# SQL

## Limbajul de interogare a datelor

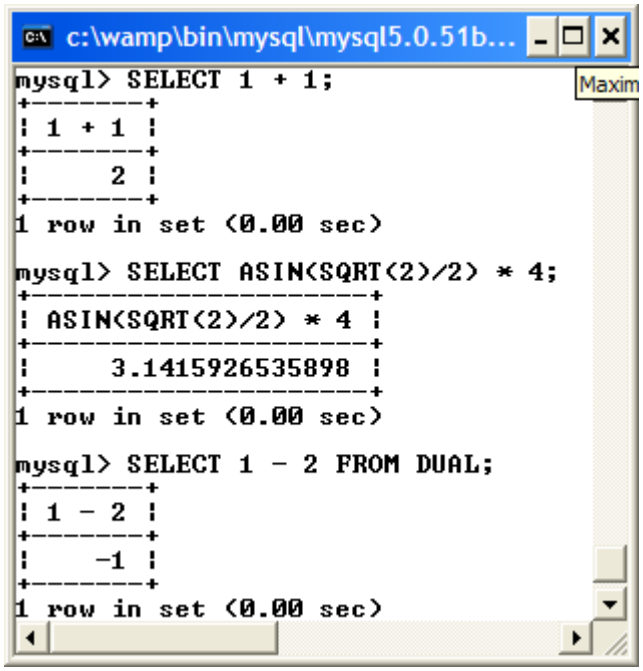
### SELECT

```
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]
```

# SQL

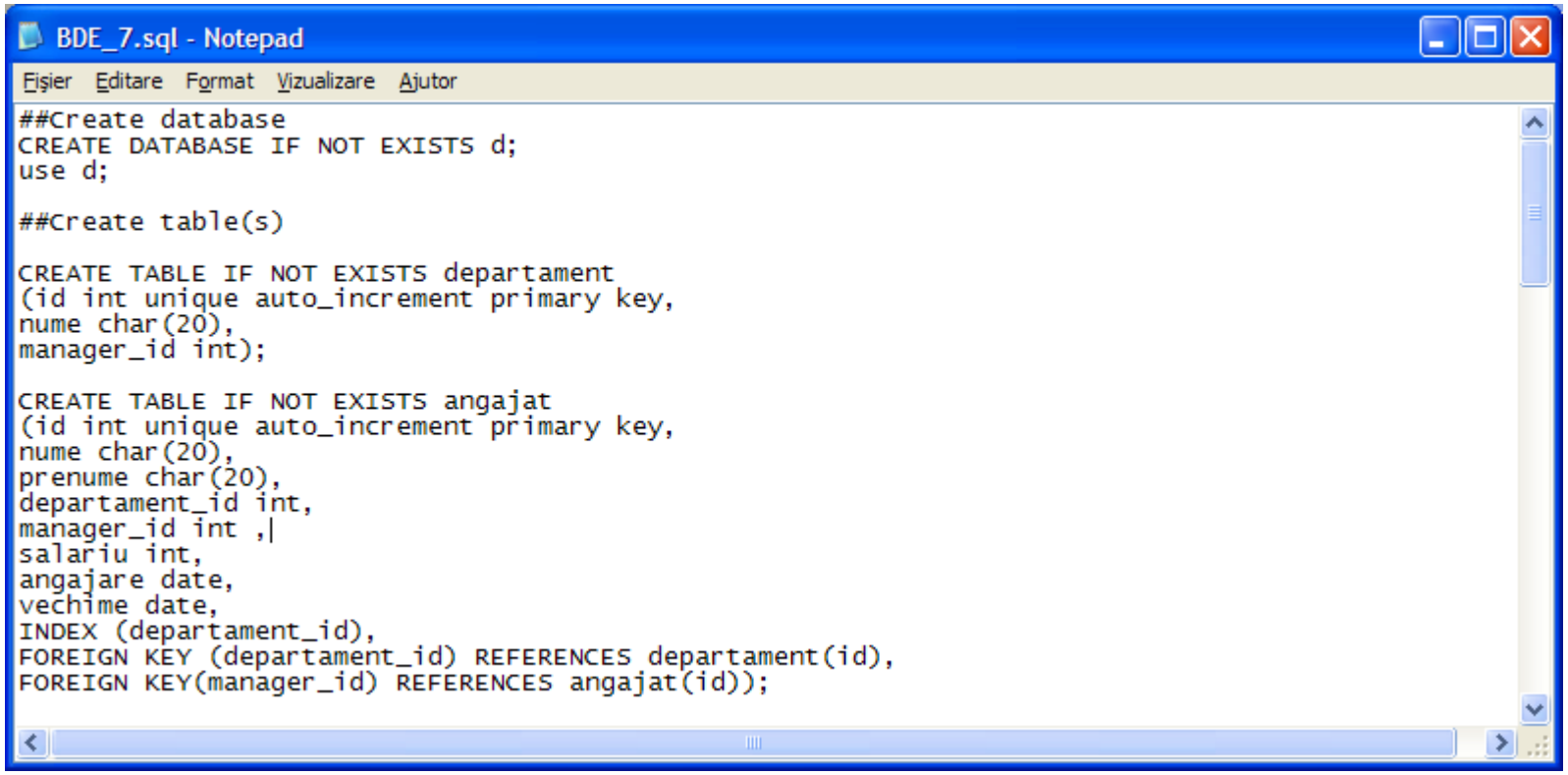
Nici o tabela selectata:



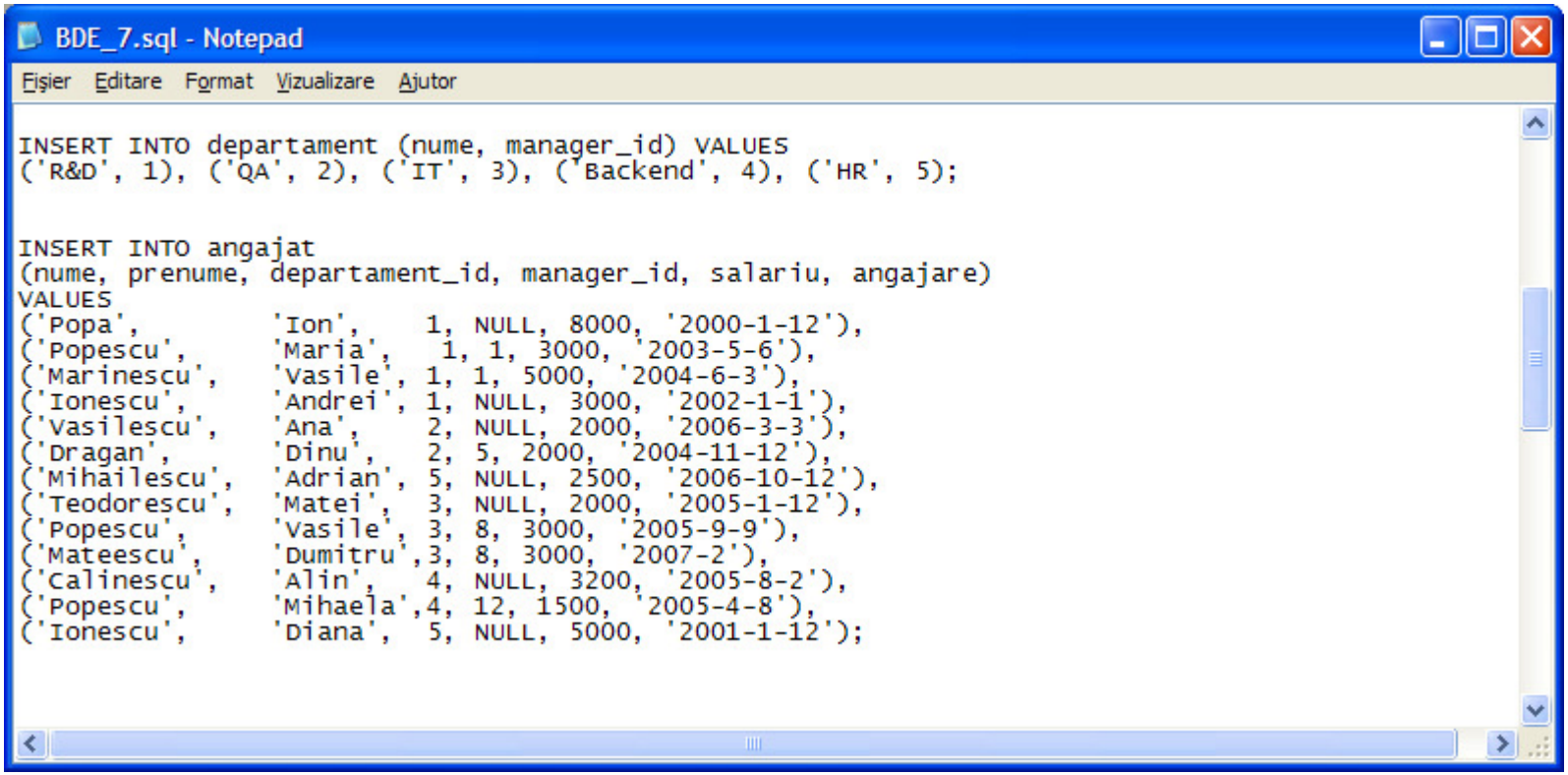

Se poate utiliza instructiunea SELECT fara a preciza nici o referinta la vreo tabela; se pot astfel calcula expresii aritmetice; pentru comoditate, se poate utiliza sintaxa incluzand referinta la tabela DUAL (nu trebuie definita).

# SQL

Exemple:


BDE_7.sql - Notepad
Fişier 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
Fişier 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');

# 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  | Vasile  |               1 |          1 |    5000 | 2004-06-03 | NULL    |
|  4 | Ionescu    | Andrei  |               1 |       NULL |    3000 | 2002-01-01 | NULL    |
|  5 | Vasilescu  | 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    | Vasile  |               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)
```

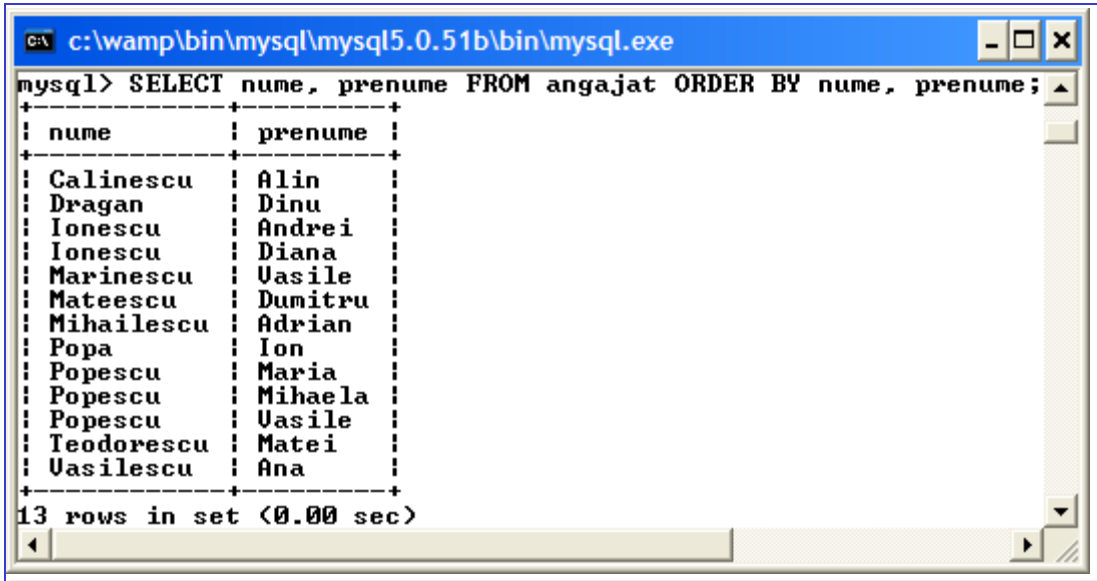
c:\wamp\bin\mysql\mysql5.0.51b\bin\...

```
mysql> SELECT nume, prenume FROM angajat
+------------+---------+
| nume       | prenume |
+------------+---------+
| Popa       | Ion     |
| Popescu    | Maria   |
| Marinescu  | Vasile  |
| Ionescu    | Andrei  |
| Vasilescu  | Ana     |
| Dragan     | Dinu    |
| Mihailescu | Adrian  |
| Teodorescu | Matei   |
| Popescu    | Vasile  |
| Mateescu   | Dumitru |
| Calinescu  | Alin    |
| Popescu    | Mihaela |
| Ionescu    | Diana   |
+------------+---------+
13 rows in set (0.00 sec)
```

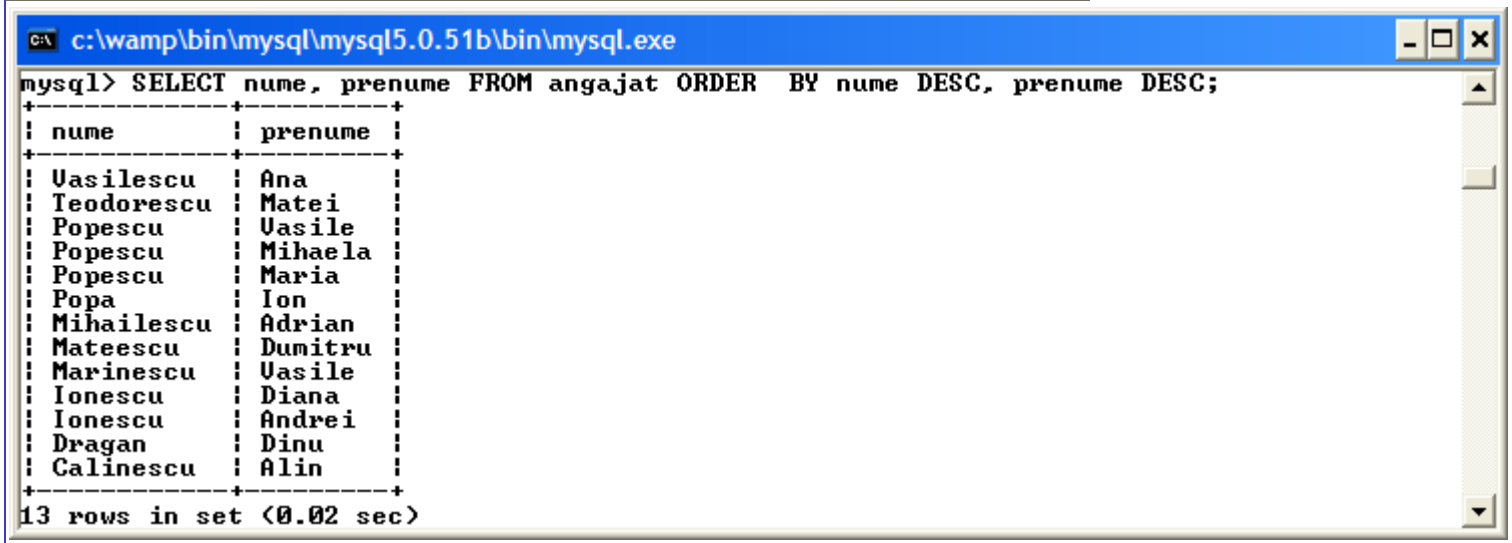
Clauza FROM precizeaza tabela (tabelele) din care se face selectia;

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

# SQL




Clauza ORDER BY se utilizeaza pentru a ordona (default e ascendent) dupa valoarea coloanelor precizate odata cu clauza ORDER BY. Ordonarea se faca 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.



# 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    | Vasile  |    3 |    3000 |
| Mateescu   | Dumitru |    3 |    3000 |
| Vasilescu  | Ana     |    2 |    2000 |
| Dragan     | Dinu    |    2 |    2000 |
| Ionescu    | Andrei  |    1 |    3000 |
| Popescu    | Maria   |    1 |    3000 |
| Marinescu  | Vasile  |    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...

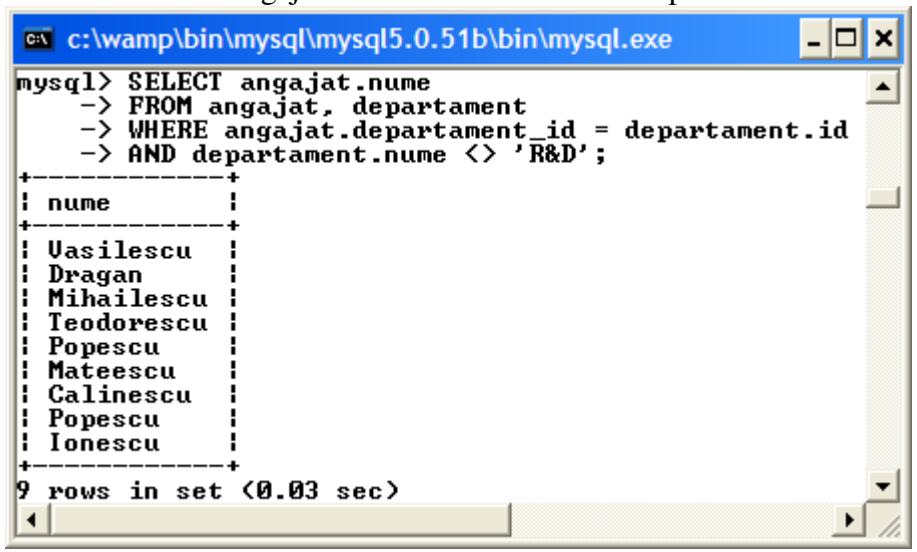
```
mysql> SELECT nume
    -> FROM angajat
    -> WHERE departament_id = 1;
+-----------+
| nume      |
+-----------+
| Popa      |
| Popescu   |
| Marinescu |
| Ionescu   |
+-----------+
4 rows in set (0.03 sec)
```

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)
```

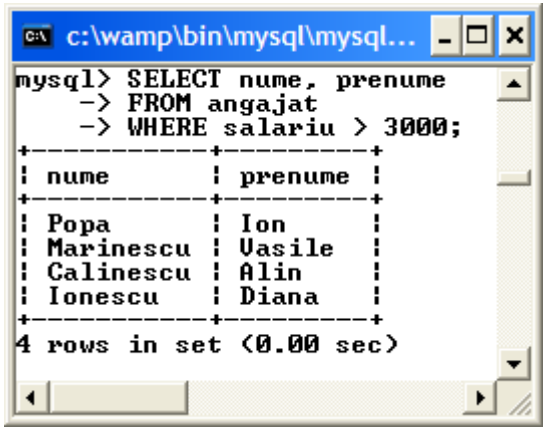
# SQL

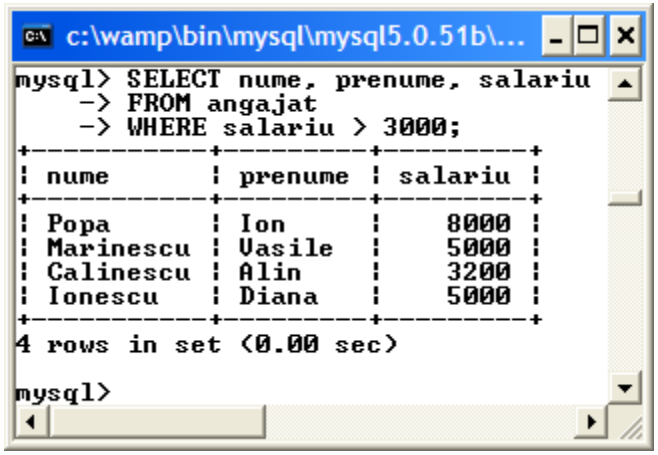
Selectati numele angajatilor care NU lucreaza la departamentul de R&D;




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.

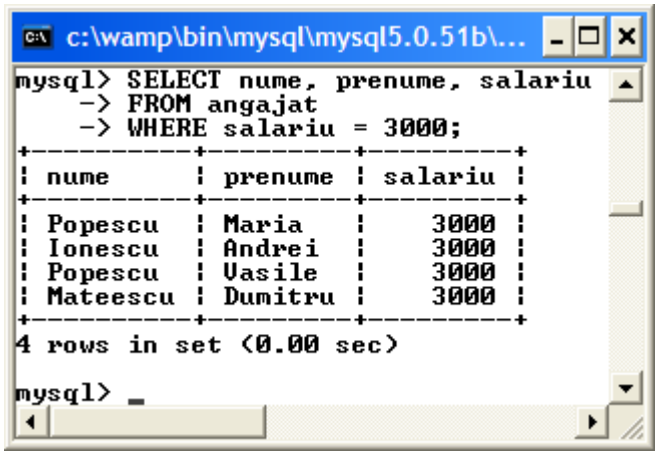
Angajatii care au salariu mai mare de 3000 lei



# SQL

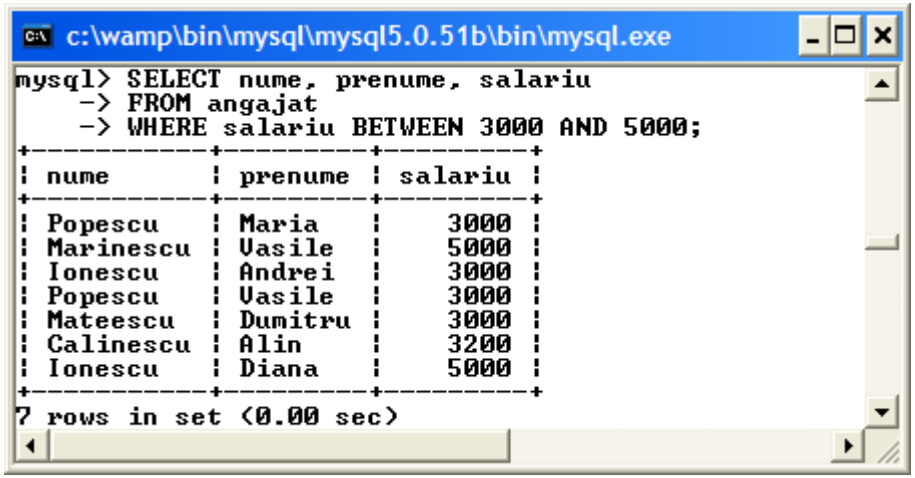
Angajatii care au salariu de 3000 lei




```
mysql> SELECT nume, prenume, salariu
    -> FROM angajat
    -> WHERE salariu = 3000;
+----------+---------+---------+
| nume     | prenume | salariu |
+----------+---------+---------+
| Popescu  | Maria   |    3000 |
| Ionescu  | Andrei  |    3000 |
| Popescu  | Vasile  |    3000 |
| Mateescu | Dumitru |    3000 |
+----------+---------+---------+
4 rows in set (0.00 sec)

mysql> _
```

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

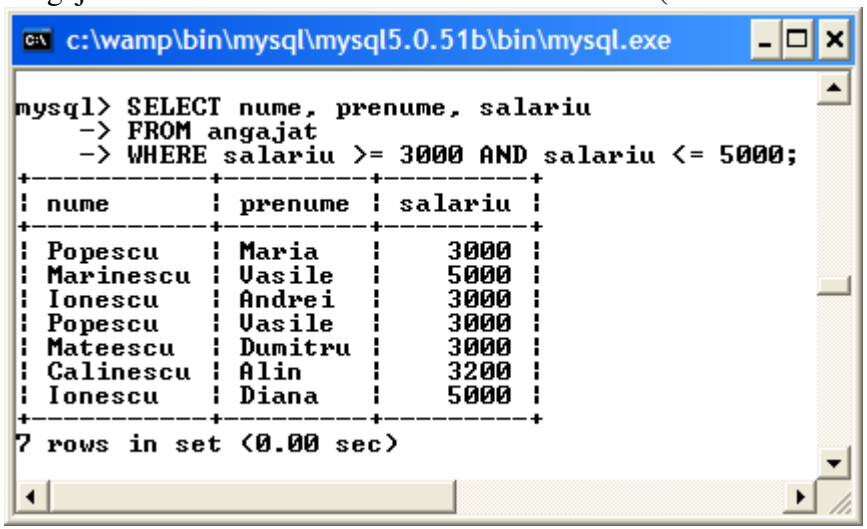



```
mysql> SELECT nume, prenume, salariu
    -> FROM angajat
    -> WHERE salariu BETWEEN 3000 AND 5000;
+-----------+---------+---------+
| nume      | prenume | salariu |
+-----------+---------+---------+
| Popescu   | Maria   |    3000 |
| Marinescu | Vasile  |    5000 |
| Ionescu   | Andrei  |    3000 |
| Popescu   | Vasile  |    3000 |
| Mateescu  | Dumitru |    3000 |
| Calinescu | Alin    |    3200 |
| Ionescu   | Diana   |    5000 |
+-----------+---------+---------+
7 rows in set (0.00 sec)
```

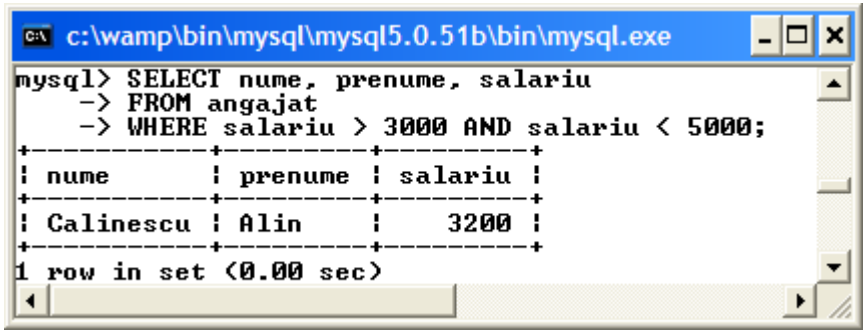
Clauza BETWEEN se utilizeaza pentru a preciza limitele de variatie ale unui atribut utilizat cu clauza WHERE; limitele sunt incluse;

# SQL

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

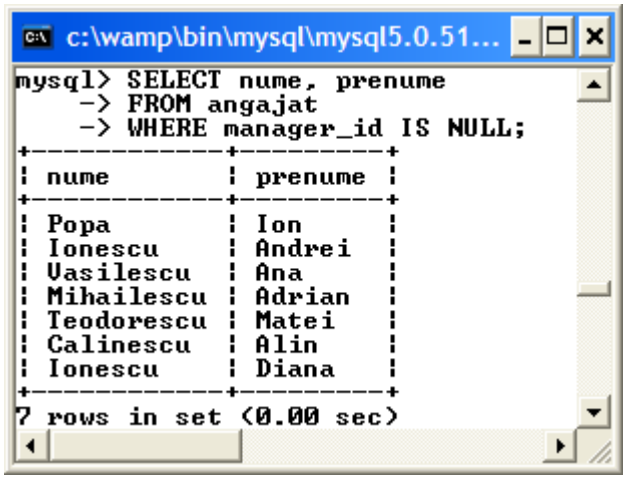



Angajatii care au salariul intre 3000 si 5000 de lei (excluzand limitele)

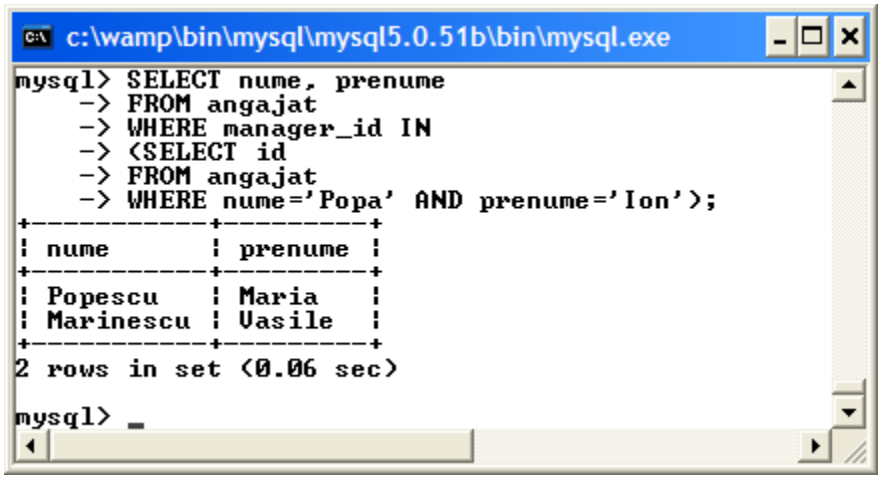


# SQL

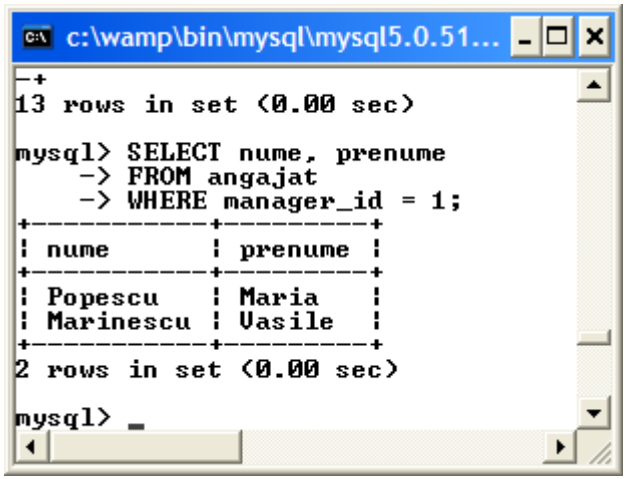
Angajatii care nu au manager




Angajatii ai caror manager este Popa Ion




Angajatii ai caror manager are ID-ul 1



# 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  | Vasile  |
| 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, salariu, manager_i
    -> FROM angajat
    -> WHERE salariu > 1000;
+------------+---------+---------+------------+
| nume       | prenume | salariu | manager_id |
+------------+---------+---------+------------+
| Popa       | Ion     |    8000 |       NULL |
| Popescu    | Maria   |    3000 |          1 |
| Marinescu  | Vasile  |    5000 |          1 |
| Ionescu    | Andrei  |    3000 |       NULL |
| Vasilescu  | Ana     |    2000 |       NULL |
| Dragan     | Dinu    |    2000 |          5 |
| Mihailescu | Adrian  |    2500 |       NULL |
| Teodorescu | Matei   |    2000 |       NULL |
| Popescu    | Vasile  |    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)
```

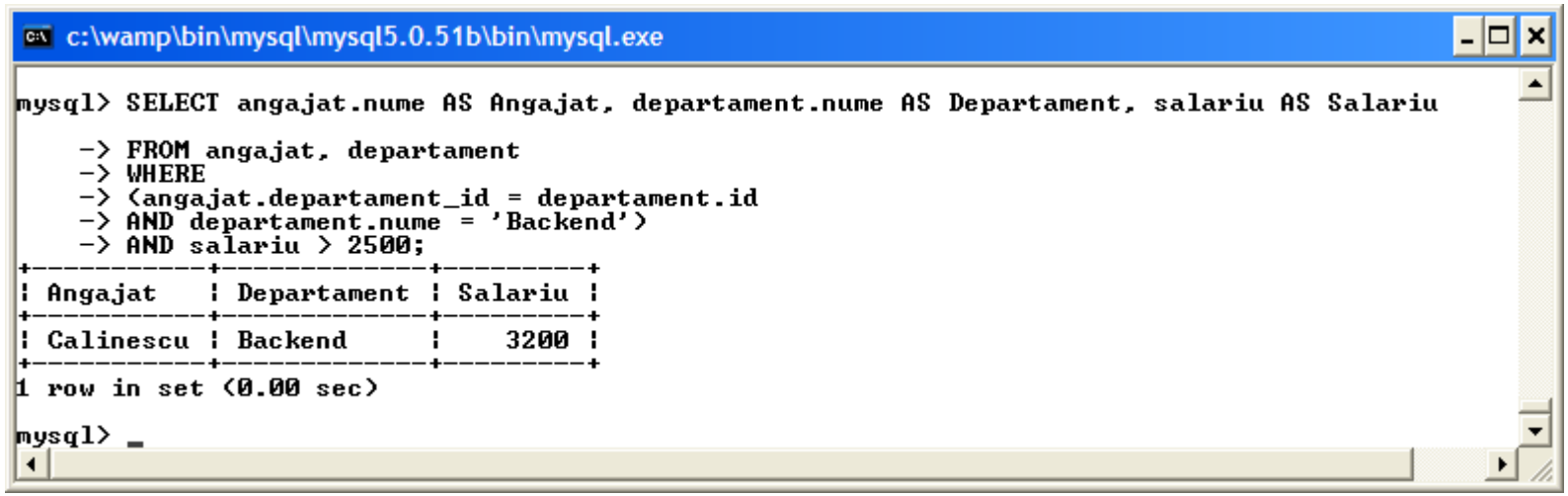
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

```
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  |
| Vasilescu  | Ana     |
| Dragan     | Dinu    |
| Mihailescu | Adrian  |
| Teodorescu | Matei   |
| Popescu    | Vasile  |
| Mateescu   | Dumitru |
| Calinescu  | Alin    |
| Popescu    | Mihaela |
| Ionescu    | Diana   |
+------------+---------+
11 rows in set (0.00 sec)

mysql> _
```

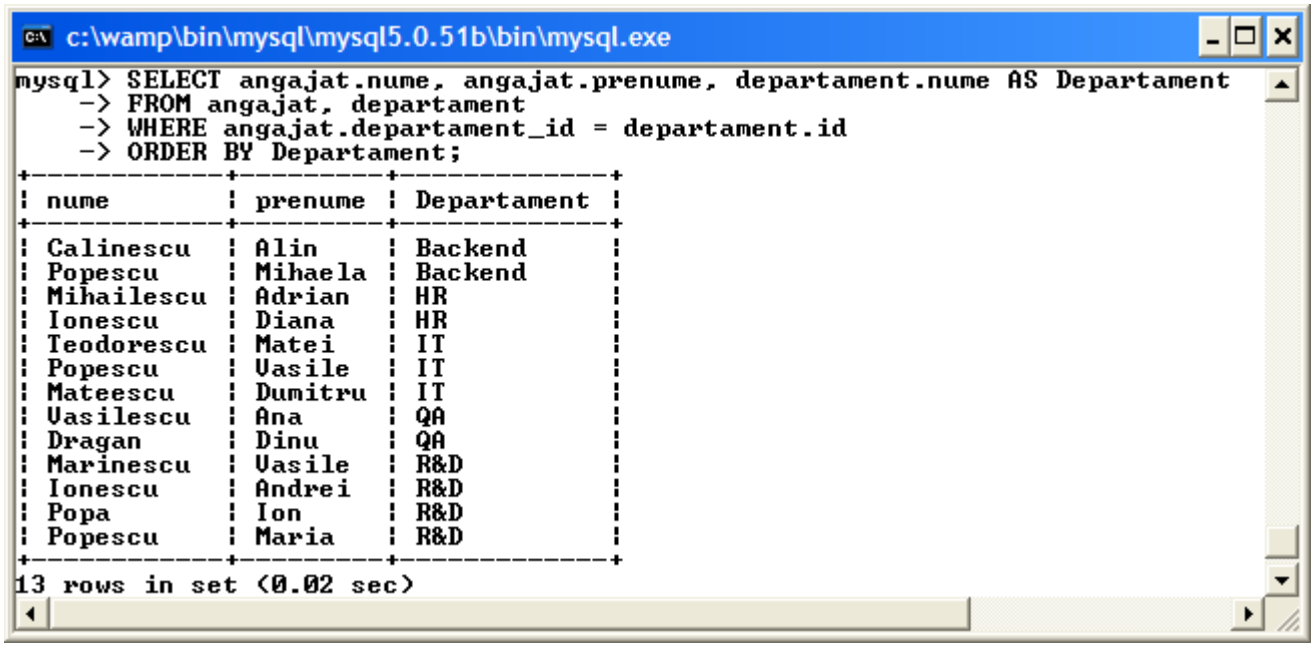
# SQL

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



# SQL

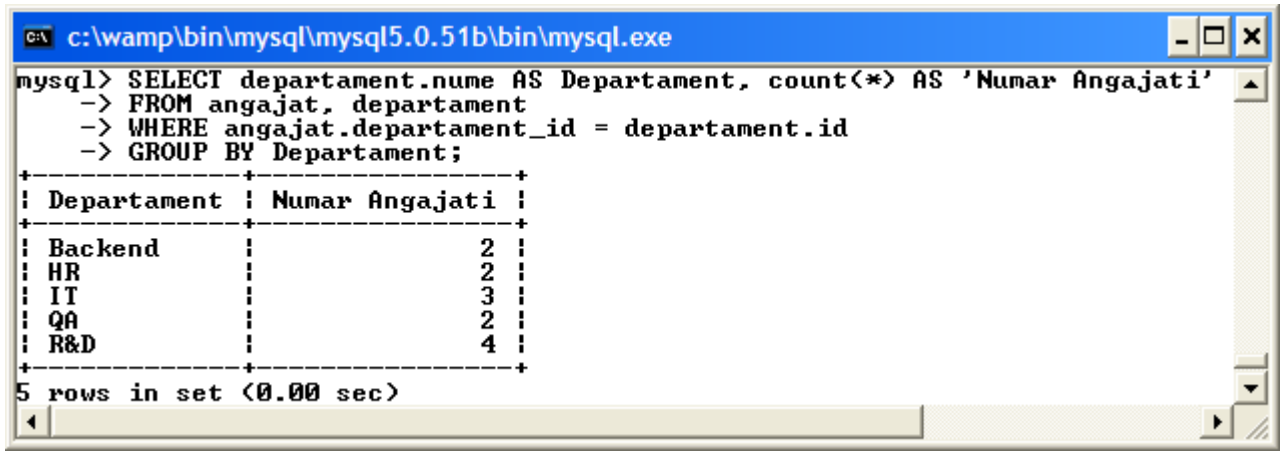
Angajatii, ordonati dupa departamente



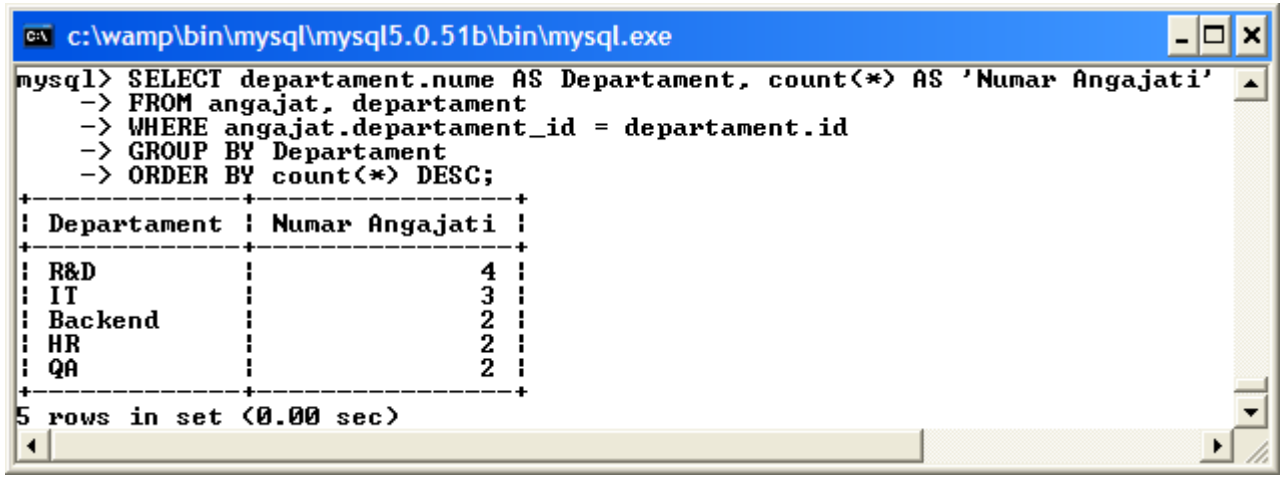

Cand se utilizeaza atat clauza WHERE cat si clauza ORDER BY, WHERE apare intotdeauna inaintea lui ORDER BY.

# SQL

Numarul de angajati de la fiecare departament




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



# SQL

Angajatii, ordonati dupa vechime