maria	R&D	Popa Ion	R&D
3	Marinescu	Vasile	R&D	Popa Ion	R&D