SQL

Tranzactionare

Update

Proceduri si functii

TRANSACTION

```
START TRANSACTION [WITH CONSISTENT SNAPSHOT] | BEGIN [WORK]
COMMIT [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
ROLLBACK [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
SET autocommit = {0 | 1}
```

START TRANSACTION sau **BEGIN** incepe o noua tranzactie.

COMMIT comite tranzactia curenta facand permanente schimbarile.

ROLLBACK da inapoi tranzactia curenta, prin anularea schimbarilor sale.

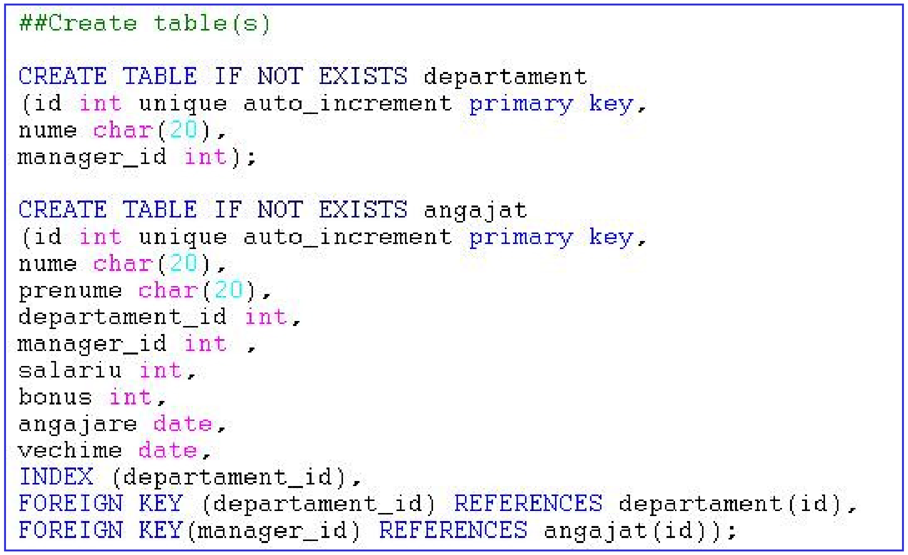
Instructiunea **SET** *autocommit* activeaza sau dezactiveaza valoarea default a flag-ului *autocommit*.

Clauza AND CHAIN face ca o noua tranzactie sa inceapa imediat ce tranzactia curenta se incheie. Clauza RELEASE face ca serverul sa deconecteze sesiunea client curenta dupa terminarea tranzactiei curente.

MySQL ruleaza in modul autocommit activat. Asta inseamna ca, imediat ce se executa o comanda care modifica o tabela, MySQL salveaza modificarea pe disc pentru a face aceasta modificare permanenta.

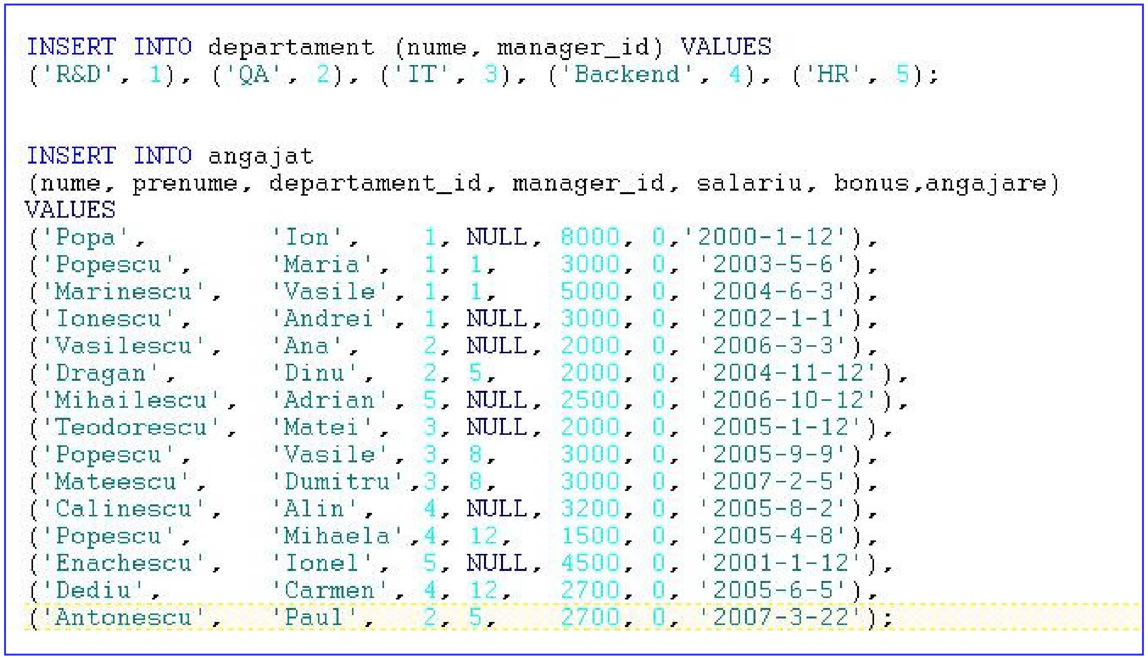
SQL

Creare structura baza de date:



```
##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,
bonus int,
angajare date,
vechime date,
INDEX (departament_id),
FOREIGN KEY (departament_id) REFERENCES departament(id),
FOREIGN KEY(manager_id) REFERENCES angajat(id));
```

Populare tabele:

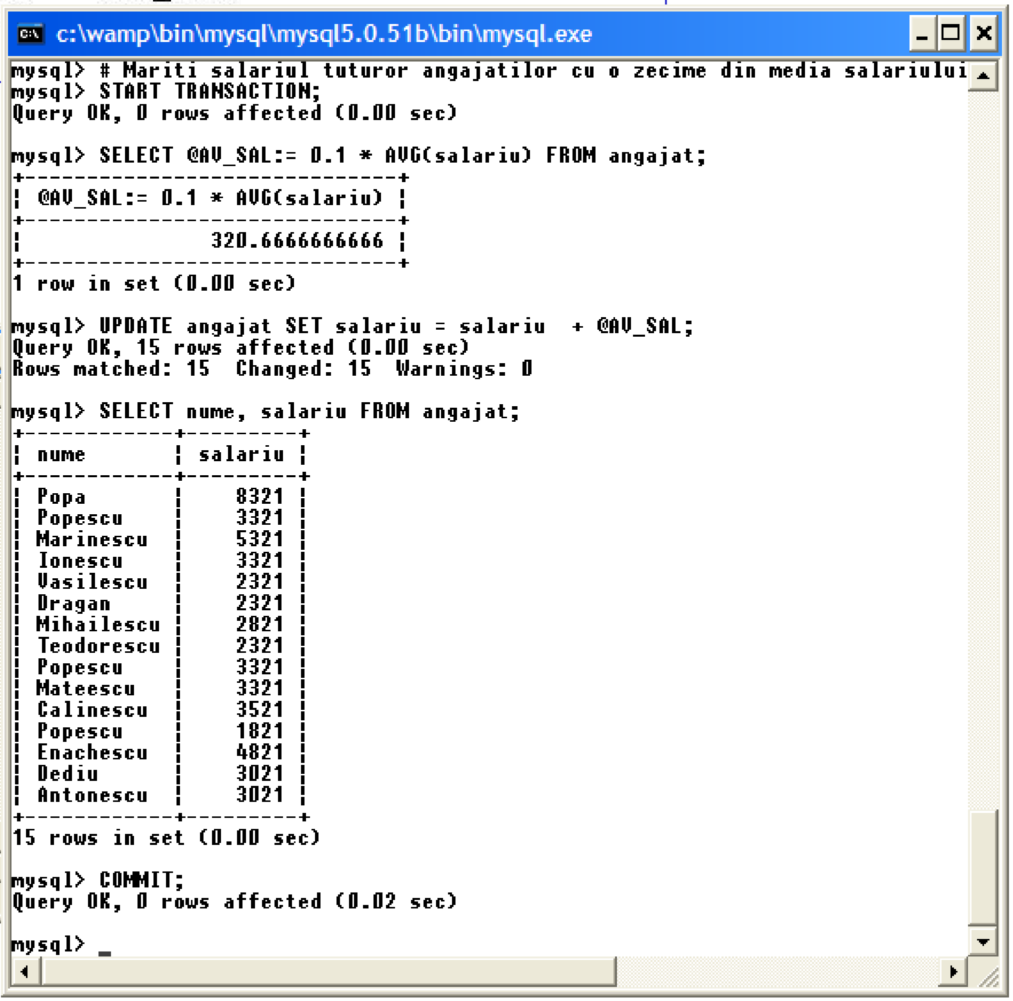


```
INSERT INTO departament (nume, manager_id) VALUES
('R&D', 1), ('QA', 2), ('IT', 3), ('Backend', 4), ('HR', 5);


INSERT INTO angajat
(nume, prenume, departament_id, manager_id, salariu, bonus,angajare)
VALUES
('Popa',        'Ion',     1, NULL, 8000, 0,'2000-1-12'),
('Popescu',     'Maria',   1, 1,    3000, 0, '2003-5-6'),
('Marinescu',   'Vasile',  1, 1,    5000, 0, '2004-6-3'),
('Ionescu',     'Andrei',  1, NULL, 3000, 0, '2002-1-1'),
('Vasilescu',   'Ana',     2, NULL, 2000, 0, '2006-3-3'),
('Dragan',      'Dinu',    2, 5,    2000, 0, '2004-11-12'),
('Mihailescu',  'Adrian',  5, NULL, 2500, 0, '2006-10-12'),
('Teodorescu',  'Matei',   3, NULL, 2000, 0, '2005-1-12'),
('Popescu',     'Vasile',  3, 8,    3000, 0, '2005-9-9'),
('Mateescu',    'Dumitru',3, 8,     3000, 0, '2007-2-5'),
('Calinescu',   'Alin',    4, NULL, 3200, 0, '2005-8-2'),
('Popescu',     'Mihaela',4, 12,    1500, 0, '2005-4-8'),
('Enachescu',   'Ionel',   5, NULL, 4500, 0, '2001-1-12'),
('Dediu',       'Carmen',  4, 12,   2700, 0, '2005-6-5'),
('Antonescu',   'Paul',    2, 5,    2700, 0, '2007-3-22');
```

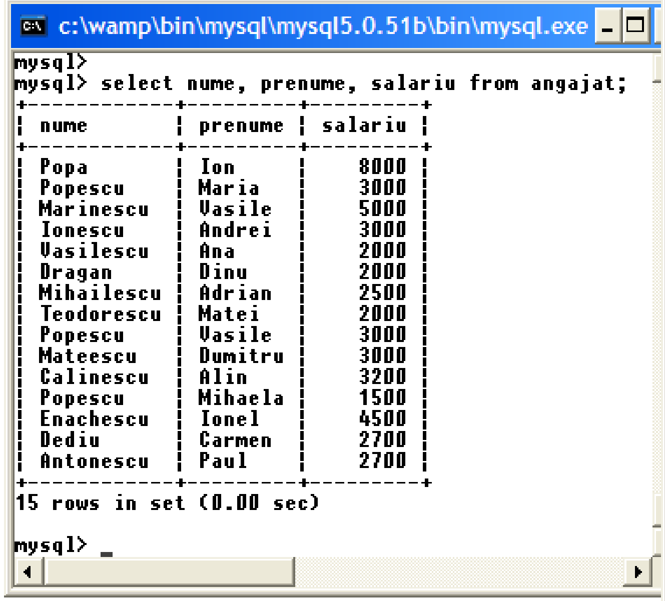
SQL

Tranzactie simpla:

```
# Mariti salariul tuturor angajatilor cu o zecime din media salariului
START TRANSACTION;
SELECT @AV_SAL:= 0.1 * AVG(salariu) FROM angajat;
UPDATE angajat SET salariu = salariu  + @AV_SAL;
SELECT nume, salariu FROM angajat;
COMMIT;
```



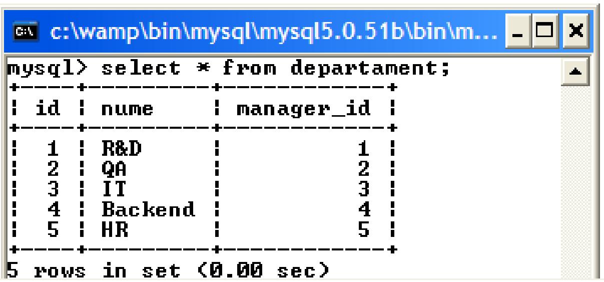
```
mysql> # Mariti salariul tuturor angajatilor cu o zecime din media salariului
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @AV_SAL:= 0.1 * AVG(salariu) FROM angajat;
+-----------------------------+
| @AV_SAL:= 0.1 * AVG(salariu) |
+-----------------------------+
|              320.6666666666 |
+-----------------------------+
1 row in set (0.00 sec)

mysql> UPDATE angajat SET salariu = salariu  + @AV_SAL;
Query OK, 15 rows affected (0.00 sec)
Rows matched: 15  Changed: 15  Warnings: 0

mysql> SELECT nume, salariu FROM angajat;
+------------+---------+
| nume       | salariu |
+------------+---------+
| Popa       |    8321 |
| Popescu    |    3321 |
| Marinescu  |    5321 |
| Ionescu    |    3321 |
| Vasilescu  |    2321 |
| Dragan     |    2321 |
| Mihailescu |    2821 |
| Teodorescu |    2321 |
| Popescu    |    3321 |
| Mateescu   |    3321 |
| Calinescu  |    3521 |
| Popescu    |    1821 |
| Enachescu  |    4821 |
| Dediu      |    3021 |
| Antonescu  |    3021 |
+------------+---------+
15 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.02 sec)

mysql> _
```



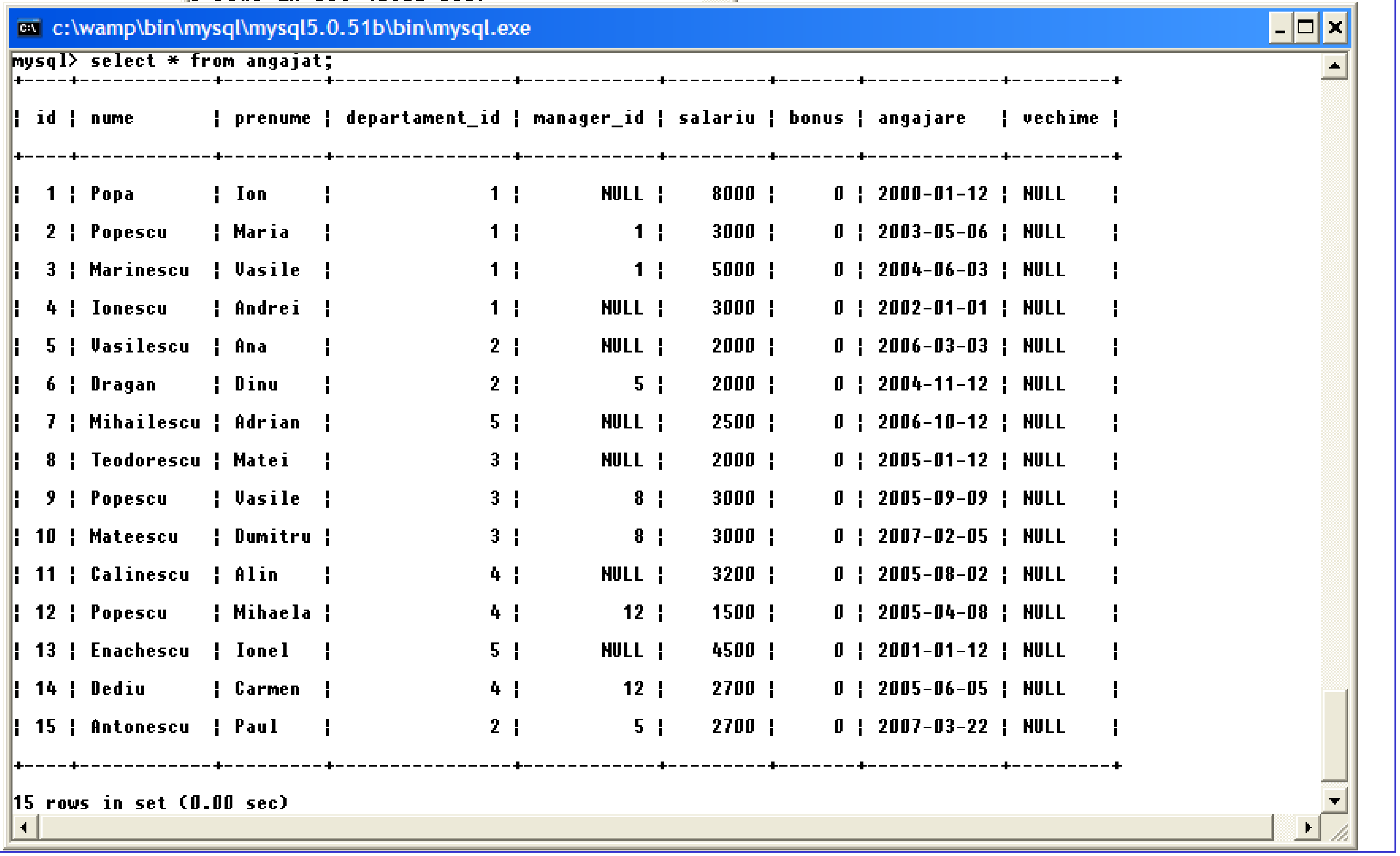
```
mysql>
mysql> select nume, prenume, salariu from angajat;
+------------+---------+---------+
| nume       | prenume | salariu |
+------------+---------+---------+
| Popa       | Ion     |    8000 |
| Popescu    | Maria   |    3000 |
| Marinescu  | Vasile  |    5000 |
| Ionescu    | Andrei  |    3000 |
| Vasilescu  | Ana     |    2000 |
| Dragan     | Dinu    |    2000 |
| Mihailescu | Adrian  |    2500 |
| Teodorescu | Matei   |    2000 |
| Popescu    | Vasile  |    3000 |
| Mateescu   | Dumitru |    3000 |
| Calinescu  | Alin    |    3200 |
| Popescu    | Mihaela |    1500 |
| Enachescu  | Ionel   |    4500 |
| Dediu      | Carmen  |    2700 |
| Antonescu  | Paul    |    2700 |
+------------+---------+---------+
15 rows in set (0.00 sec)

mysql> _
```

SQL

Tabele rezultate:



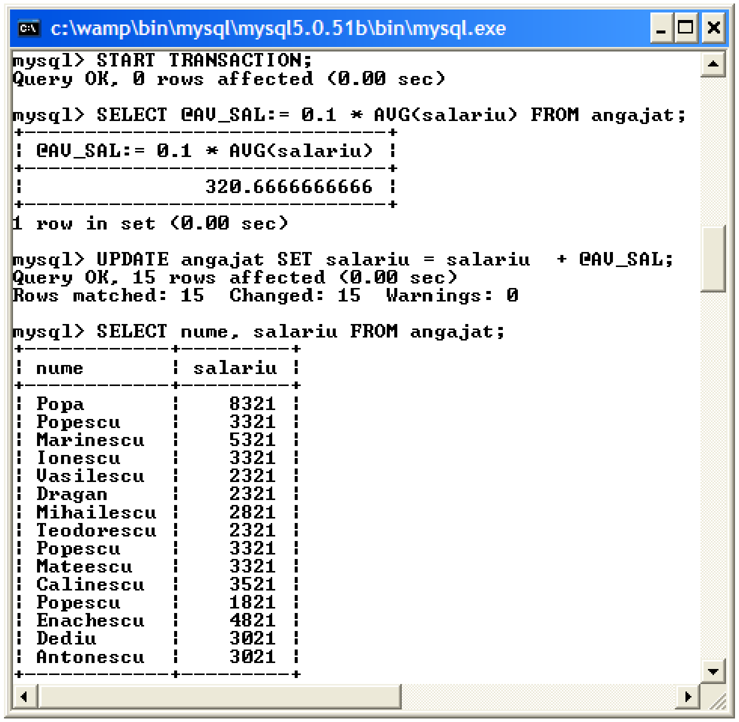
```
mysql> select * from departament;
+----+---------+------------+
| id | nume    | manager_id |
+----+---------+------------+
|  1 | R&D     |          1 |
|  2 | QA      |          2 |
|  3 | IT      |          3 |
|  4 | Backend |          4 |
|  5 | HR      |          5 |
+----+---------+------------+
5 rows in set (0.00 sec)
```



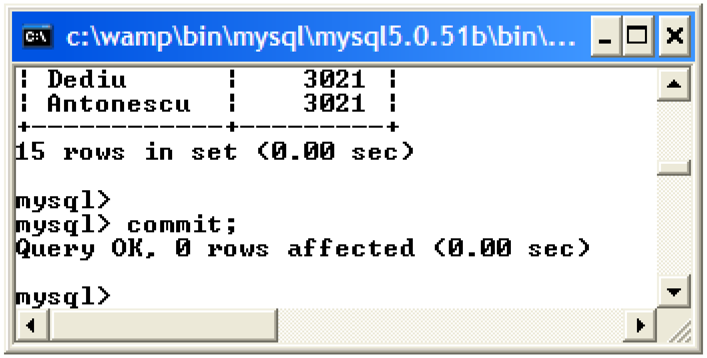
```
mysql> select * from angajat;
+----+------------+---------+----------------+------------+---------+-------+------------+---------+
| id | nume       | prenume | departament_id | manager_id | salariu | bonus | angajare   | vechime |
+----+------------+---------+----------------+------------+---------+-------+------------+---------+
|  1 | Popa       | Ion     |              1 |       NULL |    8000 |     0 | 2000-01-12 | NULL    |
|  2 | Popescu    | Maria   |              1 |          1 |    3000 |     0 | 2003-05-06 | NULL    |
|  3 | Marinescu  | Vasile  |              1 |          1 |    5000 |     0 | 2004-06-03 | NULL    |
|  4 | Ionescu    | Andrei  |              1 |       NULL |    3000 |     0 | 2002-01-01 | NULL    |
|  5 | Vasilescu  | Ana     |              2 |       NULL |    2000 |     0 | 2006-03-03 | NULL    |
|  6 | Dragan     | Dinu    |              2 |          5 |    2000 |     0 | 2004-11-12 | NULL    |
|  7 | Mihailescu | Adrian  |              5 |       NULL |    2500 |     0 | 2006-10-12 | NULL    |
|  8 | Teodorescu | Matei   |              3 |       NULL |    2000 |     0 | 2005-01-12 | NULL    |
|  9 | Popescu    | Vasile  |              3 |          8 |    3000 |     0 | 2005-09-09 | NULL    |
| 10 | Mateescu   | Dumitru |              3 |          8 |    3000 |     0 | 2007-02-05 | NULL    |
| 11 | Calinescu  | Alin    |              4 |       NULL |    3200 |     0 | 2005-08-02 | NULL    |
| 12 | Popescu    | Mihaela |              4 |         12 |    1500 |     0 | 2005-04-08 | NULL    |
| 13 | Enachescu  | Ionel   |              5 |       NULL |    4500 |     0 | 2001-01-12 | NULL    |
| 14 | Dediu      | Carmen  |              4 |         12 |    2700 |     0 | 2005-06-05 | NULL    |
| 15 | Antonescu  | Paul    |              2 |          5 |    2700 |     0 | 2007-03-22 | NULL    |
+----+------------+---------+----------------+------------+---------+-------+------------+---------+

15 rows in set (0.00 sec)
```

SQL

No commit before COMMIT!!

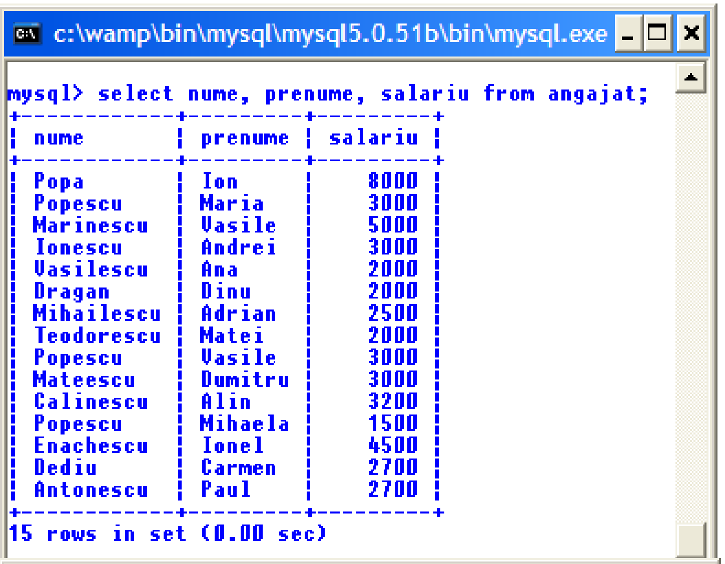


```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @AV_SAL:= 0.1 * AVG(salariu) FROM angajat;
+------------------------------+
| @AV_SAL:= 0.1 * AVG(salariu) |
+------------------------------+
|               320.6666666666 |
+------------------------------+
1 row in set (0.00 sec)

mysql> UPDATE angajat SET salariu = salariu  + @AV_SAL;
Query OK, 15 rows affected (0.00 sec)
Rows matched: 15  Changed: 15  Warnings: 0

mysql> SELECT nume, salariu FROM angajat;
+------------+---------+
| nume       | salariu |
+------------+---------+
| Popa       |    8321 |
| Popescu    |    3321 |
| Marinescu  |    5321 |
| Ionescu    |    3321 |
| Vasilescu  |    2321 |
| Dragan     |    2321 |
| Mihailescu |    2821 |
| Teodorescu |    2321 |
| Popescu    |    3321 |
| Mateescu   |    3321 |
| Calinescu  |    3521 |
| Popescu    |    1821 |
| Enachescu  |    4821 |
| Dediu      |    3021 |
| Antonescu  |    3021 |
+------------+---------+
```



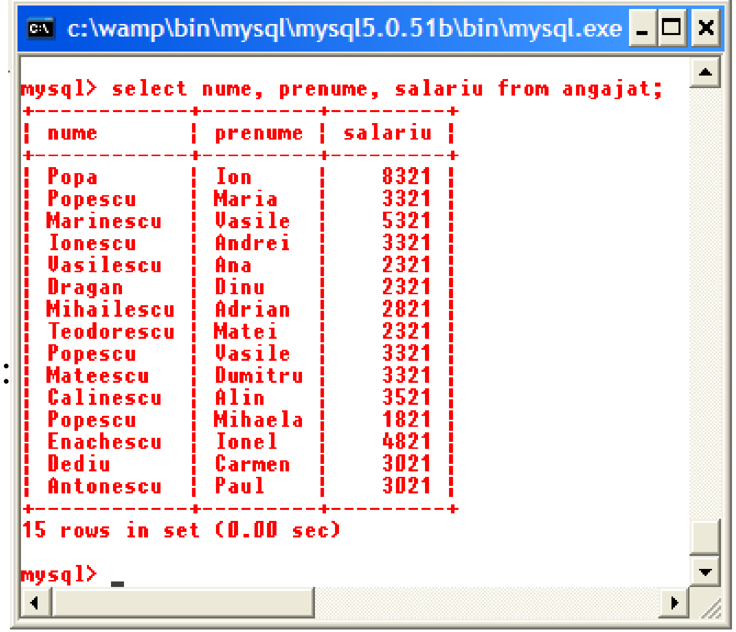
```
| Dediu      |    3021 |
| Antonescu  |    3021 |
+------------+---------+
15 rows in set (0.00 sec)

mysql>
mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Inainte de COMMIT:



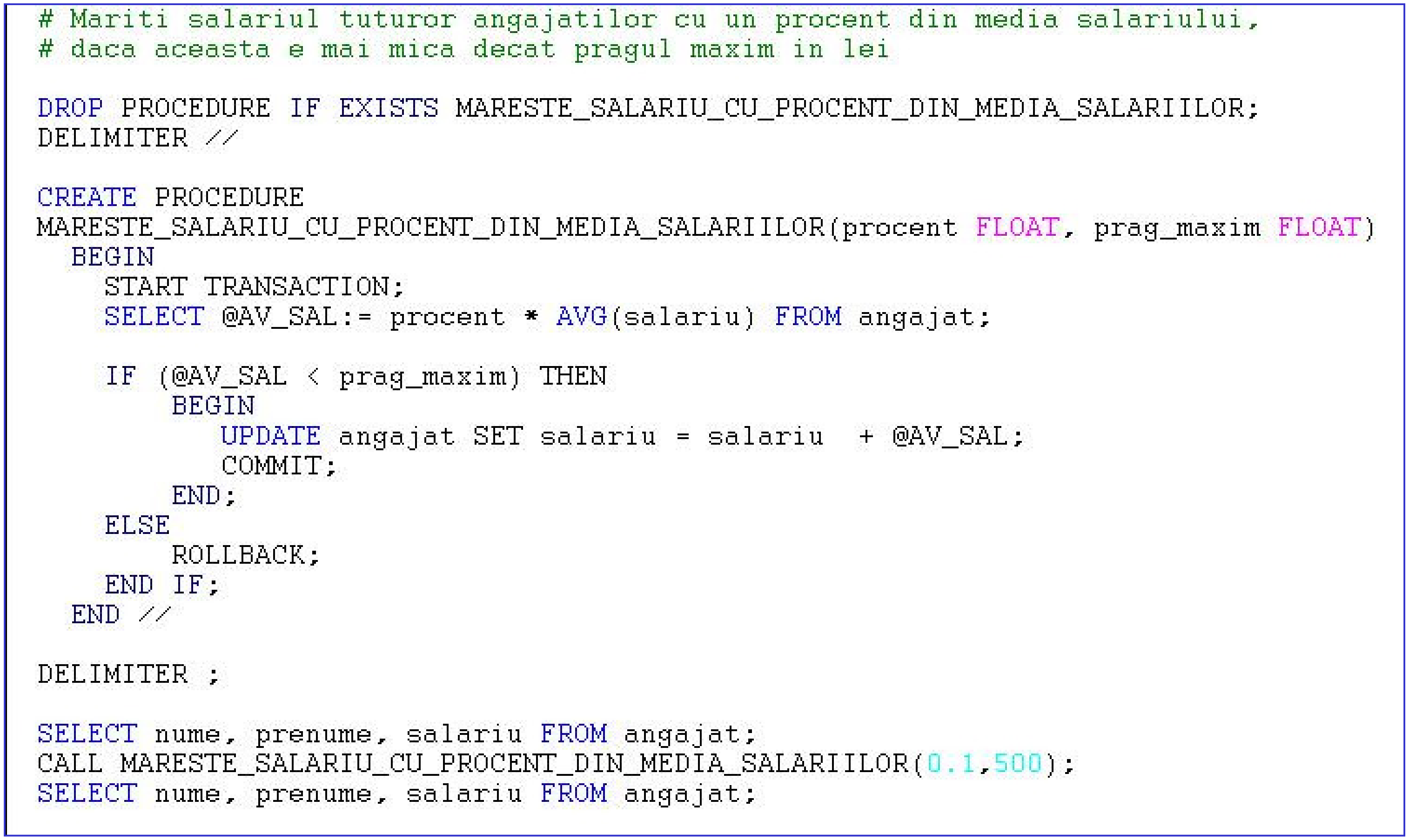
```
mysql> select nume, prenume, salariu from angajat;
+------------+---------+---------+
| nume       | prenume | salariu |
+------------+---------+---------+
| Popa       | Ion     |    8000 |
| Popescu    | Maria   |    3000 |
| Marinescu  | Vasile  |    5000 |
| Ionescu    | Andrei  |    3000 |
| Vasilescu  | Ana     |    2000 |
| Dragan     | Dinu    |    2000 |
| Mihailescu | Adrian  |    2500 |
| Teodorescu | Matei   |    2000 |
| Popescu    | Vasile  |    3000 |
| Mateescu   | Dumitru |    3000 |
| Calinescu  | Alin    |    3200 |
| Popescu    | Mihaela |    1500 |
| Enachescu  | Ionel   |    4500 |
| Dediu      | Carmen  |    2700 |
| Antonescu  | Paul    |    2700 |
+------------+---------+---------+
15 rows in set (0.00 sec)
```

Dupa COMMIT:



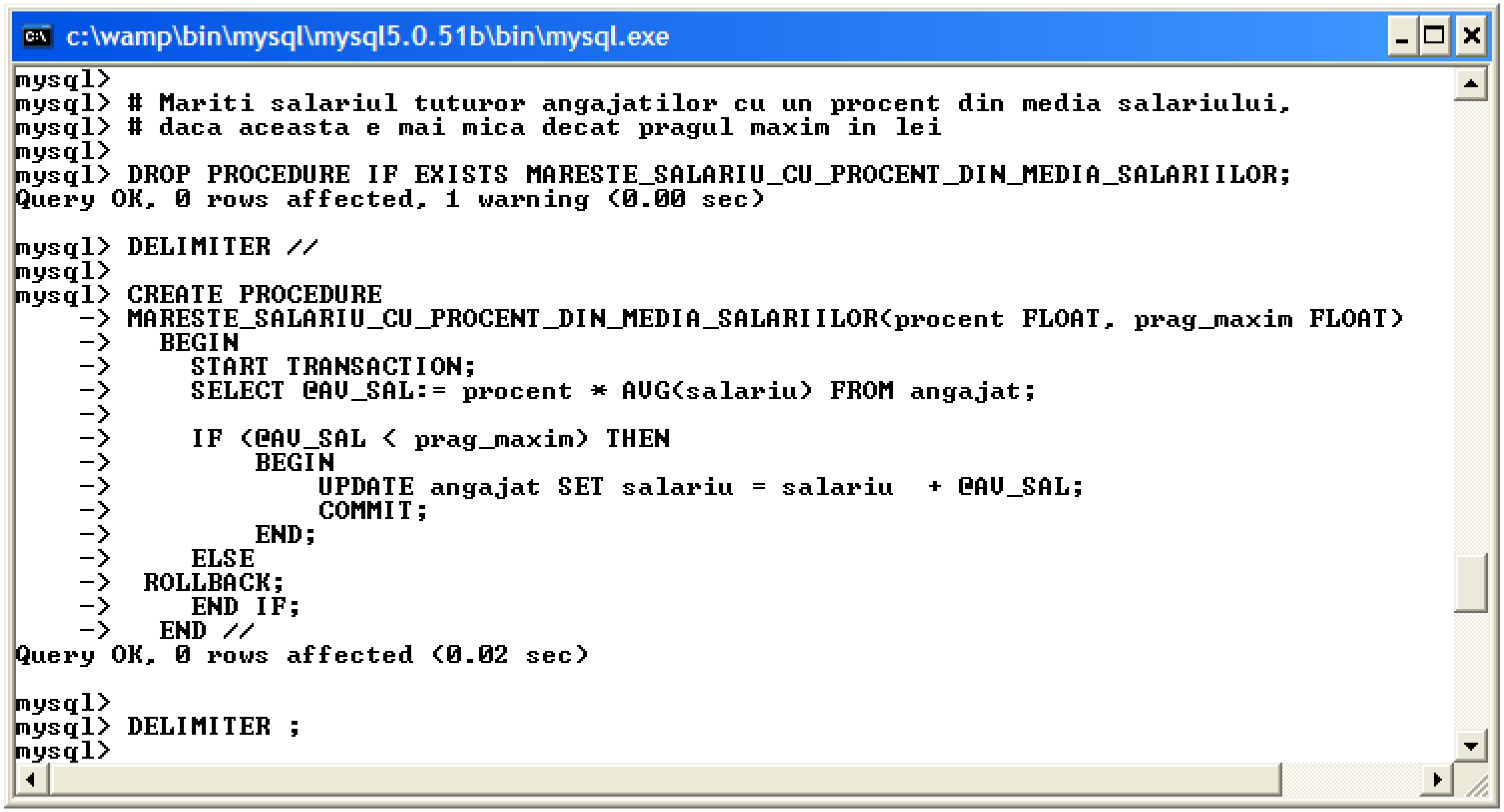
```
mysql> select nume, prenume, salariu from angajat;
+------------+---------+---------+
| nume       | prenume | salariu |
+------------+---------+---------+
| Popa       | Ion     |    8321 |
| Popescu    | Maria   |    3321 |
| Marinescu  | Vasile  |    5321 |
| Ionescu    | Andrei  |    3321 |
| Vasilescu  | Ana     |    2321 |
| Dragan     | Dinu    |    2321 |
| Mihailescu | Adrian  |    2821 |
| Teodorescu | Matei   |    2321 |
| Popescu    | Vasile  |    3321 |
| Mateescu   | Dumitru |    3321 |
| Calinescu  | Alin    |    3521 |
| Popescu    | Mihaela |    1821 |
| Enachescu  | Ionel   |    4821 |
| Dediu      | Carmen  |    3021 |
| Antonescu  | Paul    |    3021 |
+------------+---------+---------+
15 rows in set (0.00 sec)

mysql> _
```

SQL



```
# Mariti salariul tuturor angajatilor cu un procent din media salariului,
# daca aceasta e mai mica decat pragul maxim in lei

DROP PROCEDURE IF EXISTS MARESTE_SALARIU_CU_PROCENT_DIN_MEDIA_SALARIILOR;
DELIMITER //

CREATE PROCEDURE
MARESTE_SALARIU_CU_PROCENT_DIN_MEDIA_SALARIILOR(procent FLOAT, prag_maxim FLOAT)
  BEGIN
    START TRANSACTION;
    SELECT @AV_SAL:= procent * AVG(salariu) FROM angajat;

    IF (@AV_SAL < prag_maxim) THEN
        BEGIN
           UPDATE angajat SET salariu = salariu  + @AV_SAL;
           COMMIT;
        END;
    ELSE
        ROLLBACK;
    END IF;
  END //

DELIMITER ;

SELECT nume, prenume, salariu FROM angajat;
CALL MARESTE_SALARIU_CU_PROCENT_DIN_MEDIA_SALARIILOR(0.1,500);
SELECT nume, prenume, salariu FROM angajat;
```

SQL

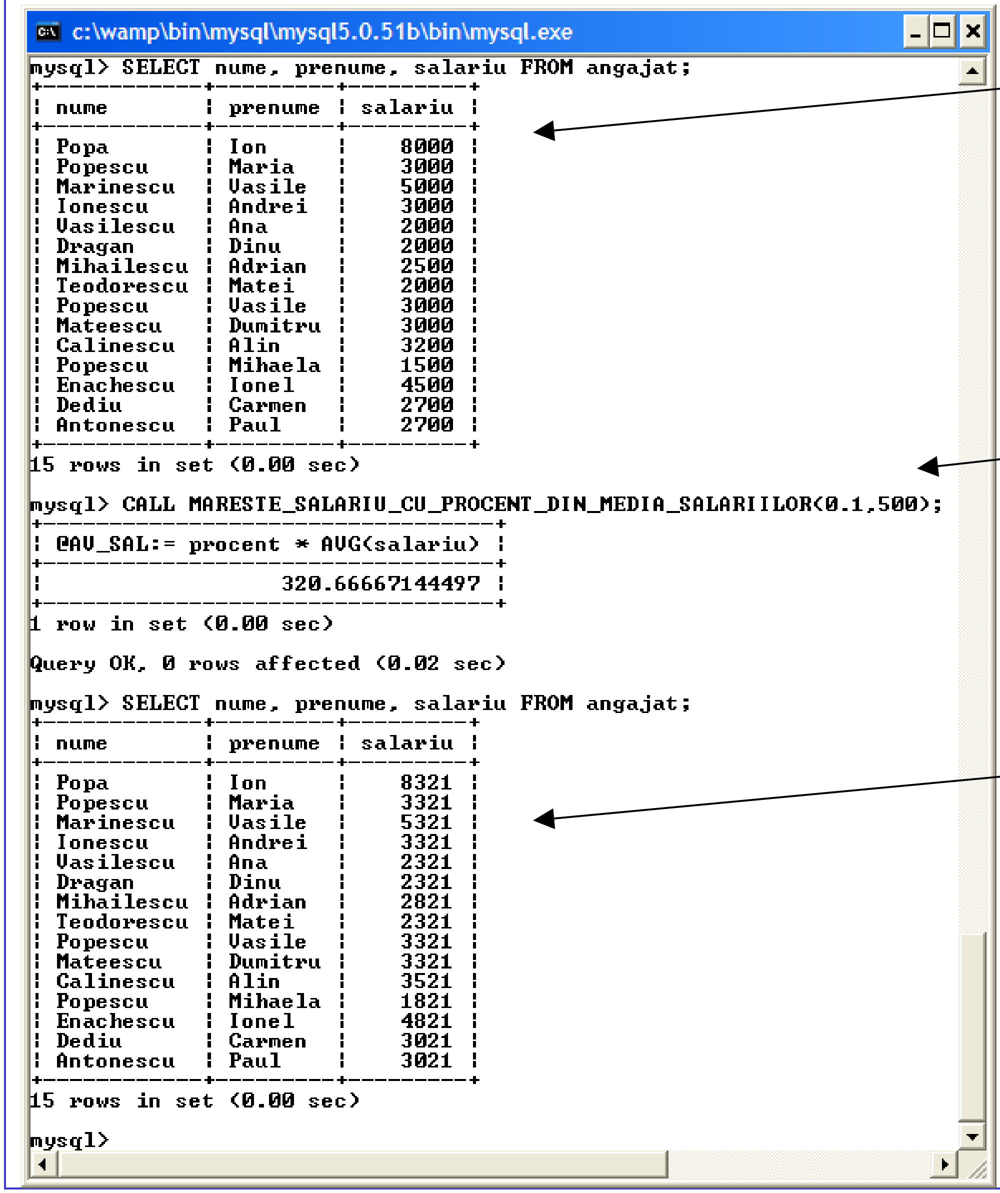


c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

```
mysql>
mysql> # Mariti salariul tuturor angajatilor cu un procent din media salariului,
mysql> # daca aceasta e mai mica decat pragul maxim in lei
mysql>
mysql> DROP PROCEDURE IF EXISTS MARESTE_SALARIU_CU_PROCENT_DIN_MEDIA_SALARIILOR;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE
    -> MARESTE_SALARIU_CU_PROCENT_DIN_MEDIA_SALARIILOR(procent FLOAT, prag_maxim FLOAT)
    ->   BEGIN
    ->     START TRANSACTION;
    ->     SELECT @AV_SAL:= procent * AVG(salariu) FROM angajat;
    ->
    ->     IF (@AV_SAL < prag_maxim) THEN
    ->         BEGIN
    ->             UPDATE angajat SET salariu = salariu  + @AV_SAL;
    ->             COMMIT;
    ->         END;
    ->     ELSE
    ->  ROLLBACK;
    ->     END IF;
    ->   END //
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

aceasta

SQL



```
mysql> SELECT nume, prenume, salariu FROM angajat;
+------------+---------+---------+
| nume       | prenume | salariu |
+------------+---------+---------+
| Popa       | Ion     |    8000 |
| Popescu    | Maria   |    3000 |
| Marinescu  | Vasile  |    5000 |
| Ionescu    | Andrei  |    3000 |
| Vasilescu  | Ana     |    2000 |
| Dragan     | Dinu    |    2000 |
| Mihailescu | Adrian  |    2500 |
| Teodorescu | Matei   |    2000 |
| Popescu    | Vasile  |    3000 |
| Mateescu   | Dumitru |    3000 |
| Calinescu  | Alin    |    3200 |
| Popescu    | Mihaela |    1500 |
| Enachescu  | Ionel   |    4500 |
| Dediu      | Carmen  |    2700 |
| Antonescu  | Paul    |    2700 |
+------------+---------+---------+
15 rows in set (0.00 sec)

mysql> CALL MARESTE_SALARIU_CU_PROCENT_DIN_MEDIA_SALARIILOR(0.1,500);
+-----------------------------------+
| @AV_SAL:= procent * AVG(salariu) |
+-----------------------------------+
|                   320.66667144497 |
+-----------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

mysql> SELECT nume, prenume, salariu FROM angajat;
+------------+---------+---------+
| nume       | prenume | salariu |
+------------+---------+---------+
| Popa       | Ion     |    8321 |
| Popescu    | Maria   |    3321 |
| Marinescu  | Vasile  |    5321 |
| Ionescu    | Andrei  |    3321 |
| Vasilescu  | Ana     |    2321 |
| Dragan     | Dinu    |    2321 |
| Mihailescu | Adrian  |    2821 |
| Teodorescu | Matei   |    2321 |
| Popescu    | Vasile  |    3321 |
| Mateescu   | Dumitru |    3321 |
| Calinescu  | Alin    |    3521 |
| Popescu    | Mihaela |    1821 |
| Enachescu  | Ionel   |    4821 |
| Dediu      | Carmen  |    3021 |
| Antonescu  | Paul    |    3021 |
+------------+---------+---------+
15 rows in set (0.00 sec)

mysql>
```

Inainte de tranzactie:

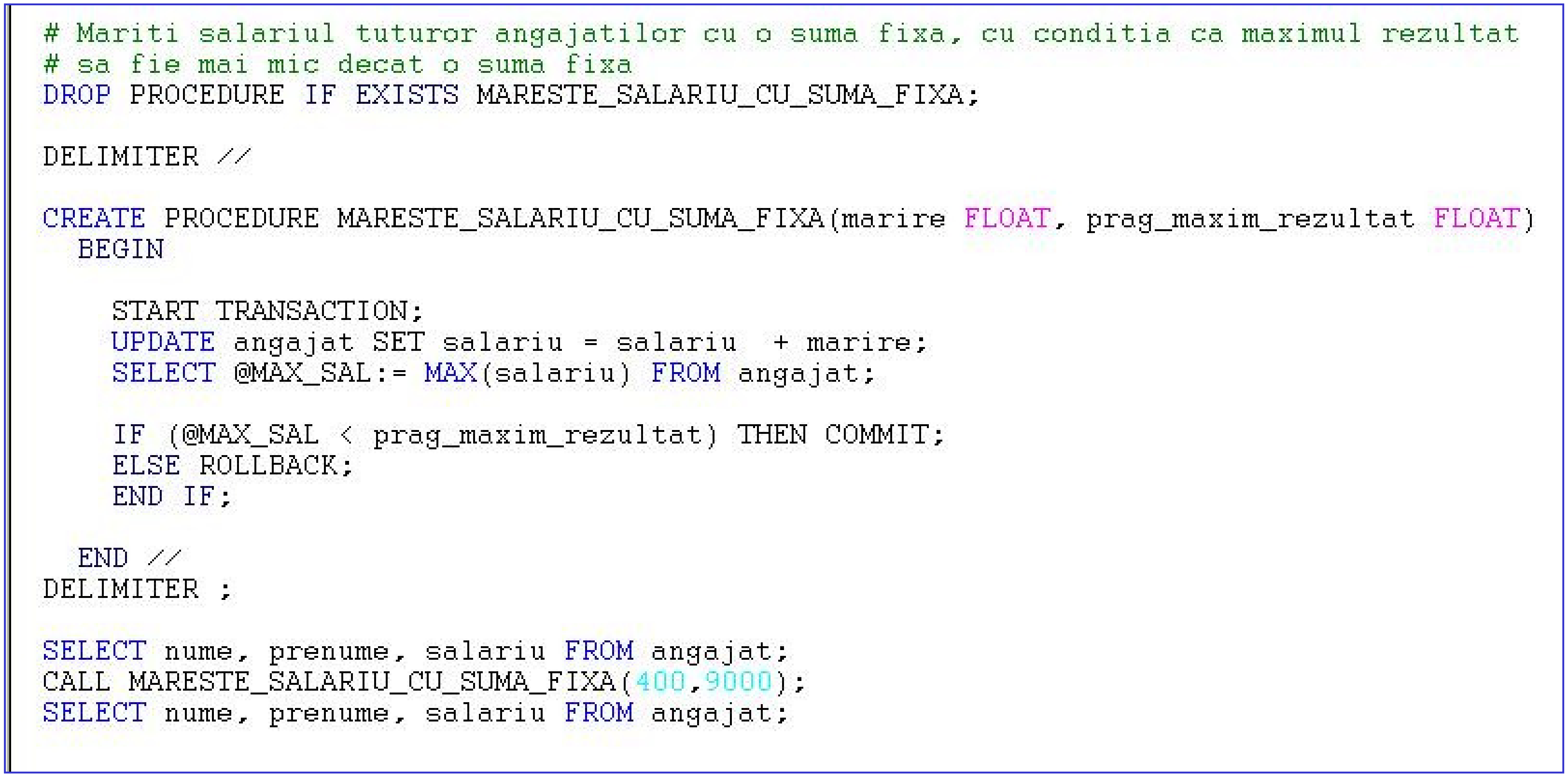
Se apeleaza procedura care include o tranzactie cu parametrii:

- procent din medie: 10%
- pragul maxim: 500 lei

Dupa tranzactie:

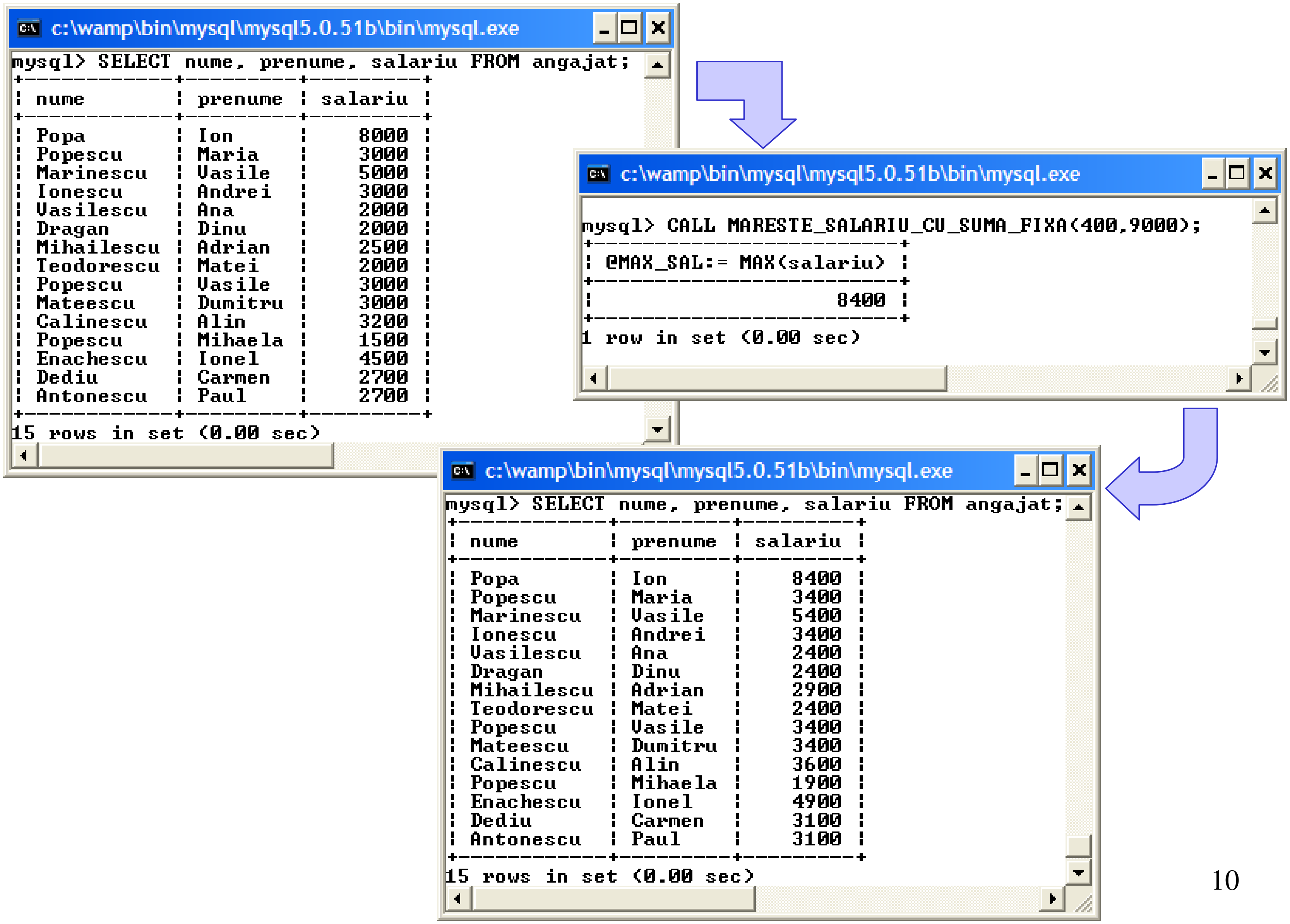
SQL

Exemplul 2: Mariti salariul tuturor angajatilor cu o suma fixa, cu conditia ca maximul rezultat sa fie mai mic decat o suma fixa



```
# Mariti salariul tuturor angajatilor cu o suma fixa, cu conditia ca maximul rezultat
# sa fie mai mic decat o suma fixa
DROP PROCEDURE IF EXISTS MARESTE_SALARIU_CU_SUMA_FIXA;

DELIMITER //

CREATE PROCEDURE MARESTE_SALARIU_CU_SUMA_FIXA(marire FLOAT, prag_maxim_rezultat FLOAT)
  BEGIN

    START TRANSACTION;
    UPDATE angajat SET salariu = salariu  + marire;
    SELECT @MAX_SAL:= MAX(salariu) FROM angajat;

    IF (@MAX_SAL < prag_maxim_rezultat) THEN COMMIT;
    ELSE ROLLBACK;
    END IF;

  END //
DELIMITER ;

SELECT nume, prenume, salariu FROM angajat;
CALL MARESTE_SALARIU_CU_SUMA_FIXA(400,9000);
SELECT nume, prenume, salariu FROM angajat;
```

SQL



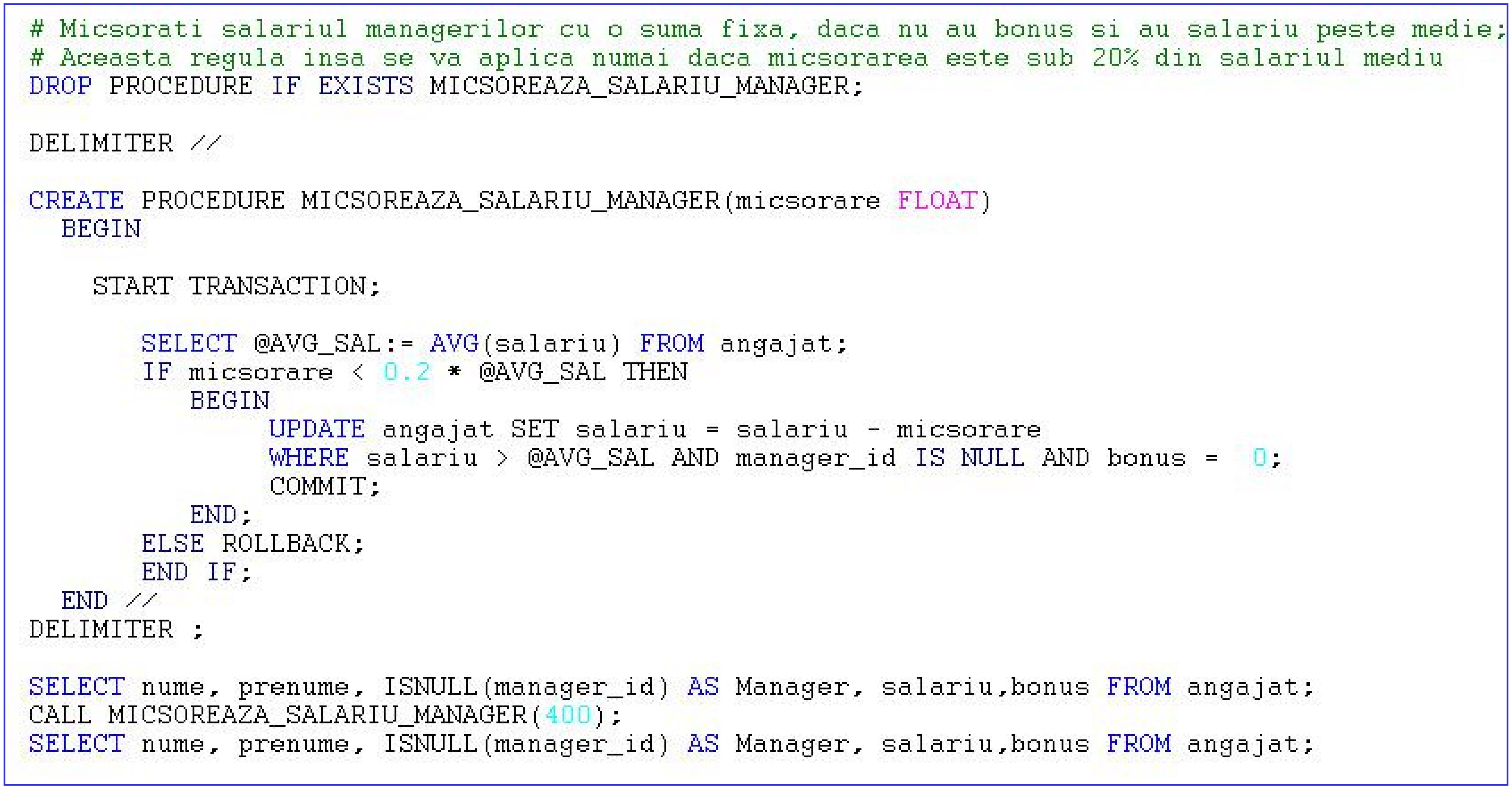
```
mysql> SELECT nume, prenume, salariu FROM angajat;
+------------+---------+---------+
| nume       | prenume | salariu |
+------------+---------+---------+
| Popa       | Ion     |    8000 |
| Popescu    | Maria   |    3000 |
| Marinescu  | Vasile  |    5000 |
| Ionescu    | Andrei  |    3000 |
| Vasilescu  | Ana     |    2000 |
| Dragan     | Dinu    |    2000 |
| Mihailescu | Adrian  |    2500 |
| Teodorescu | Matei   |    2000 |
| Popescu    | Vasile  |    3000 |
| Mateescu   | Dumitru |    3000 |
| Calinescu  | Alin    |    3200 |
| Popescu    | Mihaela |    1500 |
| Enachescu  | Ionel   |    4500 |
| Dediu      | Carmen  |    2700 |
| Antonescu  | Paul    |    2700 |
+------------+---------+---------+
15 rows in set (0.00 sec)
```

```
mysql> CALL MARESTE_SALARIU_CU_SUMA_FIXA(400,9000);
+-------------------------+
| @MAX_SAL:= MAX(salariu) |
+-------------------------+
|                    8400 |
+-------------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT nume, prenume, salariu FROM angajat;
+------------+---------+---------+
| nume       | prenume | salariu |
+------------+---------+---------+
| Popa       | Ion     |    8400 |
| Popescu    | Maria   |    3400 |
| Marinescu  | Vasile  |    5400 |
| Ionescu    | Andrei  |    3400 |
| Vasilescu  | Ana     |    2400 |
| Dragan     | Dinu    |    2400 |
| Mihailescu | Adrian  |    2900 |
| Teodorescu | Matei   |    2400 |
| Popescu    | Vasile  |    3400 |
| Mateescu   | Dumitru |    3400 |
| Calinescu  | Alin    |    3600 |
| Popescu    | Mihaela |    1900 |
| Enachescu  | Ionel   |    4900 |
| Dediu      | Carmen  |    3100 |
| Antonescu  | Paul    |    3100 |
+------------+---------+---------+
15 rows in set (0.00 sec)
```

SQL

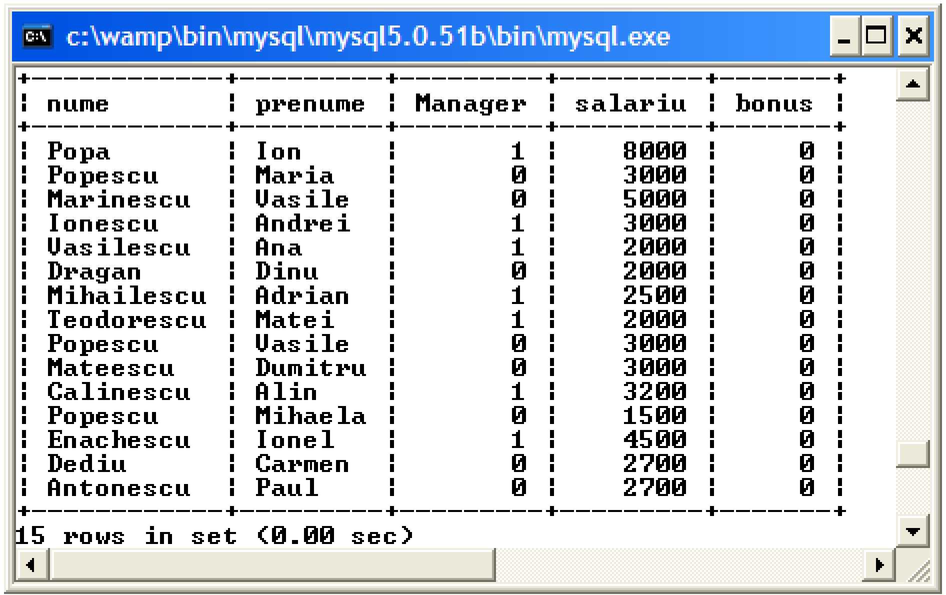
Exemplul 3: Micsorati salariul managerilor cu o suma fixa, daca nu au bonus si au salariu peste medie; aceasta regula insa se va aplica numai daca micsorarea este sub 20% din salariul mediu



```
# Micsorati salariul managerilor cu o suma fixa, daca nu au bonus si au salariu peste medie;
# Aceasta regula insa se va aplica numai daca micsorarea este sub 20% din salariul mediu
DROP PROCEDURE IF EXISTS MICSOREAZA_SALARIU_MANAGER;

DELIMITER //

CREATE PROCEDURE MICSOREAZA_SALARIU_MANAGER(micsorare FLOAT)
  BEGIN

    START TRANSACTION;

       SELECT @AVG_SAL:= AVG(salariu) FROM angajat;
       IF micsorare < 0.2 * @AVG_SAL THEN
          BEGIN
               UPDATE angajat SET salariu = salariu - micsorare
               WHERE salariu > @AVG_SAL AND manager_id IS NULL AND bonus =  0;
               COMMIT;
          END;
       ELSE ROLLBACK;
       END IF;
  END //
DELIMITER ;

SELECT nume, prenume, ISNULL(manager_id) AS Manager, salariu,bonus FROM angajat;
CALL MICSOREAZA_SALARIU_MANAGER(400);
SELECT nume, prenume, ISNULL(manager_id) AS Manager, salariu,bonus FROM angajat;
```

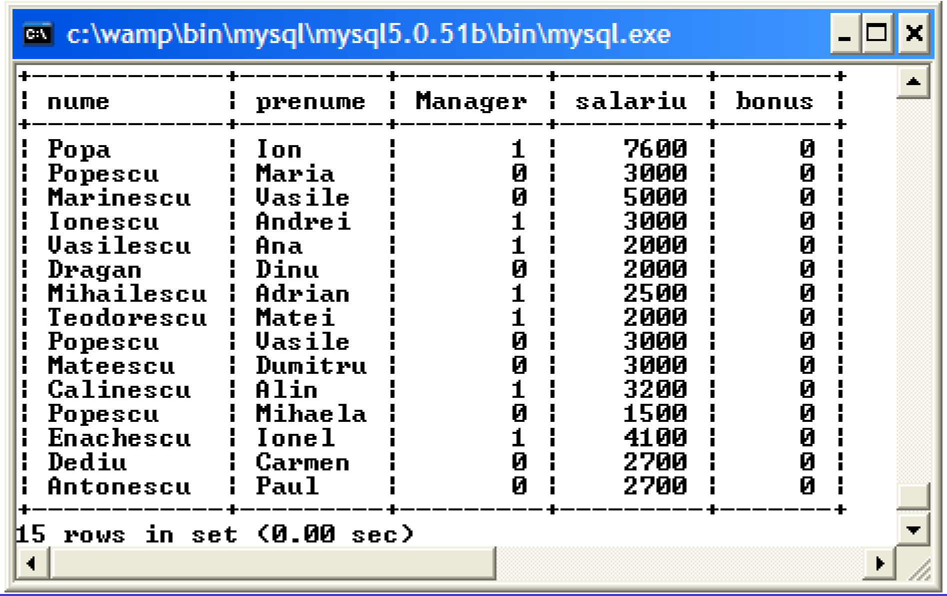
SQL

Inainte de tranzactie:



```
+------------+----------+---------+---------+-------+
| nume       | prenume  | Manager | salariu | bonus |
+------------+----------+---------+---------+-------+
| Popa       | Ion      |       1 |    8000 |     0 |
| Popescu    | Maria    |       0 |    3000 |     0 |
| Marinescu  | Vasile   |       0 |    5000 |     0 |
| Ionescu    | Andrei   |       1 |    3000 |     0 |
| Vasilescu  | Ana      |       1 |    2000 |     0 |
| Dragan     | Dinu     |       0 |    2000 |     0 |
| Mihailescu | Adrian   |       1 |    2500 |     0 |
| Teodorescu | Matei    |       1 |    2000 |     0 |
| Popescu    | Vasile   |       0 |    3000 |     0 |
| Mateescu   | Dumitru  |       0 |    3000 |     0 |
| Calinescu  | Alin     |       1 |    3200 |     0 |
| Popescu    | Mihaela  |       0 |    1500 |     0 |
| Enachescu  | Ionel    |       1 |    4500 |     0 |
| Dediu      | Carmen   |       0 |    2700 |     0 |
| Antonescu  | Paul     |       0 |    2700 |     0 |
+------------+----------+---------+---------+-------+
15 rows in set (0.00 sec)
```

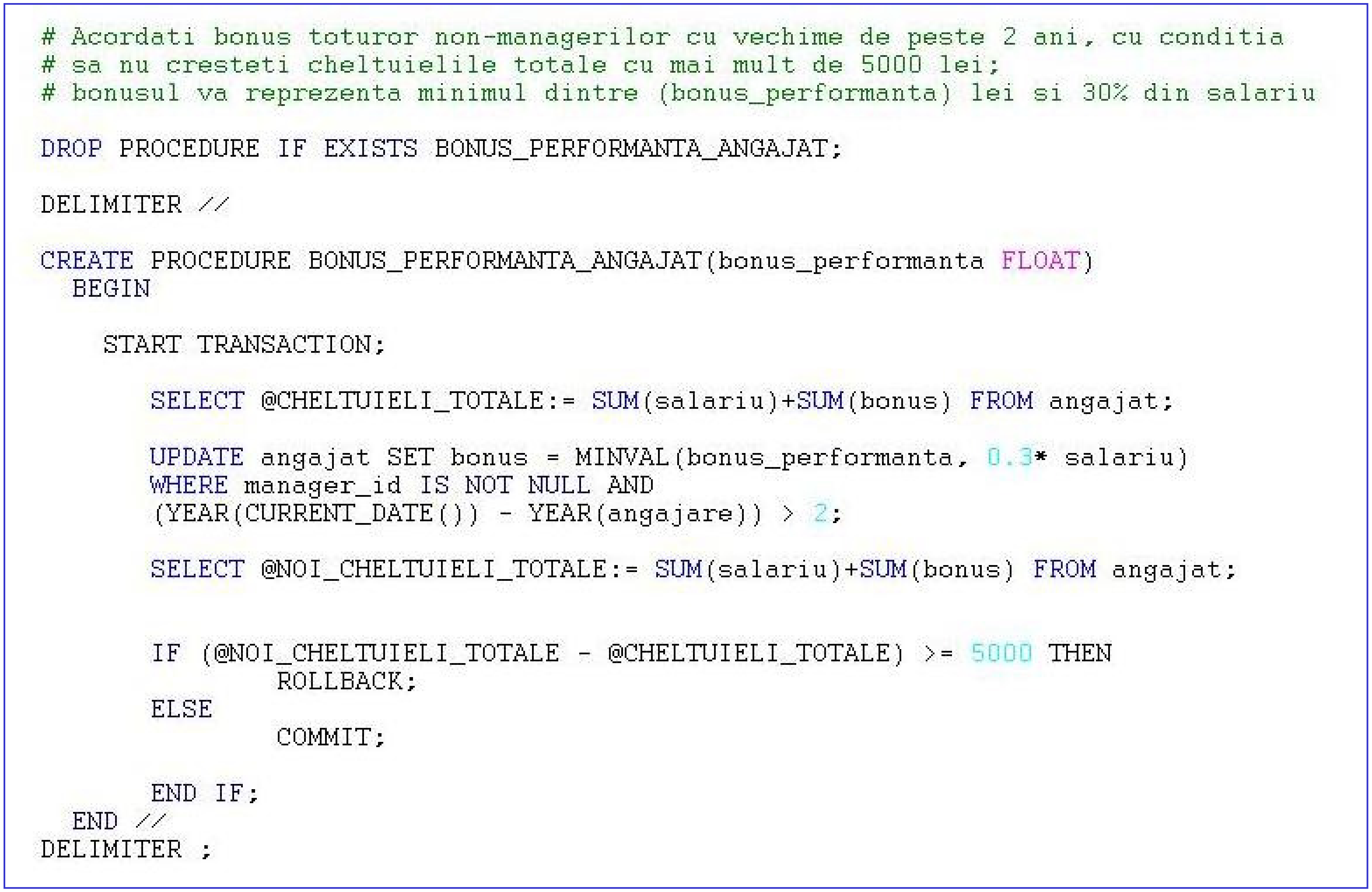
Dupa tranzactie:



```
+------------+----------+---------+---------+-------+
| nume       | prenume  | Manager | salariu | bonus |
+------------+----------+---------+---------+-------+
| Popa       | Ion      |       1 |    7600 |     0 |
| Popescu    | Maria    |       0 |    3000 |     0 |
| Marinescu  | Vasile   |       0 |    5000 |     0 |
| Ionescu    | Andrei   |       1 |    3000 |     0 |
| Vasilescu  | Ana      |       1 |    2000 |     0 |
| Dragan     | Dinu     |       0 |    2000 |     0 |
| Mihailescu | Adrian   |       1 |    2500 |     0 |
| Teodorescu | Matei    |       1 |    2000 |     0 |
| Popescu    | Vasile   |       0 |    3000 |     0 |
| Mateescu   | Dumitru  |       0 |    3000 |     0 |
| Calinescu  | Alin     |       1 |    3200 |     0 |
| Popescu    | Mihaela  |       0 |    1500 |     0 |
| Enachescu  | Ionel    |       1 |    4100 |     0 |
| Dediu      | Carmen   |       0 |    2700 |     0 |
| Antonescu  | Paul     |       0 |    2700 |     0 |
+------------+----------+---------+---------+-------+
15 rows in set (0.00 sec)
```

SQL

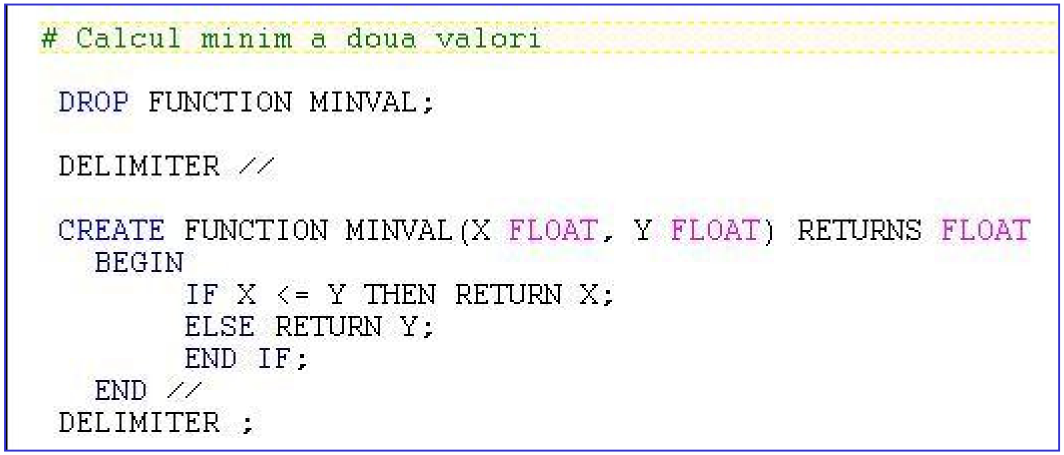
Exemplul 4: Acordati bonus toturor non-managerilor cu vechime de peste 2 ani, cu conditia sa nu cresteti cheltuielile totale cu mai mult de 5000 lei; bonusul va reprezenta minimul dintre (bonus_performanta) lei si 30% din salariu



```
# Acordati bonus toturor non-managerilor cu vechime de peste 2 ani, cu conditia
# sa nu cresteti cheltuielile totale cu mai mult de 5000 lei;
# bonusul va reprezenta minimul dintre (bonus_performanta) lei si 30% din salariu

DROP PROCEDURE IF EXISTS BONUS_PERFORMANTA_ANGAJAT;

DELIMITER //

CREATE PROCEDURE BONUS_PERFORMANTA_ANGAJAT(bonus_performanta FLOAT)
  BEGIN

    START TRANSACTION;

       SELECT @CHELTUIELI_TOTALE:= SUM(salariu)+SUM(bonus) FROM angajat;

       UPDATE angajat SET bonus = MINVAL(bonus_performanta, 0.3* salariu)
       WHERE manager_id IS NOT NULL AND
       (YEAR(CURRENT_DATE()) - YEAR(angajare)) > 2;

       SELECT @NOI_CHELTUIELI_TOTALE:= SUM(salariu)+SUM(bonus) FROM angajat;


       IF (@NOI_CHELTUIELI_TOTALE - @CHELTUIELI_TOTALE) >= 5000 THEN
               ROLLBACK;
       ELSE
               COMMIT;

       END IF;
  END //
DELIMITER ;
```

SQL

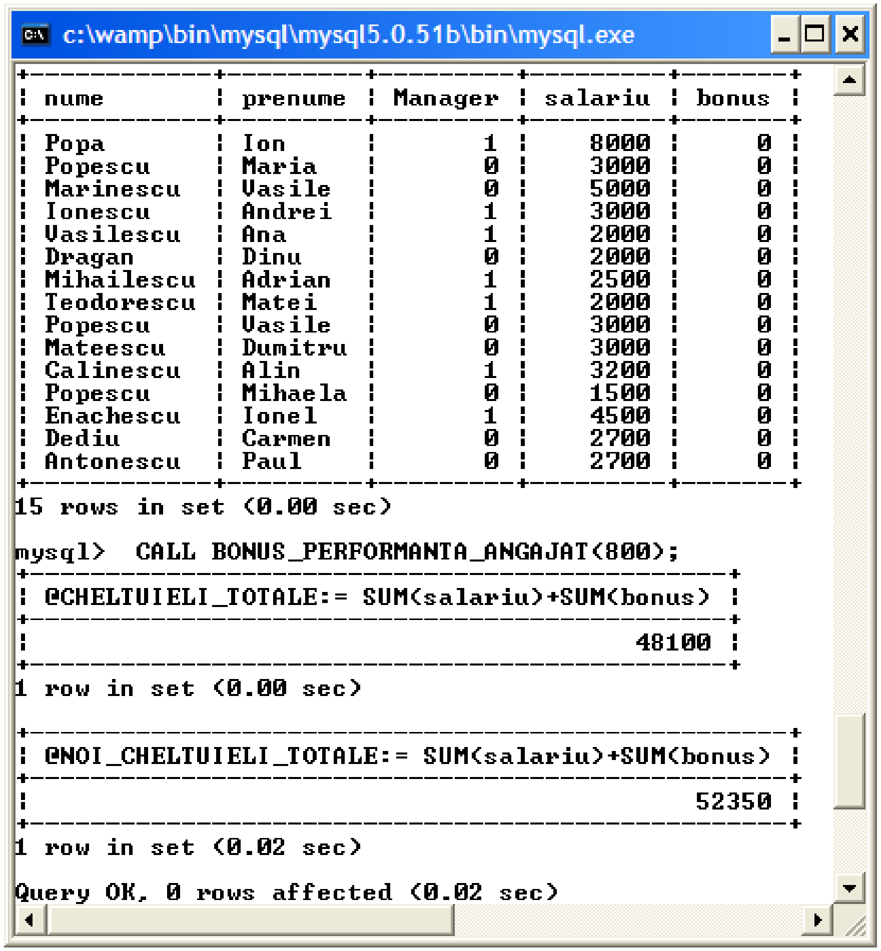
Foloseste si functia de calcul a minimului a doua valori:



```
# Calcul minim a doua valori

 DROP FUNCTION MINVAL;

 DELIMITER //

 CREATE FUNCTION MINVAL(X FLOAT, Y FLOAT) RETURNS FLOAT
   BEGIN
        IF X <= Y THEN RETURN X;
        ELSE RETURN Y;
        END IF;
   END //
 DELIMITER ;
```

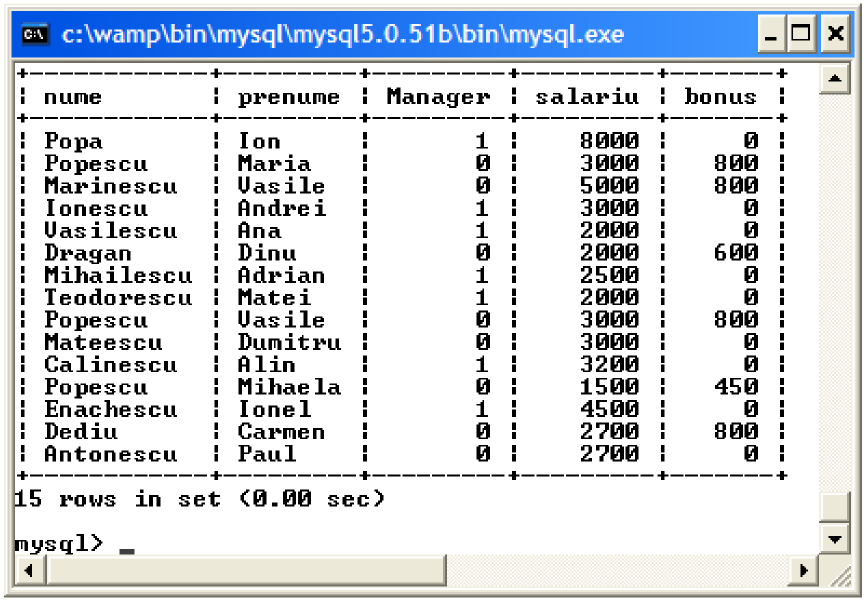
SQL

Inainte de tranzactie si in timpul tranzactie:



```
+------------+----------+---------+---------+-------+
| nume       | prenume  | Manager | salariu | bonus |
+------------+----------+---------+---------+-------+
| Popa       | Ion      |       1 |    8000 |     0 |
| Popescu    | Maria    |       0 |    3000 |     0 |
| Marinescu  | Vasile   |       0 |    5000 |     0 |
| Ionescu    | Andrei   |       1 |    3000 |     0 |
| Vasilescu  | Ana      |       1 |    2000 |     0 |
| Dragan     | Dinu     |       0 |    2000 |     0 |
| Mihailescu | Adrian   |       1 |    2500 |     0 |
| Teodorescu | Matei    |       1 |    2000 |     0 |
| Popescu    | Vasile   |       0 |    3000 |     0 |
| Mateescu   | Dumitru  |       0 |    3000 |     0 |
| Calinescu  | Alin     |       1 |    3200 |     0 |
| Popescu    | Mihaela  |       0 |    1500 |     0 |
| Enachescu  | Ionel    |       1 |    4500 |     0 |
| Dediu      | Carmen   |       0 |    2700 |     0 |
| Antonescu  | Paul     |       0 |    2700 |     0 |
+------------+----------+---------+---------+-------+
15 rows in set (0.00 sec)

mysql>  CALL BONUS_PERFORMANTA_ANGAJAT(800);
+-----------------------------------------------+
| @CHELTUIELI_TOTALE:= SUM(salariu)+SUM(bonus) |
+-----------------------------------------------+
|                                         48100 |
+-----------------------------------------------+
1 row in set (0.00 sec)

+---------------------------------------------------+
| @NOI_CHELTUIELI_TOTALE:= SUM(salariu)+SUM(bonus) |
+---------------------------------------------------+
|                                             52350 |
+---------------------------------------------------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)
```

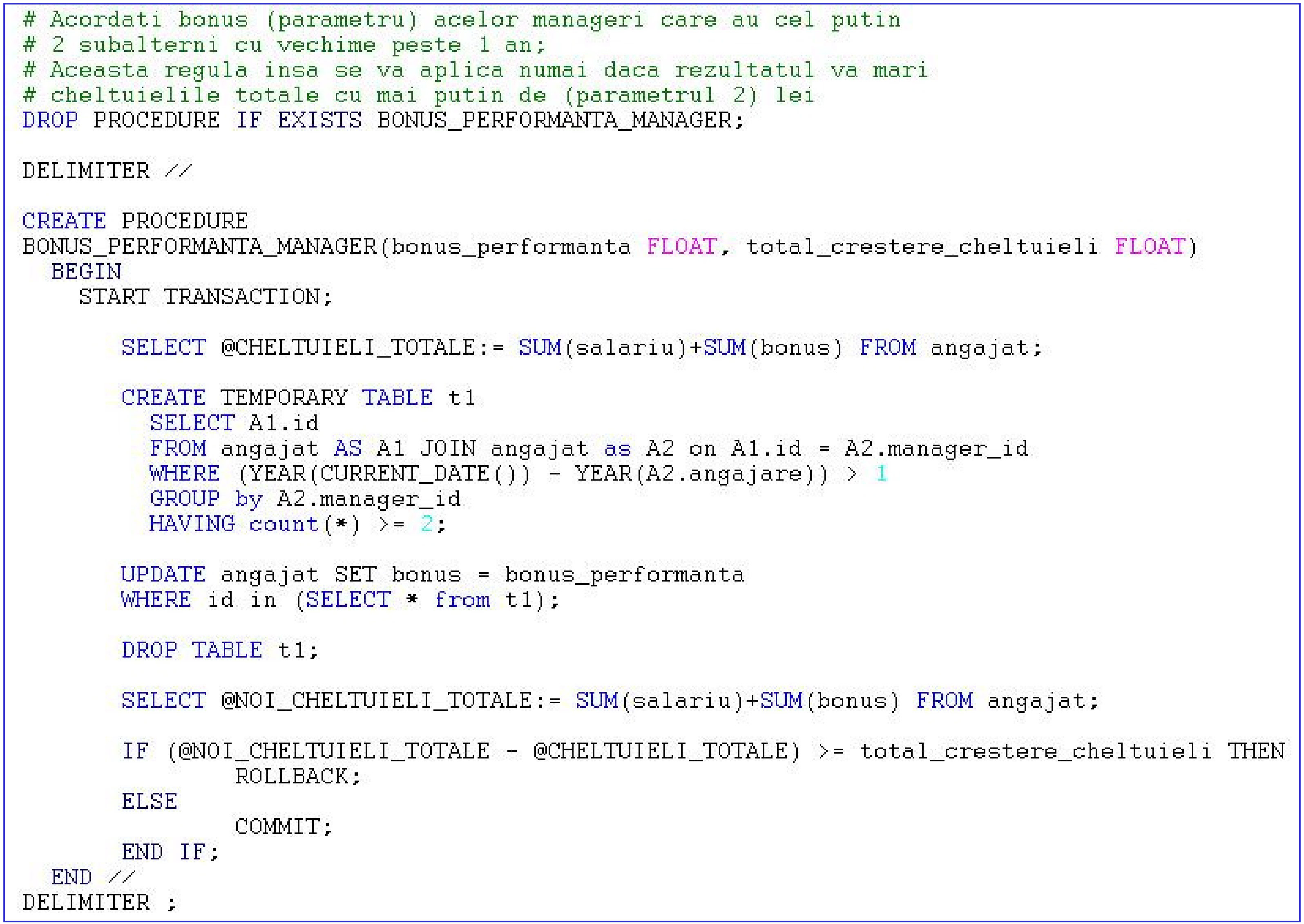
Dupa tranzactie:



```
+------------+----------+---------+---------+-------+
| nume       | prenume  | Manager | salariu | bonus |
+------------+----------+---------+---------+-------+
| Popa       | Ion      |       1 |    8000 |     0 |
| Popescu    | Maria    |       0 |    3000 |   800 |
| Marinescu  | Vasile   |       0 |    5000 |   800 |
| Ionescu    | Andrei   |       1 |    3000 |     0 |
| Vasilescu  | Ana      |       1 |    2000 |     0 |
| Dragan     | Dinu     |       0 |    2000 |   600 |
| Mihailescu | Adrian   |       1 |    2500 |     0 |
| Teodorescu | Matei    |       1 |    2000 |     0 |
| Popescu    | Vasile   |       0 |    3000 |   800 |
| Mateescu   | Dumitru  |       0 |    3000 |     0 |
| Calinescu  | Alin     |       1 |    3200 |     0 |
| Popescu    | Mihaela  |       0 |    1500 |   450 |
| Enachescu  | Ionel    |       1 |    4500 |     0 |
| Dediu      | Carmen   |       0 |    2700 |   800 |
| Antonescu  | Paul     |       0 |    2700 |     0 |
+------------+----------+---------+---------+-------+
15 rows in set (0.00 sec)

mysql> _
```

SQL

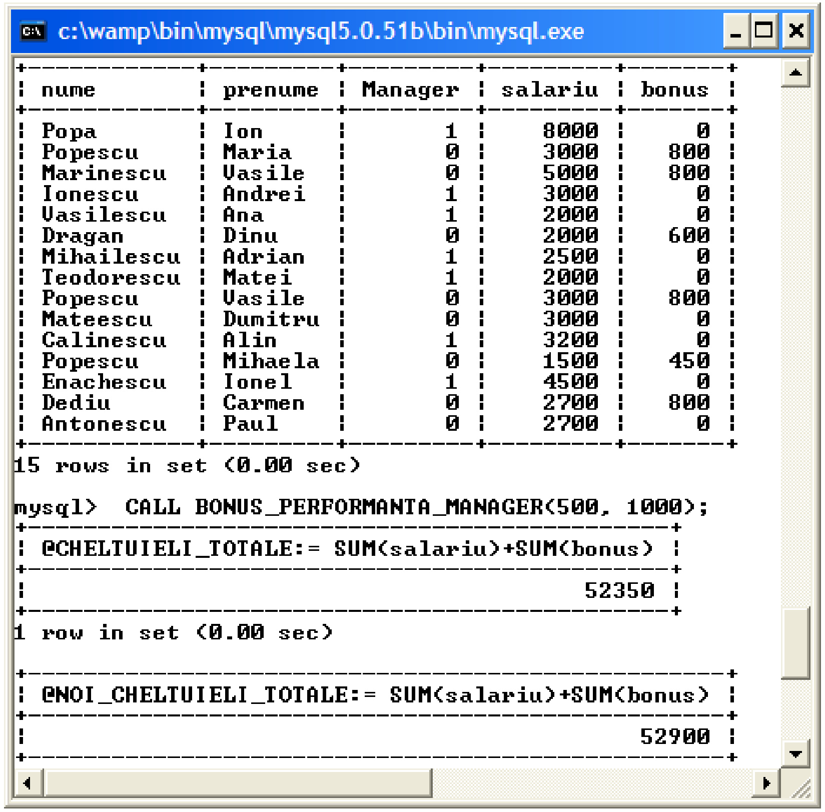
Exemplul 5: Acordati bonus (parametru) acelor manageri care au cel putin 2 subalterni cu vechime peste 1 an; Aceasta regula insa se va aplica numai daca rezultatul va maricheltuielile totale cu mai putin de (parametrul 2) lei



```
# Acordati bonus (parametru) acelor manageri care au cel putin
# 2 subalterni cu vechime peste 1 an;
# Aceasta regula insa se va aplica numai daca rezultatul va mari
# cheltuielile totale cu mai putin de (parametrul 2) lei
DROP PROCEDURE IF EXISTS BONUS_PERFORMANTA_MANAGER;

DELIMITER //

CREATE PROCEDURE
BONUS_PERFORMANTA_MANAGER(bonus_performanta FLOAT, total_crestere_cheltuieli FLOAT)
  BEGIN
    START TRANSACTION;

        SELECT @CHELTUIELI_TOTALE:= SUM(salariu)+SUM(bonus) FROM angajat;

        CREATE TEMPORARY TABLE t1
          SELECT A1.id
          FROM angajat AS A1 JOIN angajat as A2 on A1.id = A2.manager_id
          WHERE (YEAR(CURRENT_DATE()) - YEAR(A2.angajare)) > 1
          GROUP by A2.manager_id
          HAVING count(*) >= 2;

        UPDATE angajat SET bonus = bonus_performanta
        WHERE id in (SELECT * from t1);

        DROP TABLE t1;

        SELECT @NOI_CHELTUIELI_TOTALE:= SUM(salariu)+SUM(bonus) FROM angajat;

        IF (@NOI_CHELTUIELI_TOTALE - @CHELTUIELI_TOTALE) >= total_crestere_cheltuieli THEN
                ROLLBACK;
        ELSE
                COMMIT;
        END IF;
  END //
DELIMITER ;
```

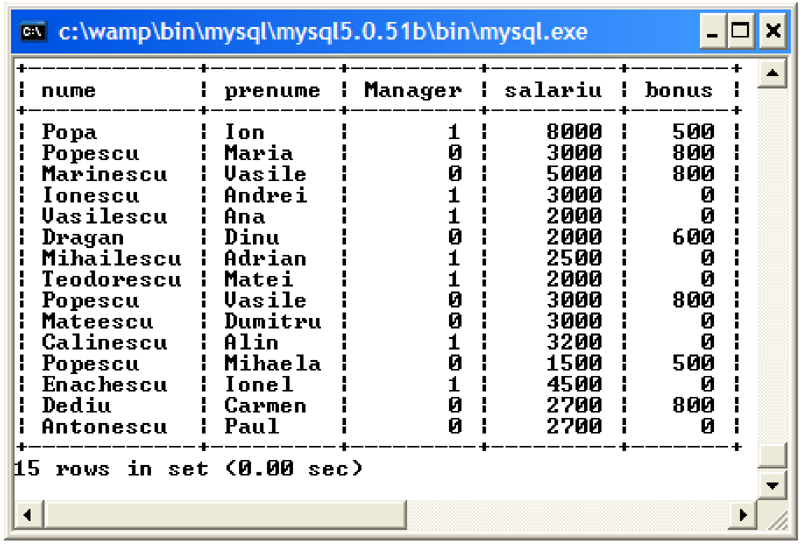
SQL

Inainte de tranzactie si in timpul tranzactie:



```
+------------+----------+---------+---------+-------+
| nume       | prenume  | Manager | salariu | bonus |
+------------+----------+---------+---------+-------+
| Popa       | Ion      |       1 |    8000 |     0 |
| Popescu    | Maria    |       0 |    3000 |   800 |
| Marinescu  | Vasile   |       0 |    5000 |   800 |
| Ionescu    | Andrei   |       1 |    3000 |     0 |
| Vasilescu  | Ana      |       1 |    2000 |     0 |
| Dragan     | Dinu     |       0 |    2000 |   600 |
| Mihailescu | Adrian   |       1 |    2500 |     0 |
| Teodorescu | Matei    |       1 |    2000 |     0 |
| Popescu    | Vasile   |       0 |    3000 |   800 |
| Mateescu   | Dumitru  |       0 |    3000 |     0 |
| Calinescu  | Alin     |       1 |    3200 |     0 |
| Popescu    | Mihaela  |       0 |    1500 |   450 |
| Enachescu  | Ionel    |       1 |    4500 |     0 |
| Dediu      | Carmen   |       0 |    2700 |   800 |
| Antonescu  | Paul     |       0 |    2700 |     0 |
+------------+----------+---------+---------+-------+
15 rows in set (0.00 sec)

mysql>  CALL BONUS_PERFORMANTA_MANAGER(500, 1000);
+-----------------------------------------+
| @CHELTUIELI_TOTALE:= SUM(salariu)+SUM(bonus) |
+-----------------------------------------+
|                                   52350 |
+-----------------------------------------+
1 row in set (0.00 sec)

+---------------------------------------------+
| @NOI_CHELTUIELI_TOTALE:= SUM(salariu)+SUM(bonus) |
+---------------------------------------------+
|                                       52900 |
+---------------------------------------------+
```

Dupa tranzactie:



```
+------------+----------+---------+---------+-------+
| nume       | prenume  | Manager | salariu | bonus |
+------------+----------+---------+---------+-------+
| Popa       | Ion      |       1 |    8000 |   500 |
| Popescu    | Maria    |       0 |    3000 |   800 |
| Marinescu  | Vasile   |       0 |    5000 |   800 |
| Ionescu    | Andrei   |       1 |    3000 |     0 |
| Vasilescu  | Ana      |       1 |    2000 |     0 |
| Dragan     | Dinu     |       0 |    2000 |   600 |
| Mihailescu | Adrian   |       1 |    2500 |     0 |
| Teodorescu | Matei    |       1 |    2000 |     0 |
| Popescu    | Vasile   |       0 |    3000 |   800 |
| Mateescu   | Dumitru  |       0 |    3000 |     0 |
| Calinescu  | Alin     |       1 |    3200 |     0 |
| Popescu    | Mihaela  |       0 |    1500 |   500 |
| Enachescu  | Ionel    |       1 |    4500 |     0 |
| Dediu      | Carmen   |       0 |    2700 |   800 |
| Antonescu  | Paul     |       0 |    2700 |     0 |
+------------+----------+---------+---------+-------+
15 rows in set (0.00 sec)
```