Limbajul de interogare a datelor – functii agregate

- COUNT
- MAX, MIN
- SUM
- AVG, VARIANCE
- GROUP BY
- HAVING

Limbajul de interogare a datelor – functii agregate

Functiile agregate: functii care returneaza o singura inregistrare pentru operatii efectuate asupra unui grup de inregistrari.

COUNT, MIN, MAX, SUM, AVG, VARIANCE

Se folosesc impreuna cu:

GROUP BY, HAVING

```
SELECT set_select_expr, ...
  [FROM table_references
  [WHERE where_condition]
  [GROUP BY {col_name | expr | position}
  [HAVING where_condition]
```

Functiile agregate care accepta un singur argument accepta si clauzele: DISTINCT si UNIQUE; Aceste clauze limiteaza la o singura instanta fiecare valoare distincta in lista valorilor asupra carora se aplica functia agregata; Clauza ALL forteaza functiile agregate sa ia in considerare toate valorile din setul asupra caruia se aplica functia.

SQL

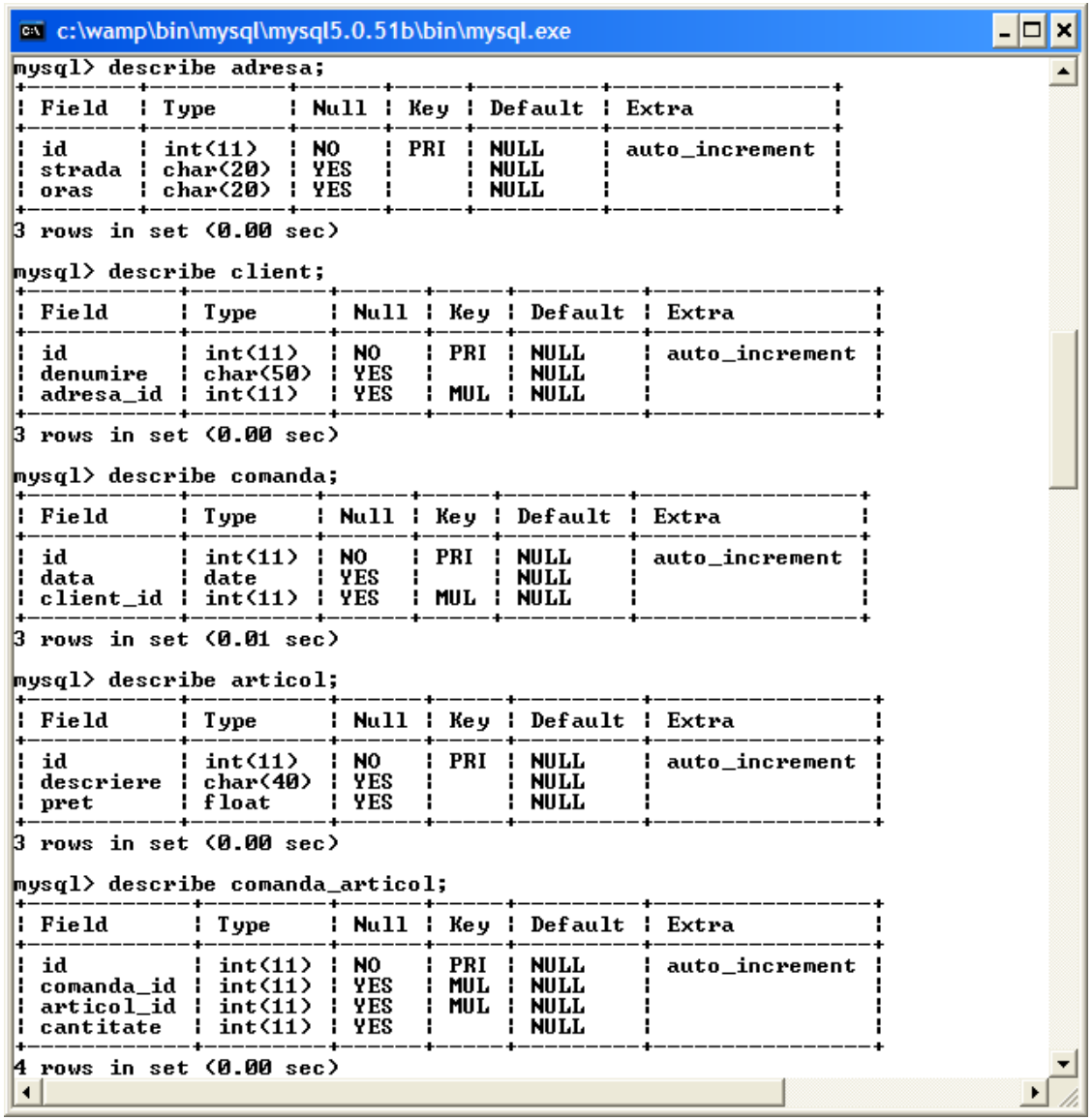
Creare structura date:

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

```
mysql> CREATE TABLE IF NOT EXISTS adresa
    -> (id int unique auto_increment primary key,
    -> strada char(20),
    -> oras char(20));
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> CREATE TABLE IF NOT EXISTS client
    -> (id int unique auto_increment primary key,
    -> denumire char(50),
    -> adresa_id int,
    -> INDEX (adresa_id),
    -> FOREIGN KEY (adresa_id) REFERENCES adresa(id));
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql>  CREATE TABLE IF NOT EXISTS comanda
    -> (id int unique auto_increment primary key,
    -> data date,
    -> client_id int,
    -> INDEX (client_id),
    -> FOREIGN KEY (client_id) REFERENCES client(id));
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql>
mysql>
mysql>  CREATE TABLE IF NOT EXISTS articol
    -> (id int unique auto_increment primary key,
    -> descriere char(40),
    -> pret  float);
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql>  CREATE TABLE IF NOT EXISTS comanda_articol
    -> (id int unique auto_increment primary key,
    -> comanda_id int,
    -> articol_id int,
    -> cantitate int,
    -> INDEX (comanda_id),
    -> INDEX (articol_id),
    ->
    -> FOREIGN KEY (comanda_id) REFERENCES comanda(id),
    -> FOREIGN KEY (articol_id) REFERENCES articol(id));
Query OK, 0 rows affected (0.00 sec)
```

SQL

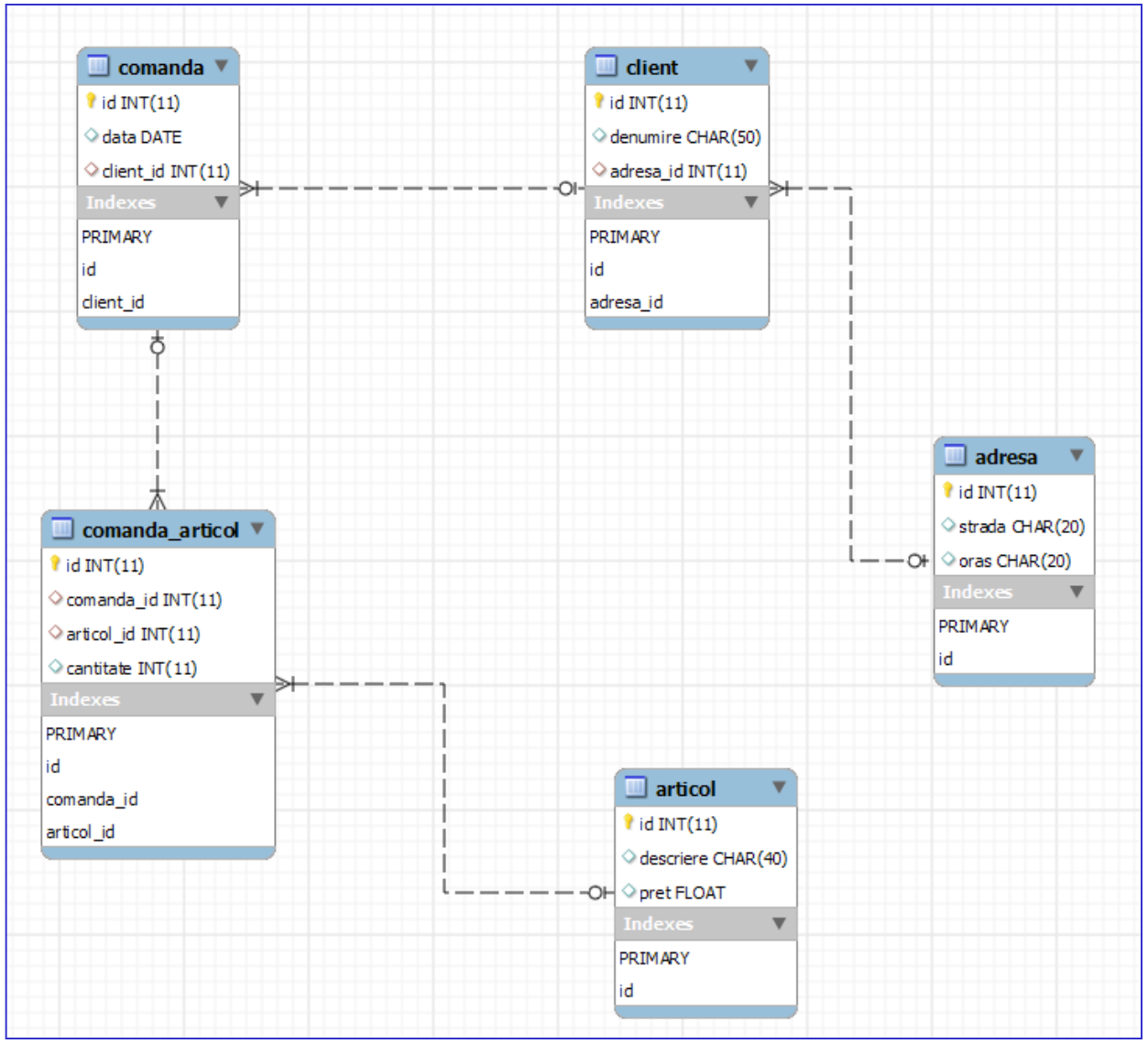
Structura rezultata:




```
mysql> describe adresa;
+---------+----------+------+-----+---------+----------------+
| Field   | Type     | Null | Key | Default | Extra          |
+---------+----------+------+-----+---------+----------------+
| id      | int(11)  | NO   | PRI | NULL    | auto_increment |
| strada  | char(20) | YES  |     | NULL    |                |
| oras    | char(20) | YES  |     | NULL    |                |
+---------+----------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> describe client;
+------------+----------+------+-----+---------+----------------+
| Field      | Type     | Null | Key | Default | Extra          |
+------------+----------+------+-----+---------+----------------+
| id         | int(11)  | NO   | PRI | NULL    | auto_increment |
| denumire   | char(50) | YES  |     | NULL    |                |
| adresa_id  | int(11)  | YES  | MUL | NULL    |                |
+------------+----------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> describe comanda;
+-----------+---------+------+-----+---------+----------------+
| Field     | Type    | Null | Key | Default | Extra          |
+-----------+---------+------+-----+---------+----------------+
| id        | int(11) | NO   | PRI | NULL    | auto_increment |
| data      | date    | YES  |     | NULL    |                |
| client_id | int(11) | YES  | MUL | NULL    |                |
+-----------+---------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql> describe articol;
+----------+----------+------+-----+---------+----------------+
| Field    | Type     | Null | Key | Default | Extra          |
+----------+----------+------+-----+---------+----------------+
| id       | int(11)  | NO   | PRI | NULL    | auto_increment |
| descriere| char(40) | YES  |     | NULL    |                |
| pret     | float    | YES  |     | NULL    |                |
+----------+----------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> describe comanda_articol;
+-----------+---------+------+-----+---------+----------------+
| Field     | Type    | Null | Key | Default | Extra          |
+-----------+---------+------+-----+---------+----------------+
| id        | int(11) | NO   | PRI | NULL    | auto_increment |
| comanda_id| int(11) | YES  | MUL | NULL    |                |
| articol_id| int(11) | YES  | MUL | NULL    |                |
| cantitate | int(11) | YES  |     | NULL    |                |
+-----------+---------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

SQL

Structura rezultata:



SQL

Inseram cateva date:

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

```
mysql> INSERT INTO adresa (strada, oras) VALUES
    -> ('Magheru', 'Bucuresti'),
    -> ('Brasov',  'Bucuresti'),
    -> ('Elisabeta', 'Bucuresti'),
    -> ('Maniu', 'Brasov'),
    -> ('Aviatorilor', 'Bucuresti'),
    -> ('Balcescu', 'Buzau'),
    -> ('Mihalache', 'Bucuresti'),
    -> ('Horea', 'Cluj Napoca'),
    -> ('Macaralei', 'Barlad'),
    -> ('Minerului', 'Petrosani');
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO client (denumire, adresa_id) VALUES
    -> ('SC LEON SRL', 1),
    -> ('GEOMIL', 2),
    -> ('LIBERTY', 3),
    -> ('MINERON', 10),
    -> ('AEROFUN', 4),
    -> ('MEDITEL', 5);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> INSERT INTO comanda (data, client_id) VALUES
    -> ('2007-4-8', 1),
    -> ('2007-4-8', 1),
    -> ('2007-4-7', 2),
    -> ('2008-4-8', 2),
    -> ('2008-4-9', 3),
    -> ('2008-4-8', 4),
    -> ('2008-4-8', 5);
Query OK, 7 rows affected (0.02 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> INSERT INTO articol (descriere, pret) VALUES
    -> ('stilou', 5),
    -> ('pix bila', 4),
    -> ('creion', 0.5),
    -> ('marker', 2),
    -> ('guma', 0.75),
    -> ('rigla', 1),
    -> ('echer', 1),
    -> ('penar', 2),
    -> ('top hartie', 9.5),
    -> ('top hartie gloss', 9.5),
    -> ('cutie cerneala', 2.25);
Query OK, 11 rows affected (0.02 sec)
Records: 11  Duplicates: 0  Warnings: 0
```

SQL

Inseram cateva date

(continuare):

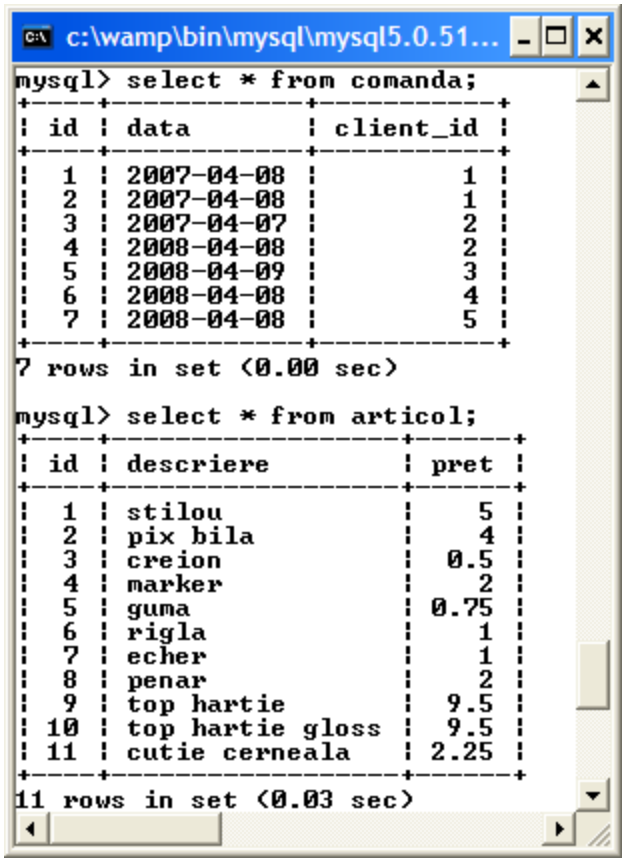
```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> INSERT INTO comanda_articol (comanda_id, articol_id, cantitate) VAL
    -> 
    -> (1,1,10),
    -> (1,2,10),
    -> (1,3,10),
    -> 
    -> 
    -> (2,4,2),
    -> (2,5,4),
    -> (2,6,12),
    -> 
    -> (3,8,20),
    -> (3,5,42),
    -> 
    -> 
    -> (4,1,20),
    -> (4,2,40),
    -> (4,3,20),
    -> (4,4,40),
    -> (4,5,20),
    -> (4,6,20),
    -> 
    -> (5,1,100),
    -> (5,7,200),
    -> 
    -> (6,2,25),
    -> (6,6,25),
    -> 
    -> (7,3,100);
Query OK, 19 rows affected (0.01 sec)
Records: 19  Duplicates: 0  Warnings: 0
```

Tabelele populate cu date:

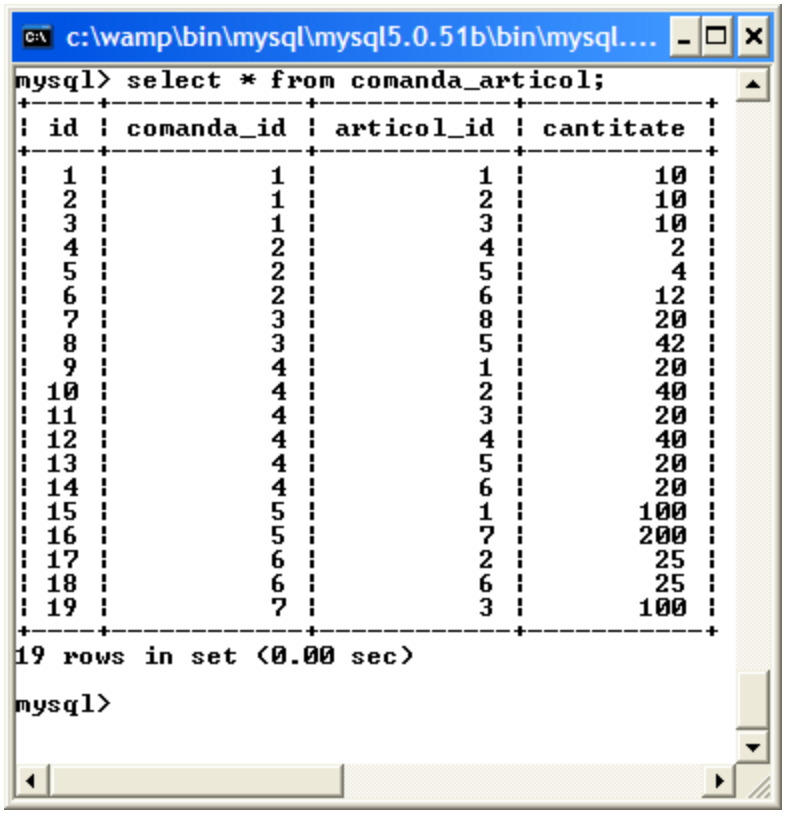
```
c:\wamp\bin\mysql\mysql5.0.51...
mysql> select * from adresa;
+----+-------------+-------------+
| id | strada      | oras        |
+----+-------------+-------------+
|  1 | Magheru     | Bucuresti   |
|  2 | Brasov      | Bucuresti   |
|  3 | Elisabeta   | Bucuresti   |
|  4 | Maniu       | Brasov      |
|  5 | Aviatorilor | Bucuresti   |
|  6 | Balcescu    | Buzau       |
|  7 | Mihalache   | Bucuresti   |
|  8 | Horea       | Cluj Napoca |
|  9 | Macaralei   | Barlad      |
| 10 | Minerului   | Petrosani   |
+----+-------------+-------------+
10 rows in set (0.00 sec)

mysql> select * from client;
+----+-------------+------------+
| id | denumire    | adresa_id  |
+----+-------------+------------+
|  1 | SC LEON SRL |          1 |
|  2 | GEOMIL      |          2 |
|  3 | LIBERTY     |          3 |
|  4 | MINERON     |         10 |
|  5 | AEROFUN     |          4 |
|  6 | MEDITEL     |          5 |
+----+-------------+------------+
6 rows in set (0.00 sec)

mysql> _
```

SQL

Tabelele populate cu date:



```
c:\wamp\bin\mysql\mysql5.0.51...
mysql> select * from comanda;
+----+------------+-----------+
| id | data       | client_id |
+----+------------+-----------+
|  1 | 2007-04-08 |         1 |
|  2 | 2007-04-08 |         1 |
|  3 | 2007-04-07 |         2 |
|  4 | 2008-04-08 |         2 |
|  5 | 2008-04-09 |         3 |
|  6 | 2008-04-08 |         4 |
|  7 | 2008-04-08 |         5 |
+----+------------+-----------+
7 rows in set (0.00 sec)

mysql> select * from articol;
+----+-----------------+------+
| id | descriere       | pret |
+----+-----------------+------+
|  1 | stilou          |    5 |
|  2 | pix bila        |    4 |
|  3 | creion          |  0.5 |
|  4 | marker          |    2 |
|  5 | guma            | 0.75 |
|  6 | rigla           |    1 |
|  7 | echer           |    1 |
|  8 | penar           |    2 |
|  9 | top hartie      |  9.5 |
| 10 | top hartie gloss |  9.5 |
| 11 | cutie cerneala  | 2.25 |
+----+-----------------+------+
11 rows in set (0.03 sec)
```



```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql....
mysql> select * from comanda_articol;
+----+------------+------------+----------+
| id | comanda_id | articol_id | cantitate |
+----+------------+------------+----------+
|  1 |          1 |          1 |       10 |
|  2 |          1 |          2 |       10 |
|  3 |          1 |          3 |       10 |
|  4 |          2 |          4 |        2 |
|  5 |          2 |          5 |        4 |
|  6 |          2 |          6 |       12 |
|  7 |          3 |          8 |       20 |
|  8 |          3 |          5 |       42 |
|  9 |          4 |          1 |       20 |
| 10 |          4 |          2 |       40 |
| 11 |          4 |          3 |       20 |
| 12 |          4 |          4 |       40 |
| 13 |          4 |          5 |       20 |
| 14 |          4 |          6 |       20 |
| 15 |          5 |          1 |      100 |
| 16 |          5 |          7 |      200 |
| 17 |          6 |          2 |       25 |
| 18 |          6 |          6 |       25 |
| 19 |          7 |          3 |      100 |
+----+------------+------------+----------+
19 rows in set (0.00 sec)

mysql>
```

COUNT

(1) Calculati numarul de comenzi

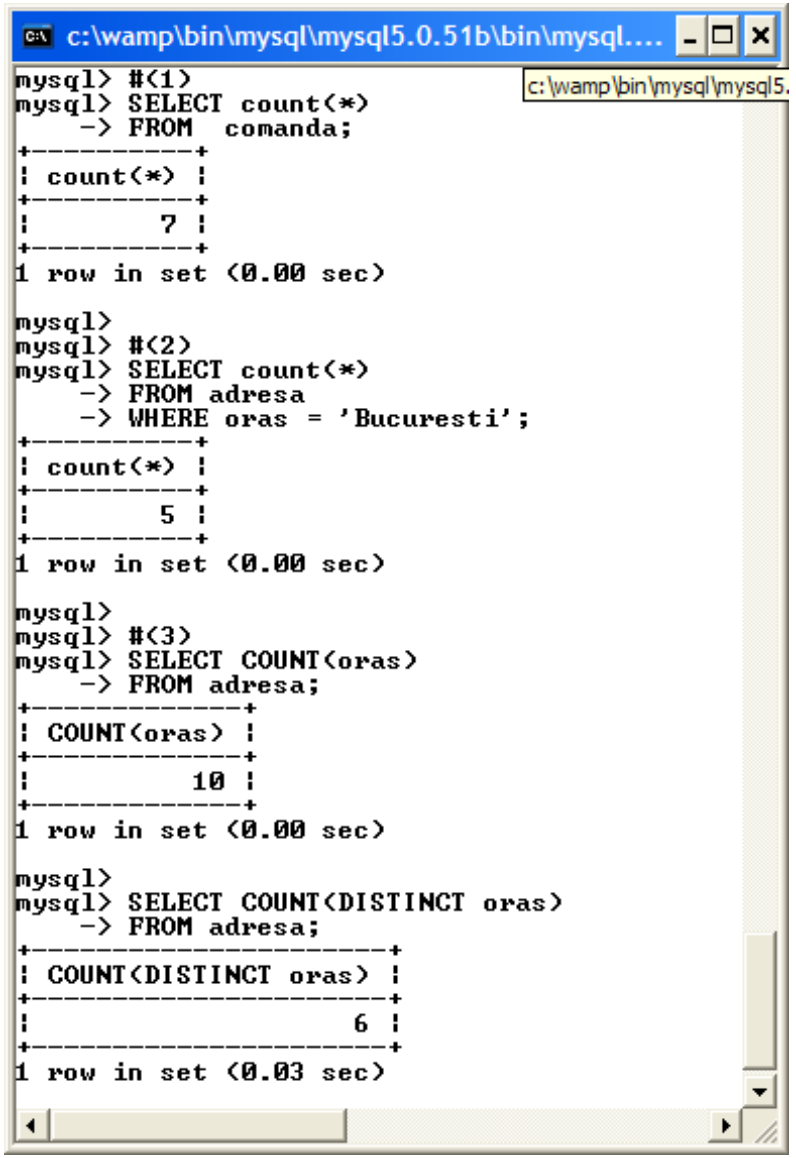
(2) Calculati numarul de adrese ale clientilor din Bucuresti

(3) Calculati numarul de orase diferite din toate adresele

(1) Se utilizeaza functia COUNT

(2) Se utilizeaza functia COUNT si se filtreaza doar acele adrese pentru care orasul este 'Bucuresti'

(3) Pentru a se numara toate inregistrarile din tabela 'adresa', se foloseste functia COUNT simpla; pentru a numara toate orasele, fara redundante, se utilizeaza clauza DISTINCT (al doilea exemplu).



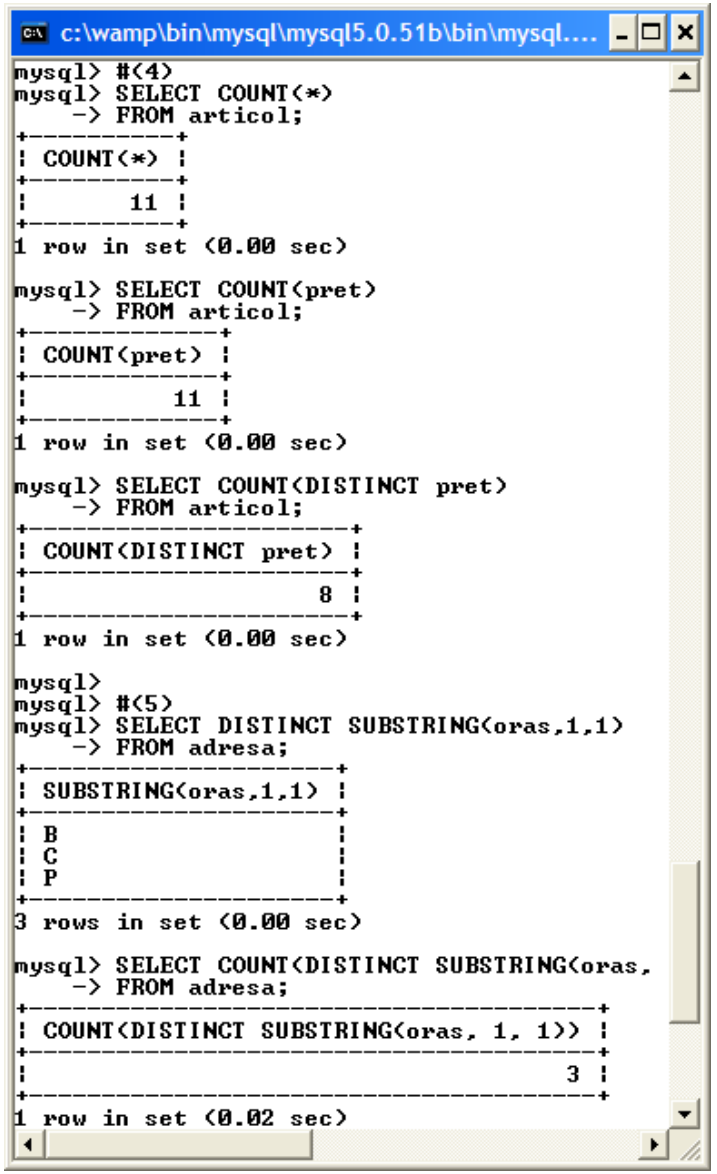
SQL

(4) Calculati numarul de preturi deosebite dintre toate articolele

(5) Calculati numarul de orase din toate adresele al caror nume incepe cu litere diferite

(4) Folosim functia COUNT, fie cu argumentul '*', fie cu argumentul 'pret'; rezultatul este acelasi si insumeaza toate inregistrarile din tabela de articole, indiferent daca sunt mai multe articole cu acelasi pret; pentru a obtine numarul de preturi diferite din tabela (pentru ca pot exista articole cu acelasi pret), se utilizeaza clauza DISTINCT.

(5) Se selecteaza mai intai initialele oraselor (exemplul intai); se extrage prima litera din coloana 'oras'. Pentru a se numara instantele, distincte (utilizand clauza DISTINCT) ale initialelor, se aplica functia COUNT



MAX, MIN

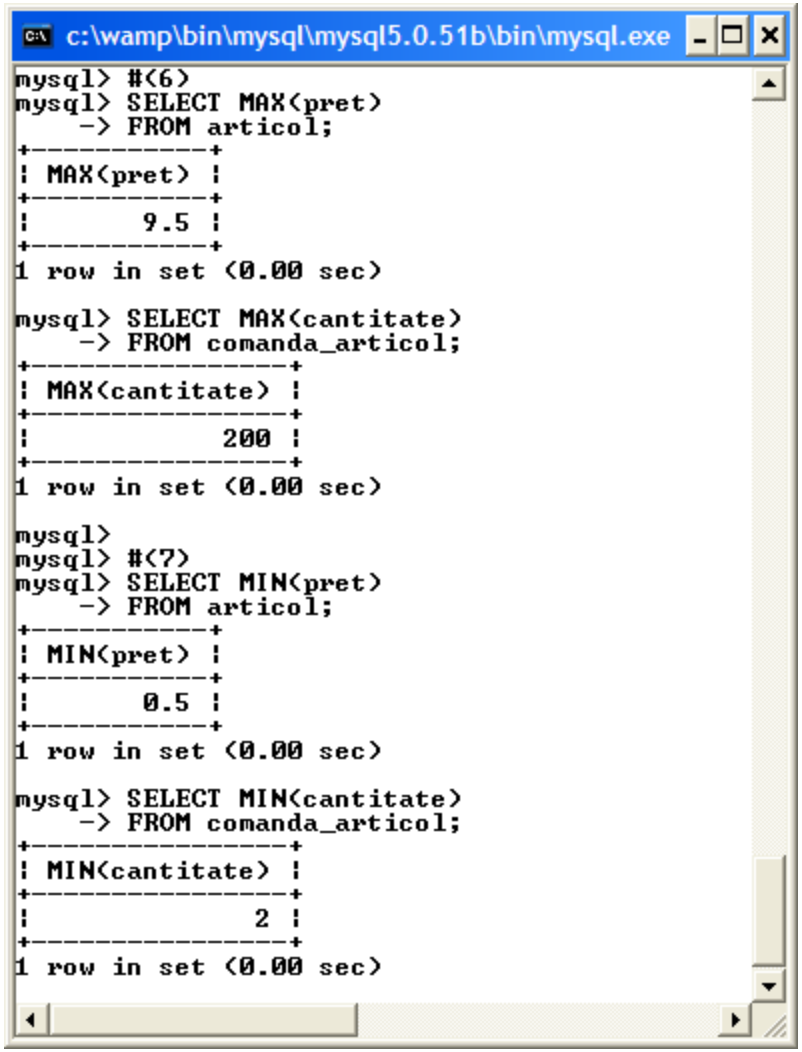
(6) Care este pretul maxim al articolelor; care este cantitatea maxima de articole dintr-o comanda

(7) Care este pretul minim al articolelor; care e cantitatea minima de articole dintr-o comanda

Fiecare articol are un pret, precizat in tabela ‘articol’; o comanda poate avea mai multe articole, fiecare articol fiind livrat intr-o anume cantitate;

(6) Utilizam functia MAX

(7) Utilizam functia MIN;

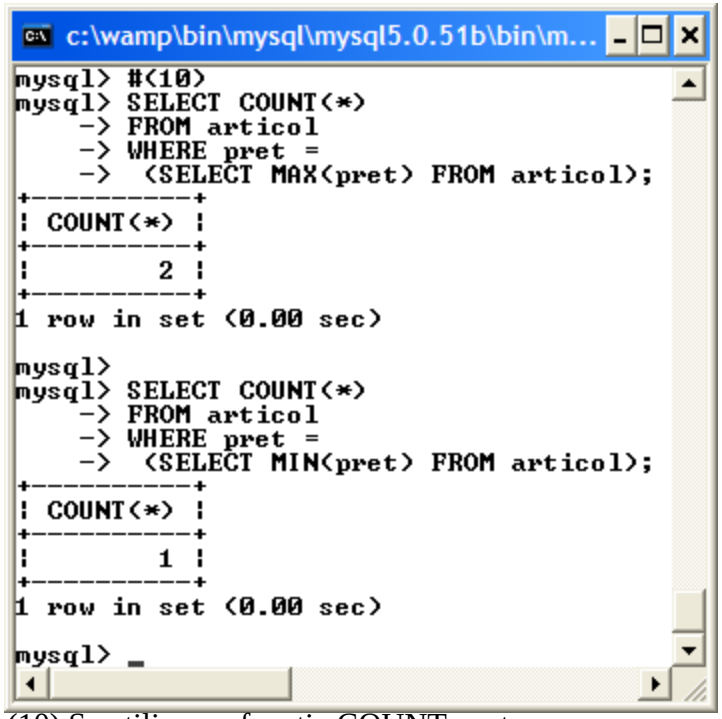


SQL

(8) Care este pretul minim al articolelor al caror nume incepe cu 'p' ?

(9) Care este pretul maxim al articolelor din comanda numarul 3 ?

(10) Cate articole au cel mai mare pret ? Dar cel mai mic pret ?

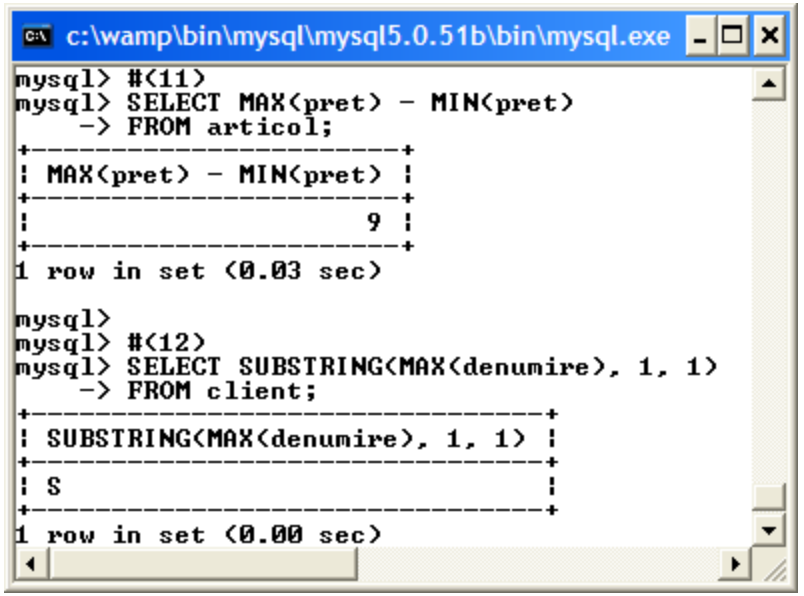


(10) Se utilizeaza functia COUNT pentru a numara inregistrarile pentru care pretul este pretul minim/maxim (calculat intr-o subquery, folosind functia MIN/ MAX)

SQL

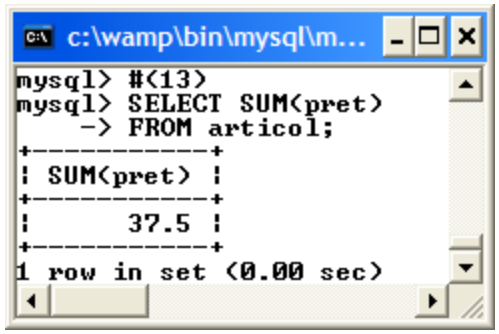
(11) Care este diferenta intre cel mai mare si mai mic pret (in bani) ?

(12) Care este initiala clientului cu ultimul nume, in ordine alfabetica ?



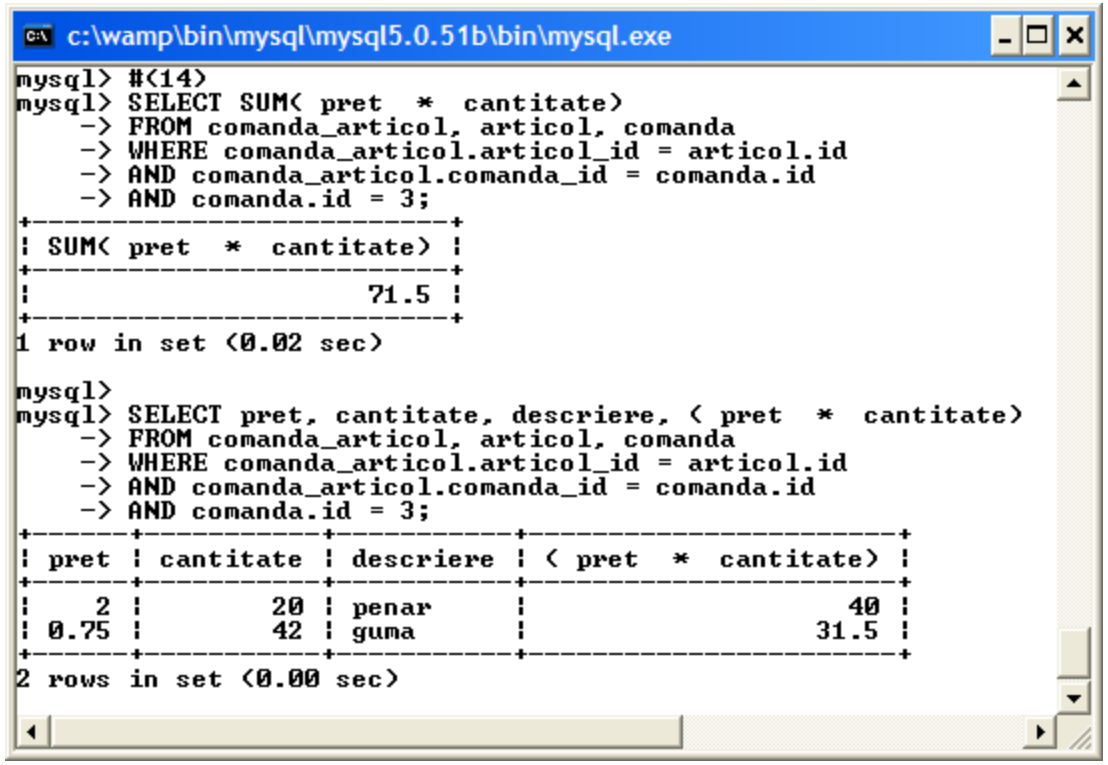
SUM

(13) Care este suma preturilor articolelor ?



SQL

(14) Care este pretul total al comenzii cu numarul 3 ?



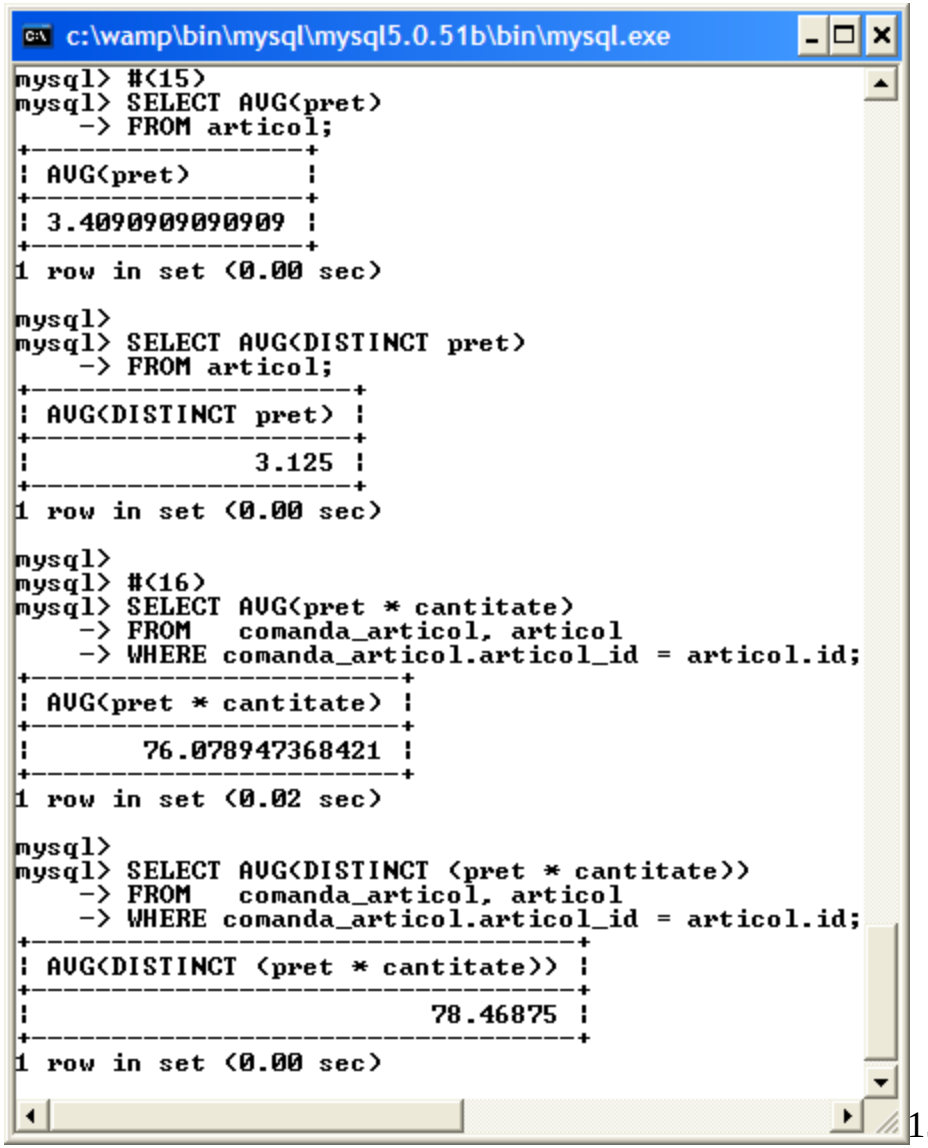
Pretul total al unei comenzi se calculeaza ca suma produselor *pret articol* × *cantitate*, pentru toate articolele dintr-o comanda; pe langa rezultatul cerut, se listeaza si informatia totala despre comanda selectata (comanda numarul 3), cu pretul articolelor, cantitatea articolelor din comanda, descrierea fiecarui articol si pretul total per articol (pretul intregii cantitati).

SQL

(15) Calculati pretul mediu al articolelor. Calculati pretul mediu al articolelor, neponderat.

(16) Calculati pretul mediu al comenzilor. Calculati pretul mediu al comenzilor, neponderat.

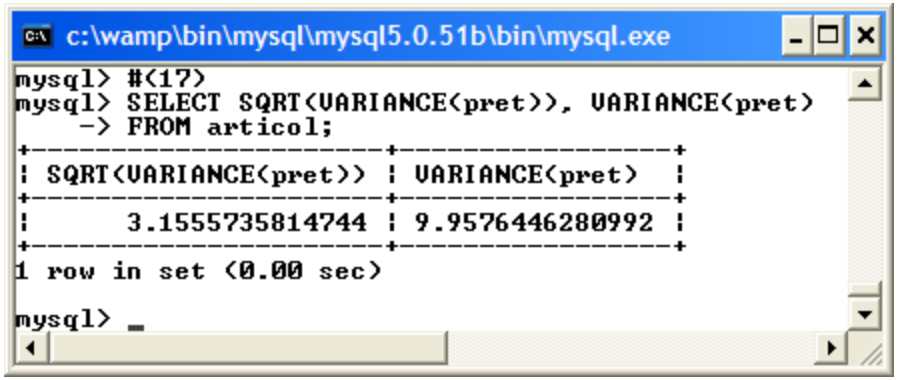
Calculand media preturilor cu functia AVG, se calculeaza de fapt o medie ponderata: acele preturi care apar de mai multe ori in tabela de articole vor avea o pondere mai mare, proportionala cu numarul de ocurente; pentru a calcula media neponderata, se utilizeaza clauza DISTINCT.



VARIANCE: varianță, deviație standard

(17) Calculati varianta preturilor articolelor; calculati deviatia standard a preturilor articolelor.

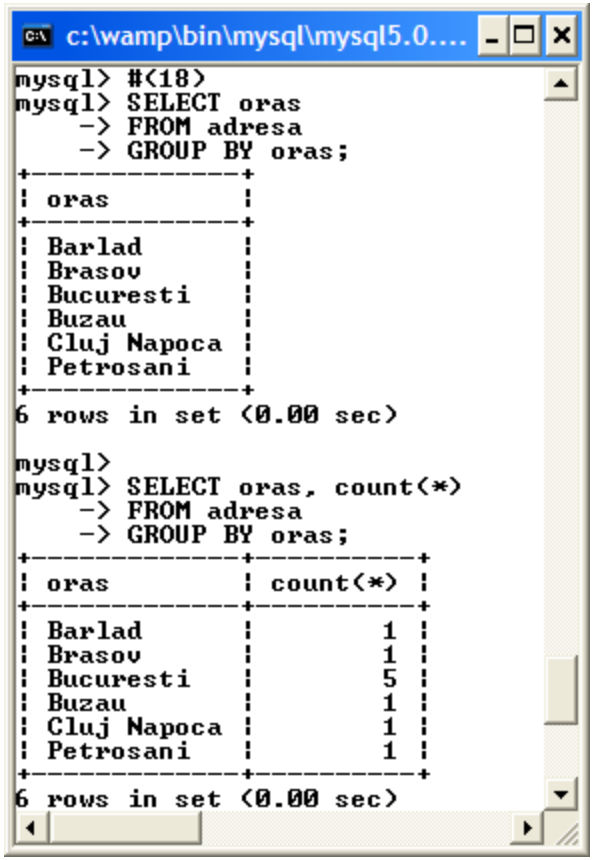
In cazul in care un limbaj are suport pentru functia ‘Standard deviation’, se utilizeaza acesta functie; daca nu, se poate utiliza in schimb formula:
StDev = Sqrt(variance)

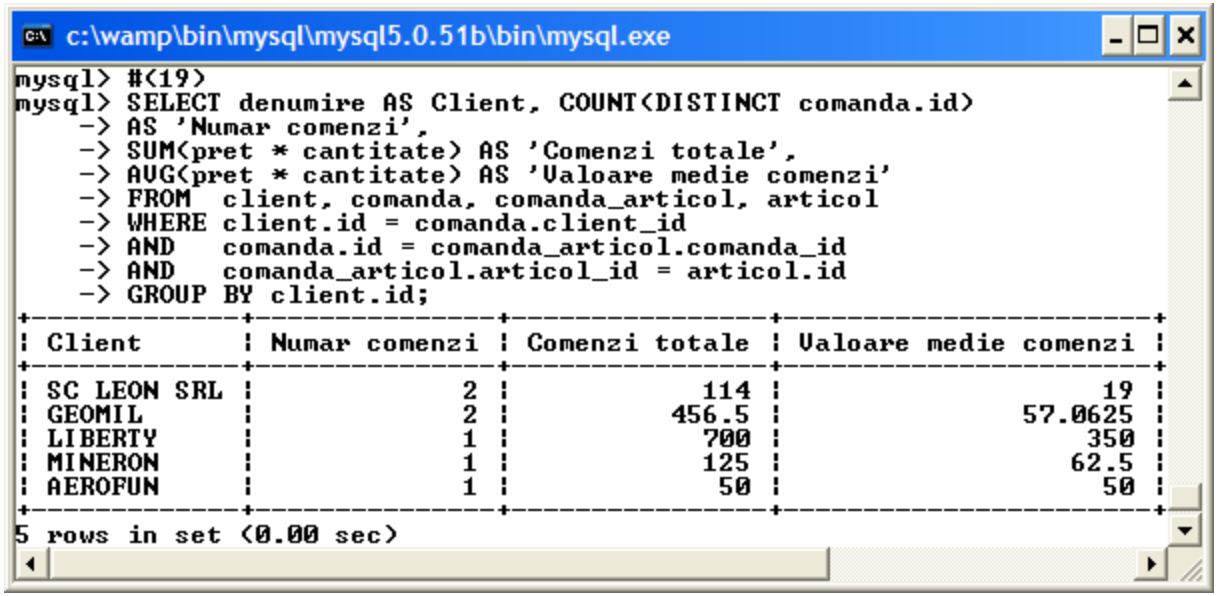


GROUP BY

(18) Listati orasele care apar in adrese; Listati orasele care apar in adresele clientilor, cu numarul corespunzator de adrese distincte;

(19) Listati numarul de comenzi pentru fiecare client, cu suma totala si media valorii comenzilor.





(18) Se foloseste clauza GROUP BY pentru a selecta orasele DISTINCTE care apar in tabela de adrese; pentru a socoti numarul de aparitii ale fiecarui oras in tabela de adrese, se foloseste functia COUNT in combinatie cu clauza GROUP BY

(19) Se grupeaza dupa codul clientului folosind clauza GROUP BY; se calculeaza numarul de comenzi per client utilizand functia COUNT si clauza DISTINCT aplicata id-ului comenzii.

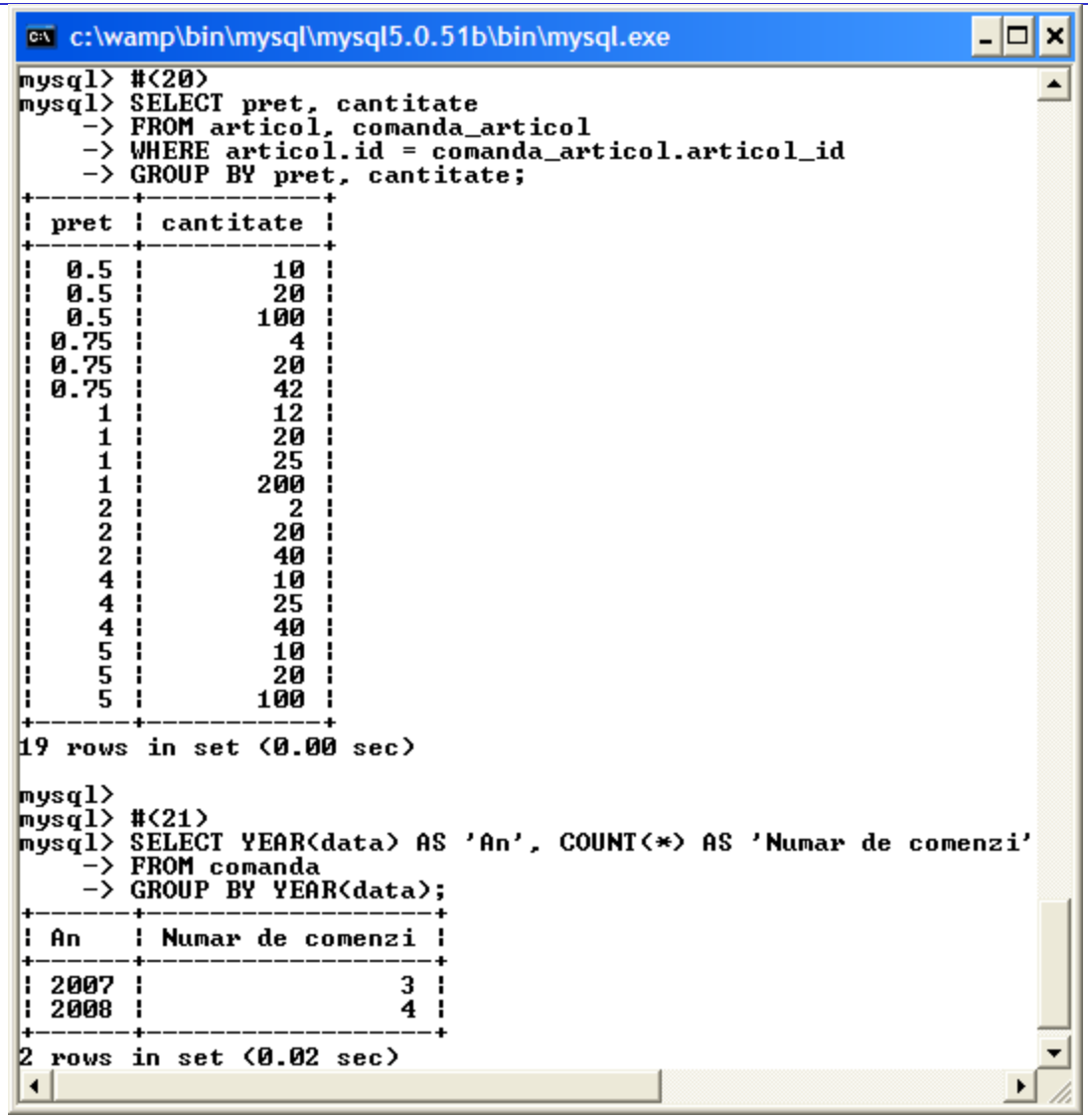
SQL

(20) (GROUP BY dupa mai multe coloane) Listati comenzile, cu pretul articolelor si cantitatea.

(21) (GROUP BY expresii) Listati numarul de comenzi pe fiecare an

(20) (GROUP BY se poate utiliza cu mai multe coloane; in exemplul acesta se listeaza articolele din fiecare comanda, precizand pretul si cantitatea

(21) Se utilizeaza GROUP BY folosind o expresie drept criteriu de grupare; in acest caz, expresia e foarte simpla, si anume anul comenzii (comanda are una dintre coloane data).

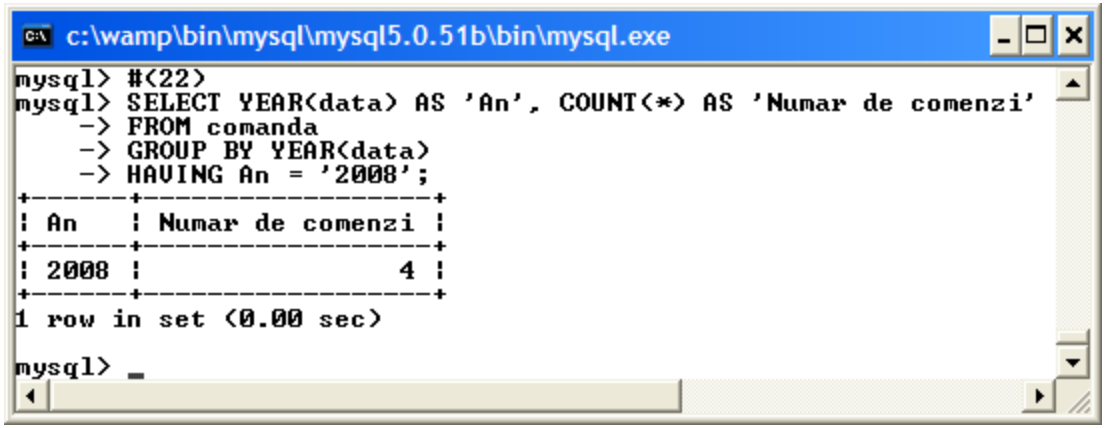


GROUP BY … HAVING

In cazul utilizarii clauzei GROUP BY, pentru filtrarea aplicata rezultatului aplicarii acestei clauze, se utilizeaza clauza HAVING

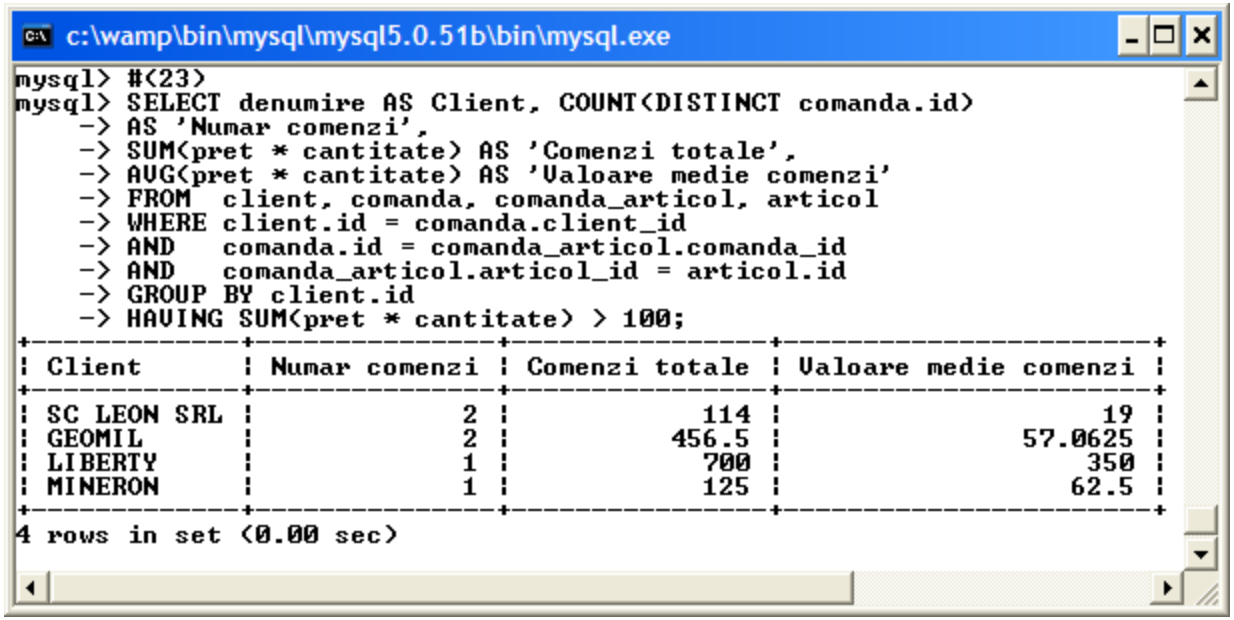
Fiecare specificatie de coloana specificata in clauza HAVING trebuie sa apara fie intr-o functie agregata fie in lista coloanelor invocate in claua GROUP BY

(22) Listati numarul de comenzi din anul 2008



SQL

(23) Listati numarul de comenzi pentru fiecare client, cu suma totala si media valorii comenzilor, avand suma totala peste 100 lei.



SQL

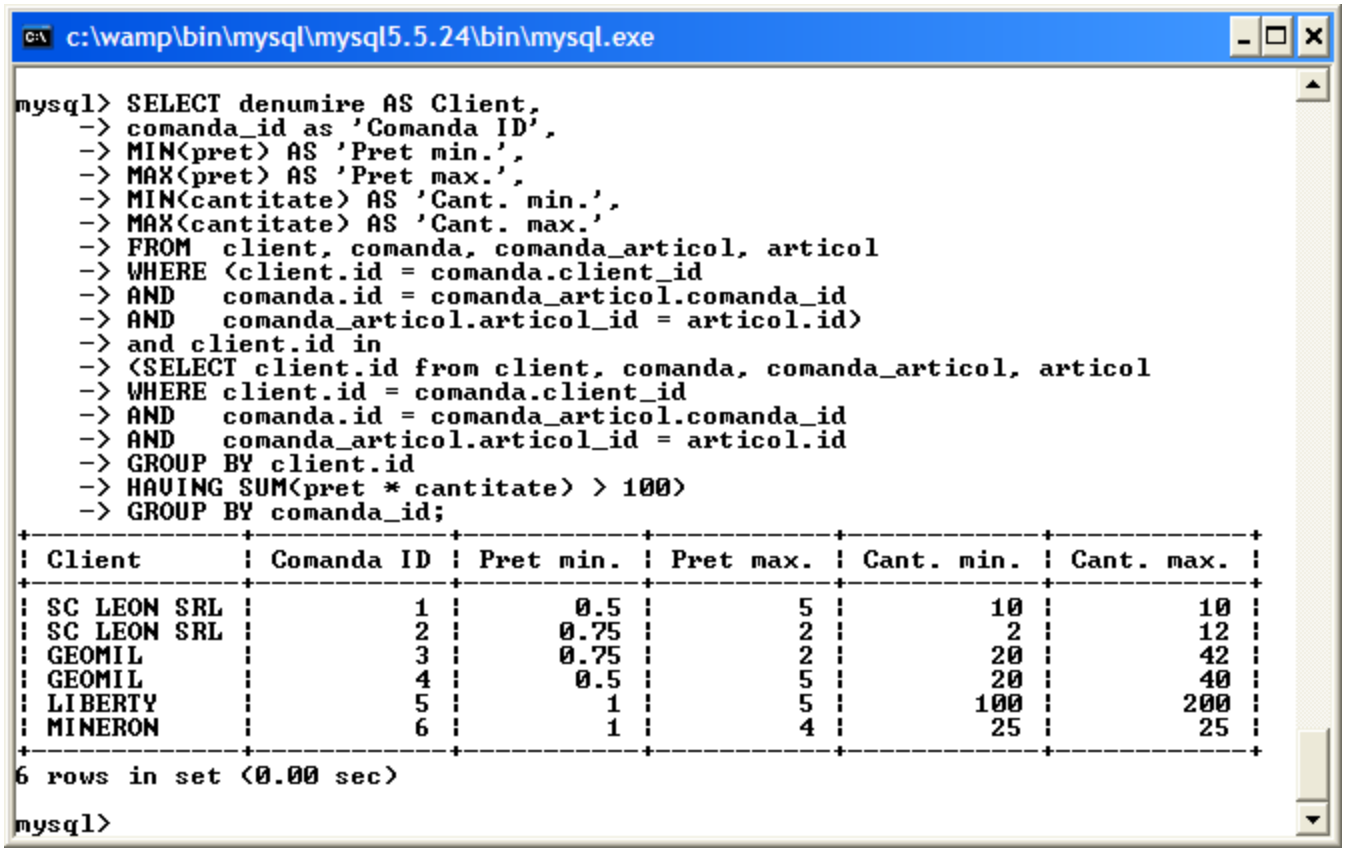
(24) (Drill down) Listati comenzile, cu articolele si cantitatea, pentru acele comenzi pentru care suma totala a comenzii este peste 100 lei.

c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe

```
mysql> SELECT denumire AS Client,
    -> comanda.id as Comanda,
    -> articol.descriere As Articol,
    -> comanda_articol.cantitate As Cantitate
    -> FROM  client, comanda, comanda_articol, articol
    -> WHERE (client.id = comanda.client_id
    -> AND   comanda.id = comanda_articol.comanda_id
    -> AND   comanda_articol.articol_id = articol.id)
    -> and client.id in
    -> (SELECT client.id from client, comanda, comanda_articol, articol
    -> WHERE client.id = comanda.client_id
    -> AND   comanda.id = comanda_articol.comanda_id
    -> AND   comanda_articol.articol_id = articol.id
    -> GROUP BY client.id
    -> HAVING SUM(pret * cantitate) > 100);
+-------------+---------+----------+-----------+
| Client      | Comanda | Articol  | Cantitate |
+-------------+---------+----------+-----------+
| SC LEON SRL |       1 | stilou   |        10 |
| SC LEON SRL |       1 | pix bila |        10 |
| SC LEON SRL |       1 | creion   |        10 |
| SC LEON SRL |       2 | marker   |         2 |
| SC LEON SRL |       2 | guma     |         4 |
| SC LEON SRL |       2 | rigla    |        12 |
| GEOMIL      |       3 | penar    |        20 |
| GEOMIL      |       3 | guma     |        42 |
| GEOMIL      |       4 | stilou   |        20 |
| GEOMIL      |       4 | pix bila |        40 |
| GEOMIL      |       4 | creion   |        20 |
| GEOMIL      |       4 | marker   |        40 |
| GEOMIL      |       4 | guma     |        20 |
| GEOMIL      |       4 | rigla    |        20 |
| LIBERTY     |       5 | stilou   |       100 |
| LIBERTY     |       5 | echer    |       200 |
| MINERON     |       6 | pix bila |        25 |
| MINERON     |       6 | rigla    |        25 |
+-------------+---------+----------+-----------+
18 rows in set (0.00 sec)

mysql>
```

SQL

(25) (Drill down) Listati pretul minim, pretul maxim, cantitatea minima, cantitatea maxima, pentru fiecare comanda de la acei clienti pentru care suma totala a comenzilor este intre 100 si 500 de lei.



```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe

mysql> SELECT denumire AS Client,
    -> comanda_id as 'Comanda ID',
    -> MIN(pret) AS 'Pret min.',
    -> MAX(pret) AS 'Pret max.',
    -> MIN(cantitate) AS 'Cant. min.',
    -> MAX(cantitate) AS 'Cant. max.'
    -> FROM  client, comanda, comanda_articol, articol
    -> WHERE (client.id = comanda.client_id
    -> AND   comanda.id = comanda_articol.comanda_id
    -> AND   comanda_articol.articol_id = articol.id)
    -> and client.id in
    -> (SELECT client.id from client, comanda, comanda_articol, articol
    -> WHERE client.id = comanda.client_id
    -> AND   comanda.id = comanda_articol.comanda_id
    -> AND   comanda_articol.articol_id = articol.id
    -> GROUP BY client.id
    -> HAVING SUM(pret * cantitate) > 100)
    -> GROUP BY comanda_id;
+------------+------------+----------+----------+------------+------------+
| Client     | Comanda ID | Pret min. | Pret max. | Cant. min. | Cant. max. |
+------------+------------+----------+----------+------------+------------+
| SC LEON SRL |          1 |      0.5 |        5 |         10 |         10 |
| SC LEON SRL |          2 |     0.75 |        2 |          2 |         12 |
| GEOMIL     |          3 |     0.75 |        2 |         20 |         42 |
| GEOMIL     |          4 |      0.5 |        5 |         20 |         40 |
| LIBERTY    |          5 |        1 |        5 |        100 |        200 |
| MINERON    |          6 |        1 |        4 |         25 |         25 |
+------------+------------+----------+----------+------------+------------+
6 rows in set (0.00 sec)

mysql>
```