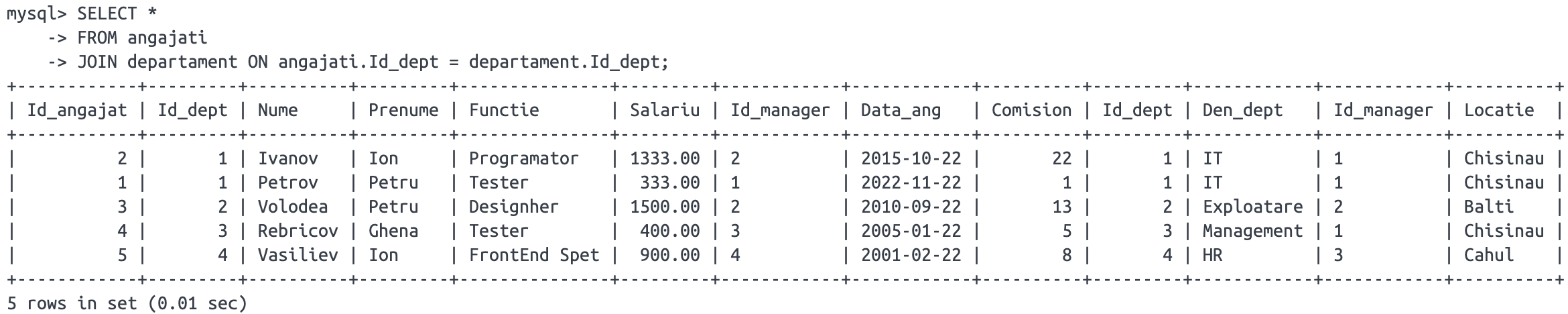
SARCINI CU JOIN PENTRU BAZA DE DATE intreprindere

1. **Sa se afiseze detalii complete despre angajatii si departamentele din intreprindere.**

**SELECT \***

**FROM angajati**

**JOIN departament ON angajati.Id\_dept = departament.Id\_dept;**

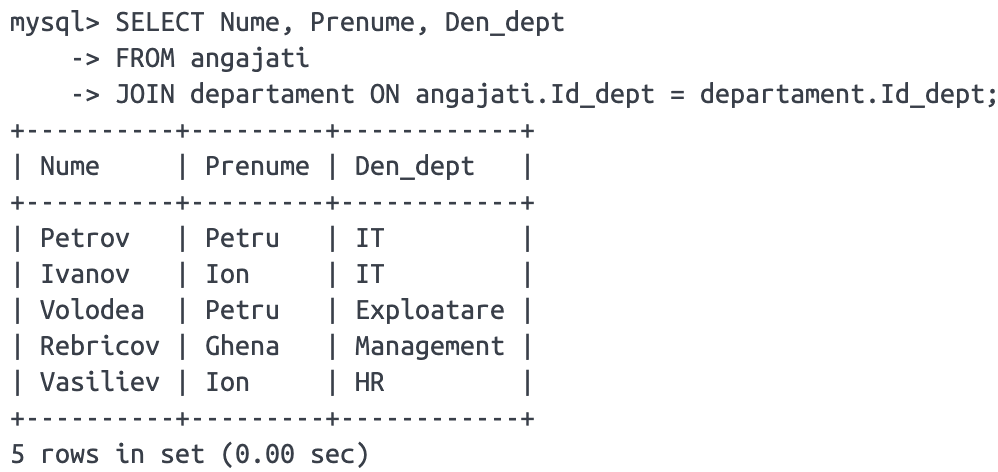


1. **Sa se afiseze numele, prenumele si denumirea departamentului pentru fiecare angajat.**

**SELECT Nume, Prenume, Den\_dept**

**FROM angajati**

**JOIN departament ON angajati.Id\_dept = departament.Id\_dept;**



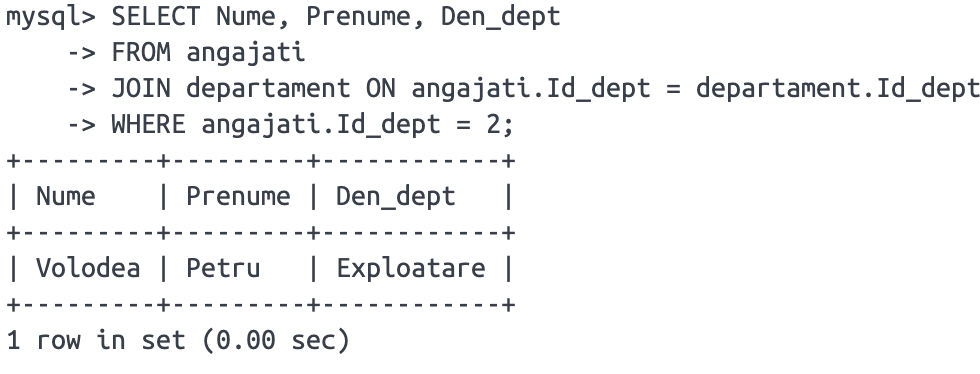
1. **Sa se afiseze numele, prenumele si denumirea departamentului pentru angajatii din departamentul cu ID-ul 2.**

**SELECT Nume, Prenume, Den\_dept**

**FROM angajati**

**JOIN departament ON angajati.Id\_dept = departament.Id\_dept**

**WHERE angajati.Id\_dept = 2;**



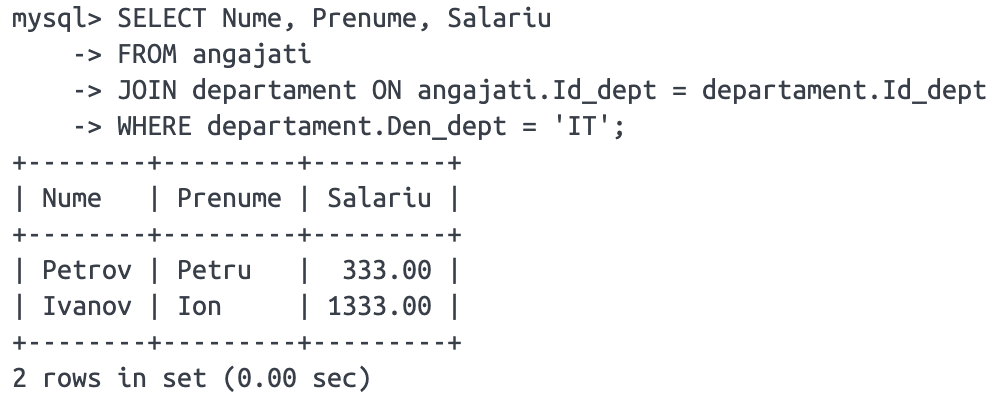
1. **Sa se afiseze numele, prenumele si salariul angajatilor din departamentul cu denumirea 'IT'.**

**SELECT Nume, Prenume, Salariu**

**FROM angajati**

**JOIN departament ON angajati.Id\_dept = departament.Id\_dept**

**WHERE departament.Den\_dept = 'IT';**



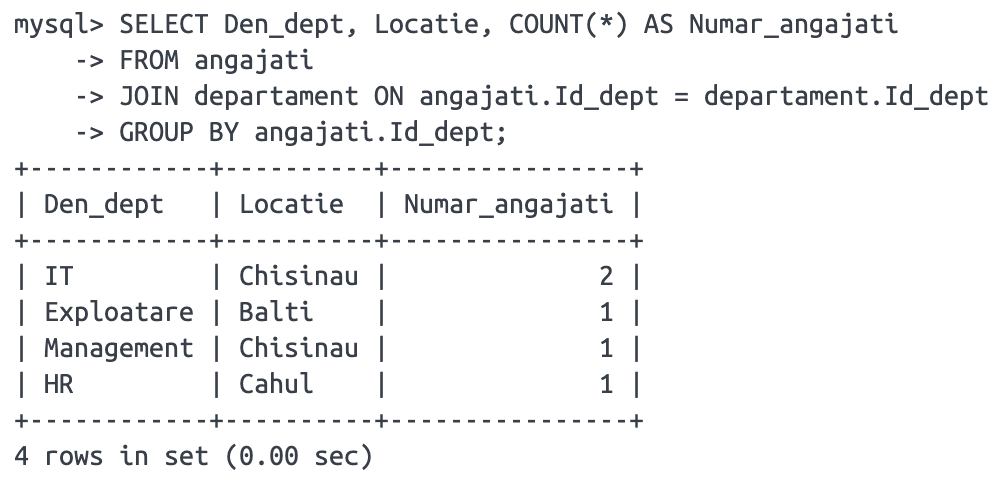
1. **Sa se afiseze denumirea departamentului, locatia si numarul total de angajati pentru fiecare departament.**

**SELECT Den\_dept, Locatie, COUNT(\*) AS Numar\_angajati**

**FROM angajati**

**JOIN departament ON angajati.Id\_dept = departament.Id\_dept**

**GROUP BY angajati.Id\_dept;**



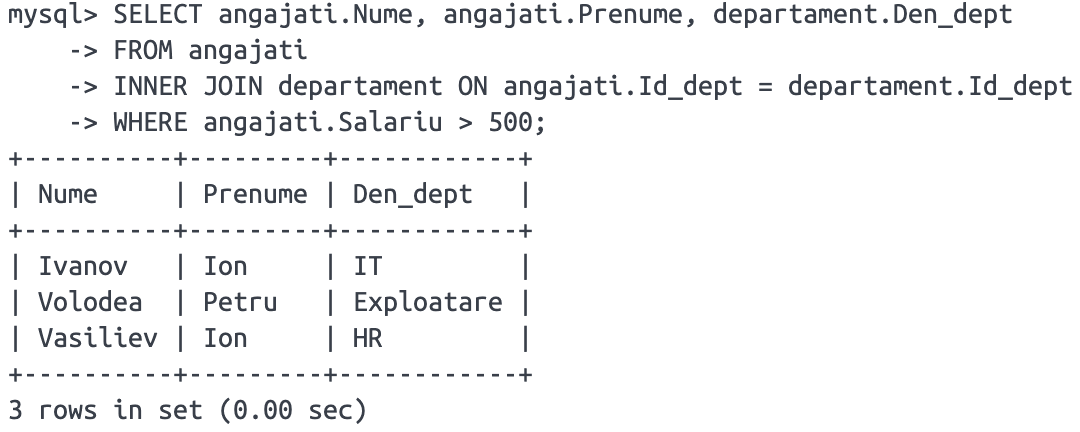
1. **Sa se afiseze numele angajaților și denumirea departamentului unde au un salariu mai mare de 500.**

**SELECT angajati.Nume, angajati.Prenume, departament.Den\_dept**

**FROM angajati**

**INNER JOIN departament ON angajati.Id\_dept = departament.Id\_dept**

**WHERE angajati.Salariu > 500;**

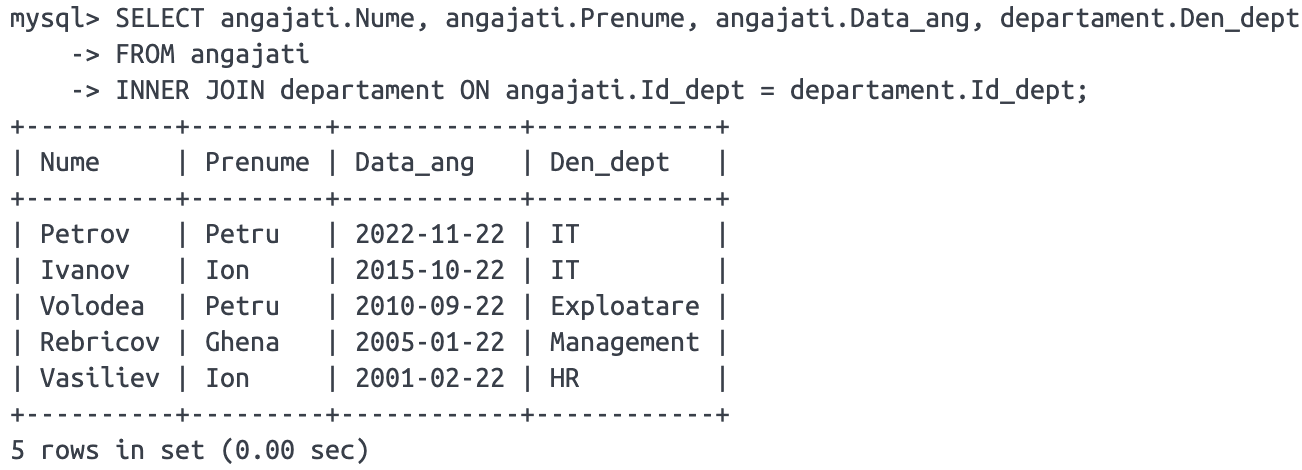


1. **Sa se afiseze numele angajaților și data la care au fost angajați, împreună cu denumirea departamentului.**

**SELECT angajati.Nume, angajati.Prenume, angajati.Data\_ang, departament.Den\_dept**

**FROM angajati**

**INNER JOIN departament ON angajati.Id\_dept = departament.Id\_dept;**

****

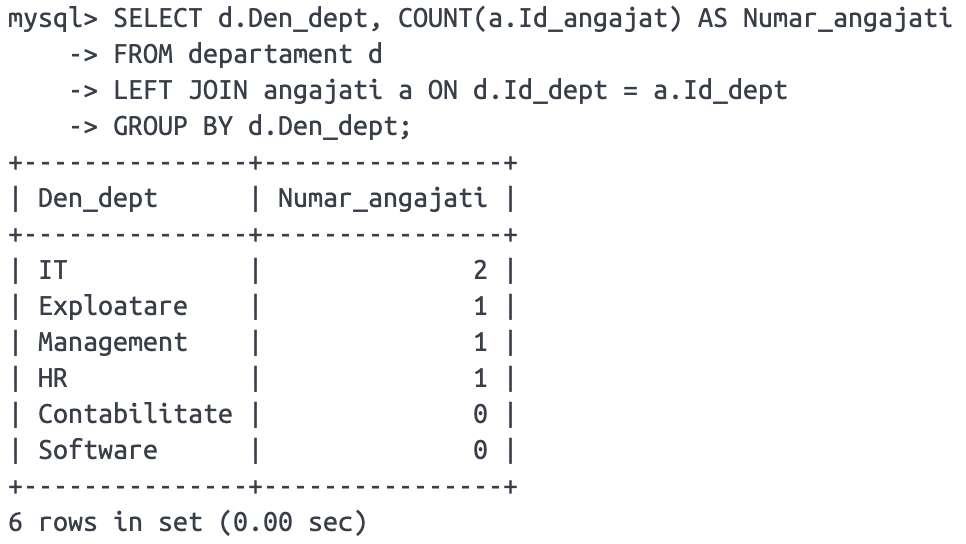
1. **Sa se afiseze denumirea departamentului și numărul total de angajați pentru fiecare departament.**

**SELECT d.Den\_dept, COUNT(a.Id\_angajat) AS Numar\_angajati**

**FROM departament d**

**LEFT JOIN angajati a ON d.Id\_dept = a.Id\_dept**

**GROUP BY d.Den\_dept;**

****

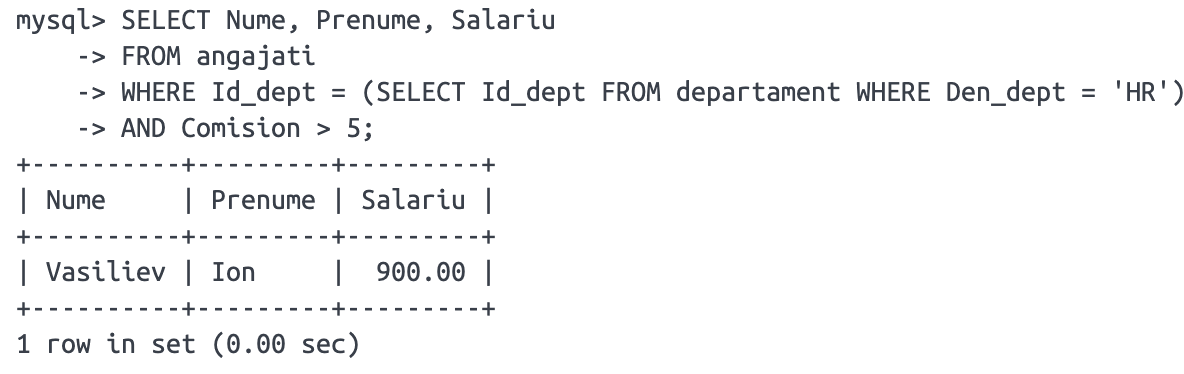
1. **Sa se afiseze numele, prenumele și salariul angajaților pentru departamentul "HR" care au comisionul mai mare de 5.**

**SELECT Nume, Prenume, Salariu**

**FROM angajati**

**WHERE Id\_dept = (SELECT Id\_dept FROM departament WHERE Den\_dept = 'HR')**

**AND Comision > 5;**

****

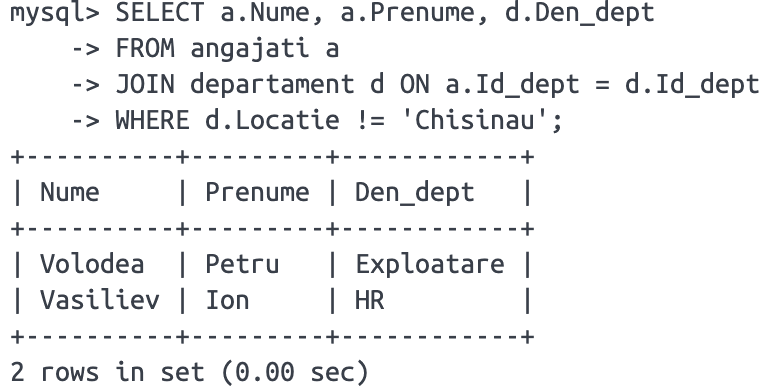
1. **Sa se afiseze numele, prenumele și denumirea departamentului pentru toți angajații care lucrează în afara orașului "Chisinau".**

**SELECT a.Nume, a.Prenume, d.Den\_dept**

**FROM angajati a**

**JOIN departament d ON a.Id\_dept = d.Id\_dept**

**WHERE d.Locatie != 'Chisinau';**

****

SARCINI CU INTEROGARI IMBRICATE PENTRU BAZA DE DATE intreprindere

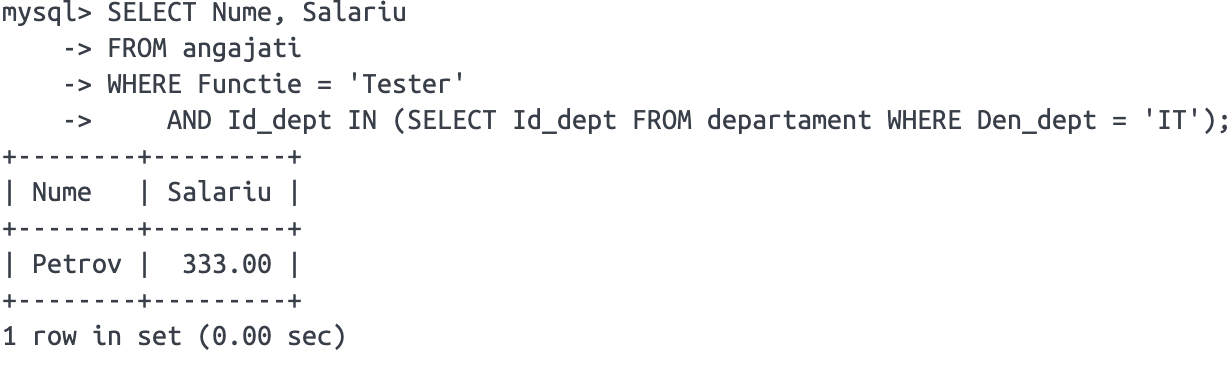
1. **Sa se afiseze toate numele si salariile angajatilor care au functia de Tester si lucreaza in departamentul cu denumirea "IT".**

**SELECT Nume, Salariu**

**FROM angajati**

**WHERE Functie = 'Tester'**

**AND Id\_dept IN (SELECT Id\_dept FROM departament WHERE Den\_dept = 'IT');**

****

1. **Gasiti denumirile departamentelor care au angajati care au salarii mai mari de 1000 si au comision mai mic decat 25.**

**SELECT DISTINCT Den\_dept**

**FROM departament**

**WHERE Id\_dept IN (**

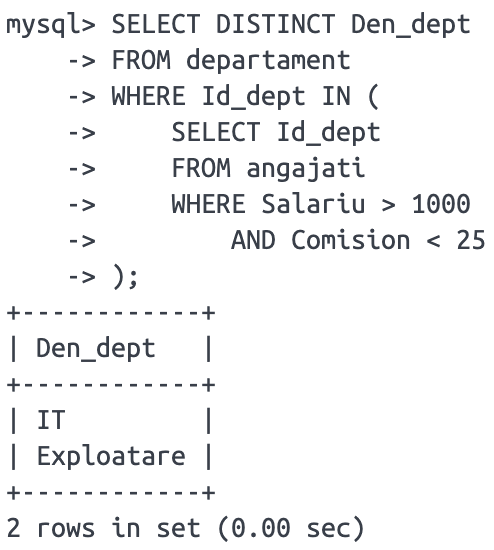
**SELECT Id\_dept**

**FROM angajati**

**WHERE Salariu > 1000**

**AND Comision < 25**

**);**

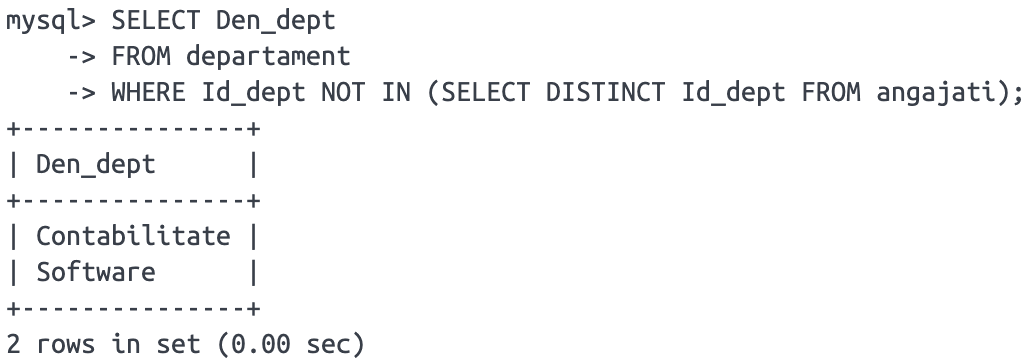
****

1. **Aflati denumirile departamentelor care nu au niciun angajat.**

**SELECT Den\_dept**

**FROM departament**

**WHERE Id\_dept NOT IN (SELECT DISTINCT Id\_dept FROM angajati);**

****

1. **Aflati denumirile departamentelor care au cel putin un angajat cu functia de Programator si unul cu functia de Tester.**

**SELECT DISTINCT Den\_dept**

**FROM departament**

**WHERE Id\_dept IN (**

**SELECT Id\_dept**

**FROM angajati**

**WHERE Functie = 'Programator'**

**)**

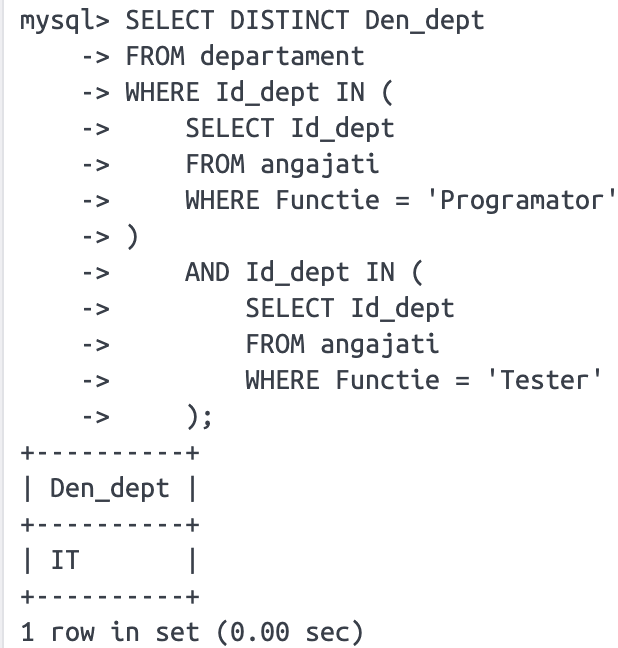
**AND Id\_dept IN (**

**SELECT Id\_dept**

**FROM angajati**

**WHERE Functie = 'Tester'**

**);**

****

1. **Gasiti departamentul cu cel mai mare salariu mediu al angajatilor.**

**SELECT Den\_dept**

**FROM departament**

**WHERE Id\_dept = (**

**SELECT Id\_dept**

**FROM (**

**SELECT Id\_dept, AVG(Salariu) AS SalariuMediu**

**FROM angajati**

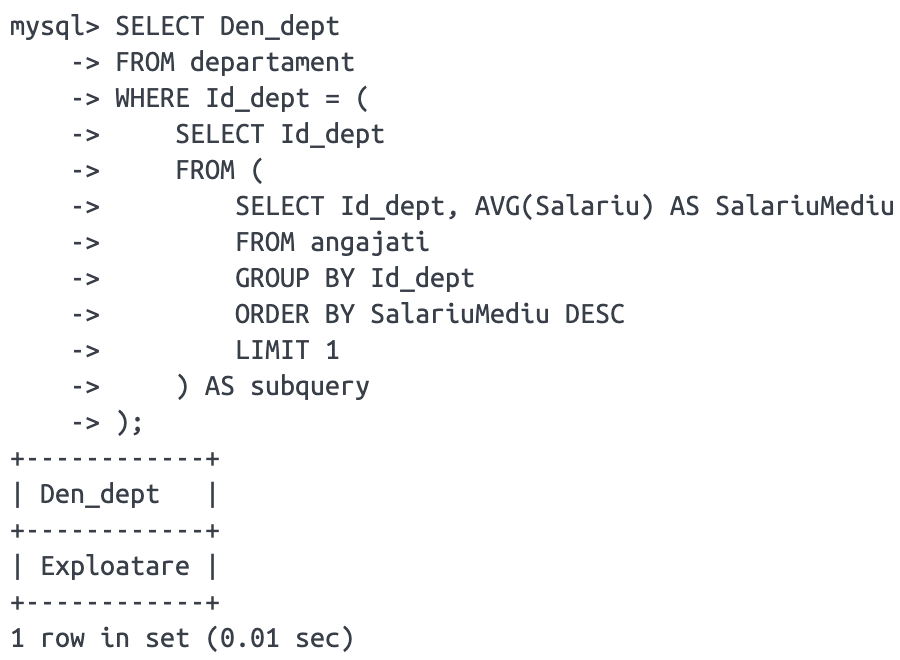
**GROUP BY Id\_dept**

**ORDER BY SalariuMediu DESC**

**LIMIT 1**

**) AS subquery**

**);**

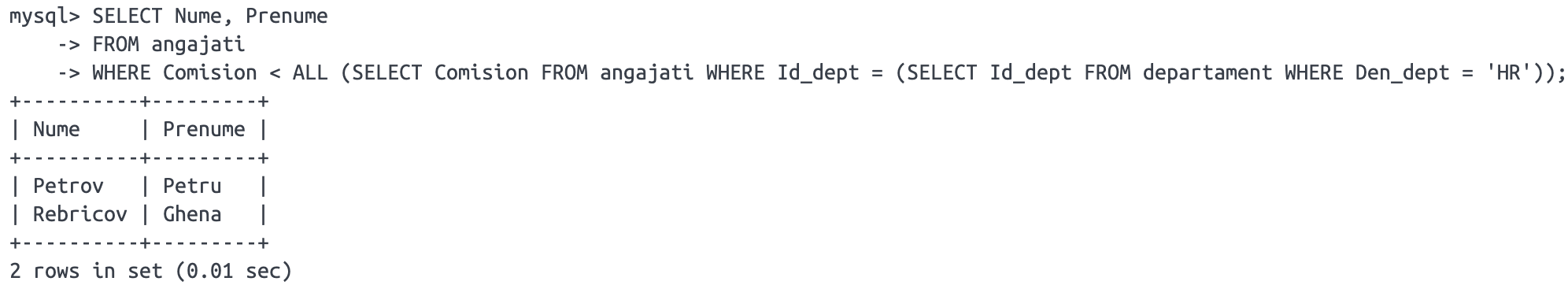
****

1. **Aflati angajatii care au comisioane mai mici decat oricare alt angajat din departamentul cu denumirea "HR".**

**SELECT Nume, Prenume**

**FROM angajati**

**WHERE Comision < ALL (SELECT Comision FROM angajati WHERE Id\_dept = (SELECT Id\_dept FROM departament WHERE Den\_dept = 'HR'));**

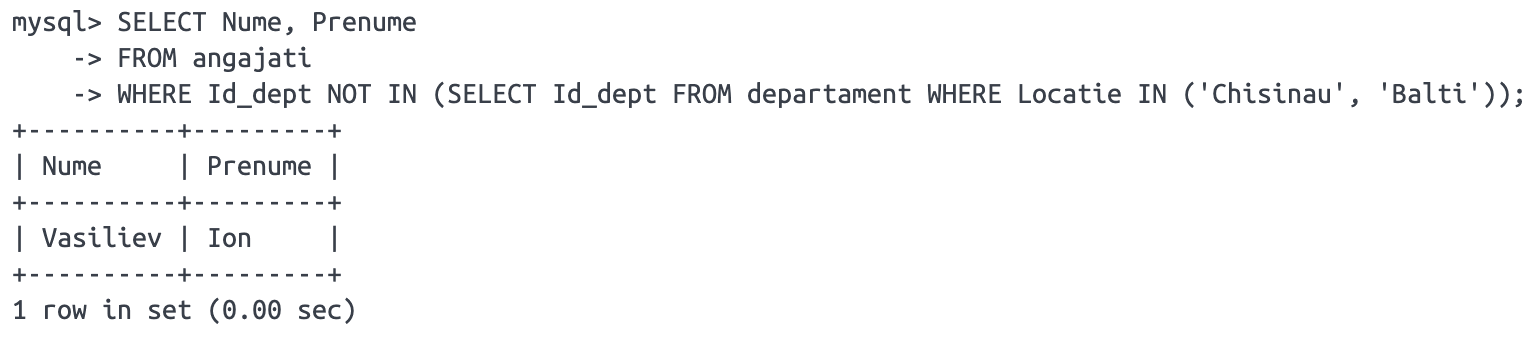
****

1. **Afisati angajatii care nu lucreaza in departamentele cu locatia "Chisinau" sau "Balti".**

**SELECT Nume, Prenume**

**FROM angajati**

**WHERE Id\_dept NOT IN (SELECT Id\_dept FROM departament WHERE Locatie IN ('Chisinau', 'Balti'));**

****

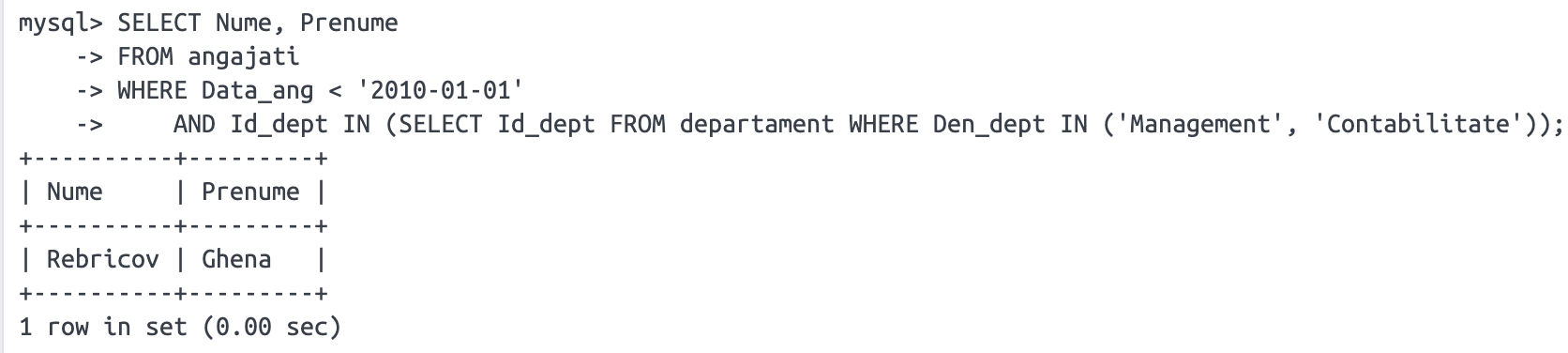
1. **Afișați angajații care au fost angajați înainte de data de 1 ianuarie 2010 în departamentele cu denumirea "Management" sau "Contabilitate".**

**SELECT Nume, Prenume**

**FROM angajati**

**WHERE Data\_ang < '2010-01-01'**

**AND Id\_dept IN (SELECT Id\_dept FROM departament WHERE Den\_dept IN ('Management', 'Contabilitate'));**

****

1. **Afișați angajații care au colegi în același departament cu salarii mai mari de 1000.**

**SELECT Nume, Prenume**

**FROM angajati a1**

**WHERE EXISTS (**

**SELECT 1**

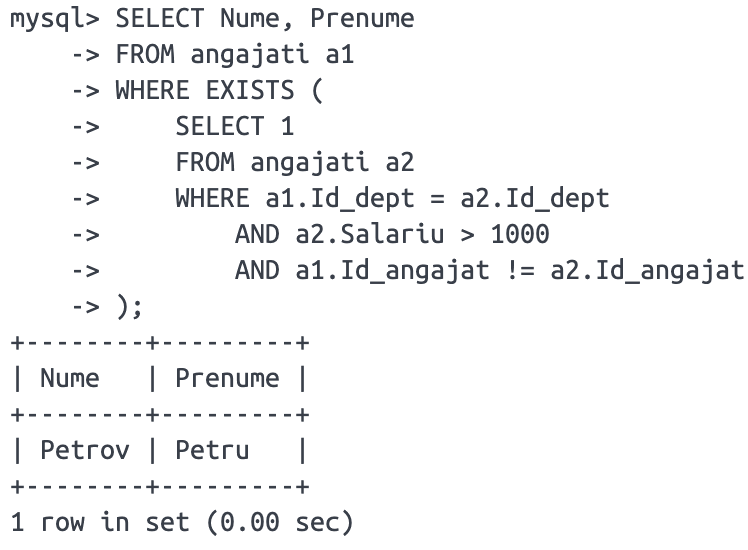
**FROM angajati a2**

**WHERE a1.Id\_dept = a2.Id\_dept**

**AND a2.Salariu > 1000**

**AND a1.Id\_angajat != a2.Id\_angajat**

**);**

****

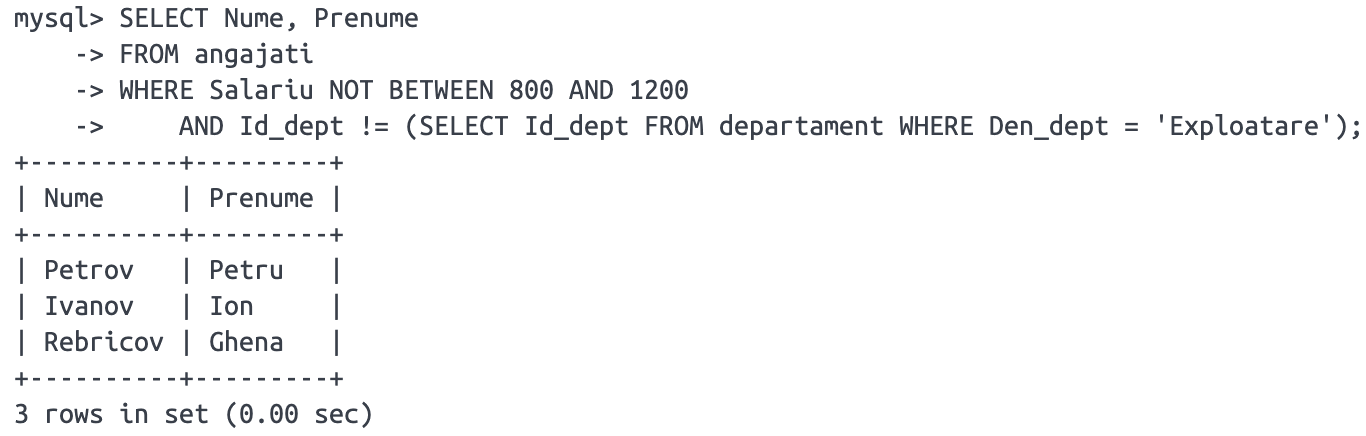
1. **Găsiți angajații care au salariul în afara intervalului [800, 1200] și nu lucrează în departamentul "Exploatare".**

**SELECT Nume, Prenume**

**FROM angajati**

**WHERE Salariu NOT BETWEEN 800 AND 1200**

**AND Id\_dept != (SELECT Id\_dept FROM departament WHERE Den\_dept = 'Exploatare');**

****

SARCINI CU UNION , INTERSECT ,EXCEPT PENTRU BAZA DE DATE intreprindere

1. **Selecteaza toate numele si prenumele angajatilor si departamentelor , cu conditia ca numele departamentului sa inceapa cu litera "I" sau "S".**

**SELECT Nume, Prenume, 'Angajat' AS Tip**

**FROM angajati**

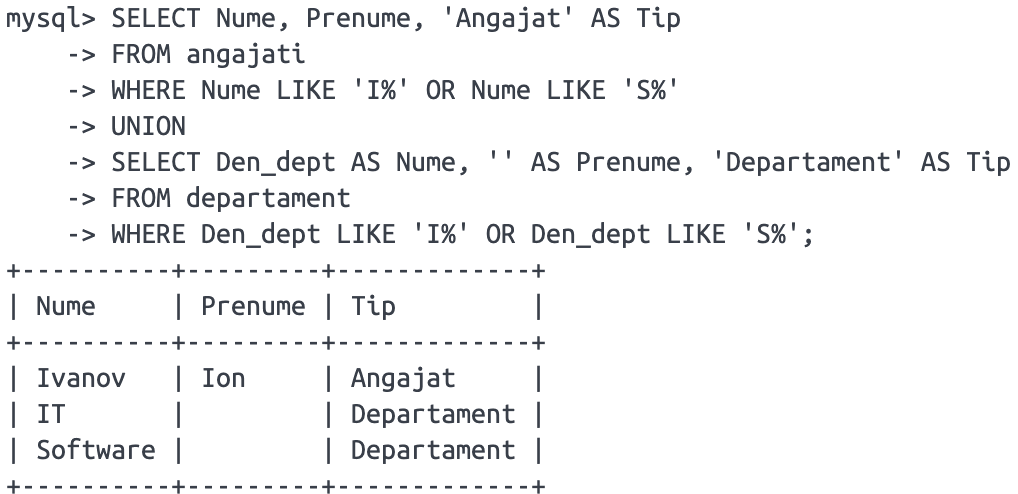
**WHERE Nume LIKE 'I%' OR Nume LIKE 'S%'**

**UNION**

**SELECT Den\_dept AS Nume, '' AS Prenume, 'Departament' AS Tip**

**FROM departament**

**WHERE Den\_dept LIKE 'I%' OR Den\_dept LIKE 'S%';**

****

**2. Selecteaza numele si prenumele angajatilor care nu sunt manageri de departamente.**

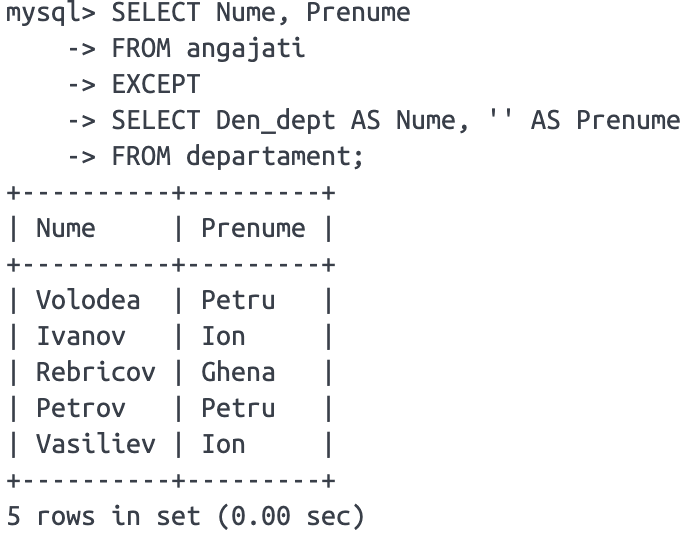
**SELECT Nume, Prenume**

**FROM angajati**

**EXCEPT**

**SELECT Den\_dept AS Nume, '' AS Prenume**

**FROM departament;**

****

1. **Selecteaza numele si prenumele angajatilor care nu sunt manageri de departamente.**

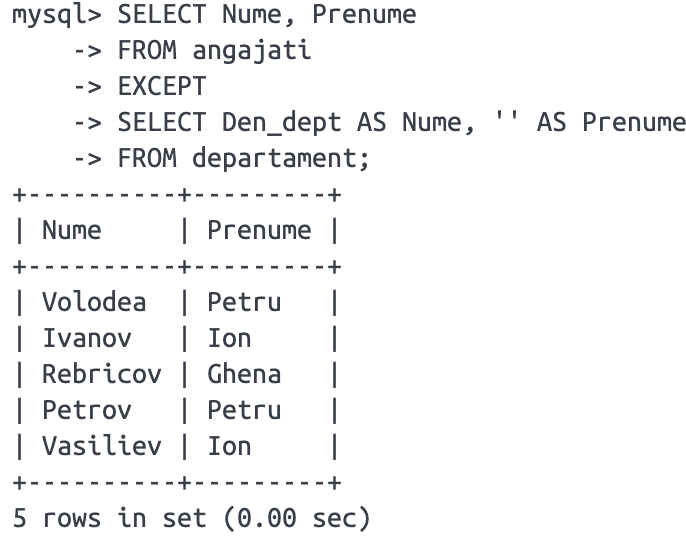
**SELECT Nume, Prenume**

**FROM angajati**

**EXCEPT**

**SELECT Den\_dept AS Nume, '' AS Prenume**

**FROM departament;**

****

1. **Selecteaza angajatii care au salariu mai mare de 1000 , si lucreaza in departamentul cu locatie "Chisinau".**

**SELECT Nume, Prenume, Salariu, Den\_dept**

**FROM angajati a**

**JOIN departament d ON a.Id\_dept = d.Id\_dept**

**WHERE Salariu > 1000**

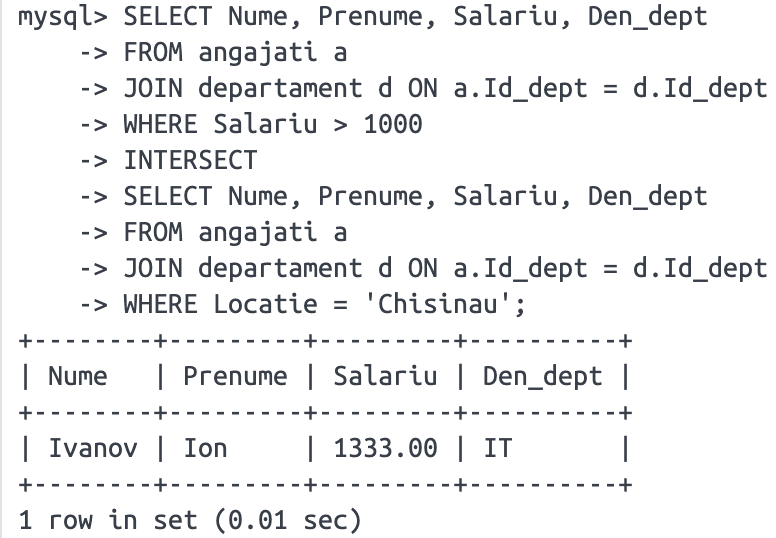
**INTERSECT**

**SELECT Nume, Prenume, Salariu, Den\_dept**

**FROM angajati a**

**JOIN departament d ON a.Id\_dept = d.Id\_dept**

**WHERE Locatie = 'Chisinau';**

****

1. **Selecteaza numele, prenumele, si denumirea departamentului pentru toti angajatii si departamentele.**

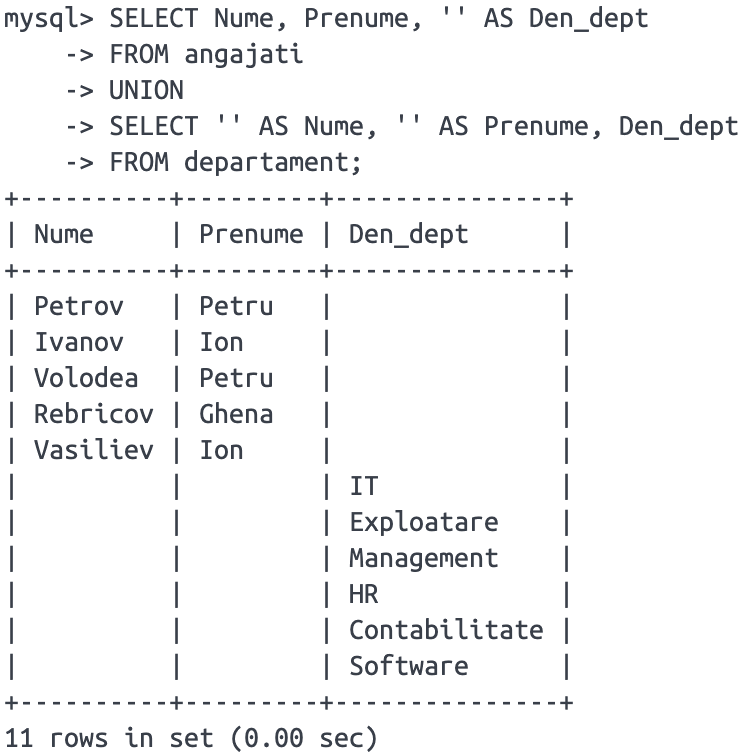
**SELECT Nume, Prenume, '' AS Den\_dept**

**FROM angajati**

**UNION**

**SELECT '' AS Nume, '' AS Prenume, Den\_dept**

**FROM departament;**

****

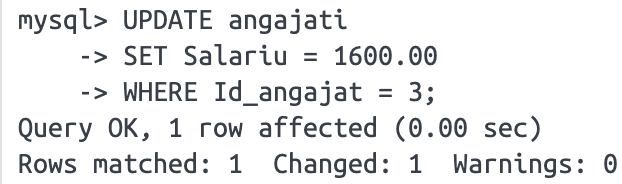
SARCINI CU UPDATE , DELETE , INSERT PENTRU BAZA DE DATE intreprindere

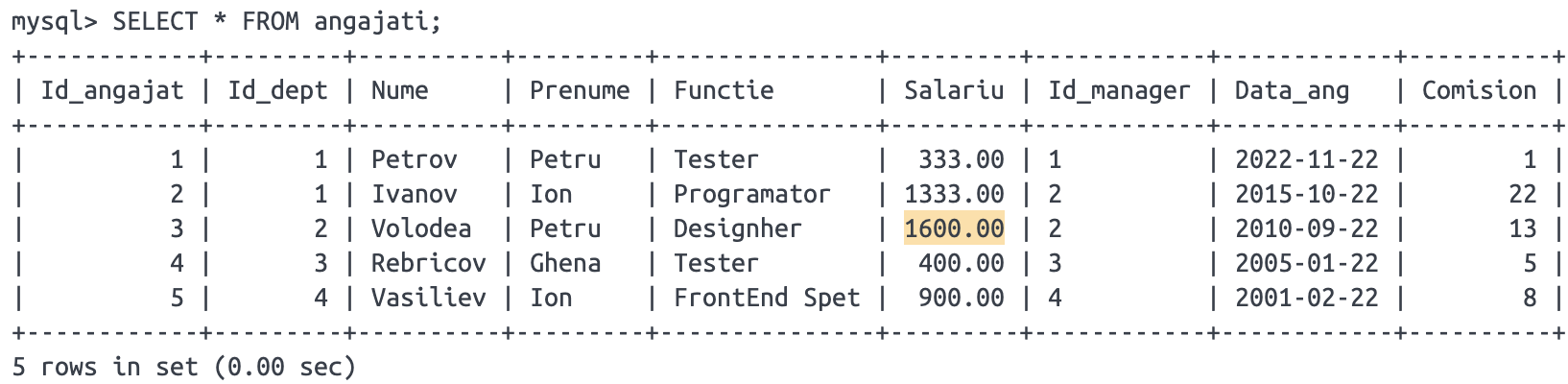
1. **Actualizeaza salariul angajatului cu Id\_angajat = 3 la 1600.00.**

**UPDATE angajati**

**SET Salariu = 1600.00**

**WHERE Id\_angajat = 3;**

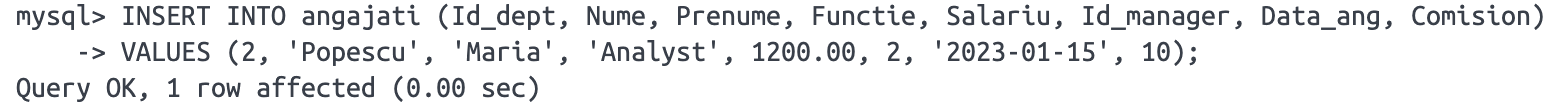
****

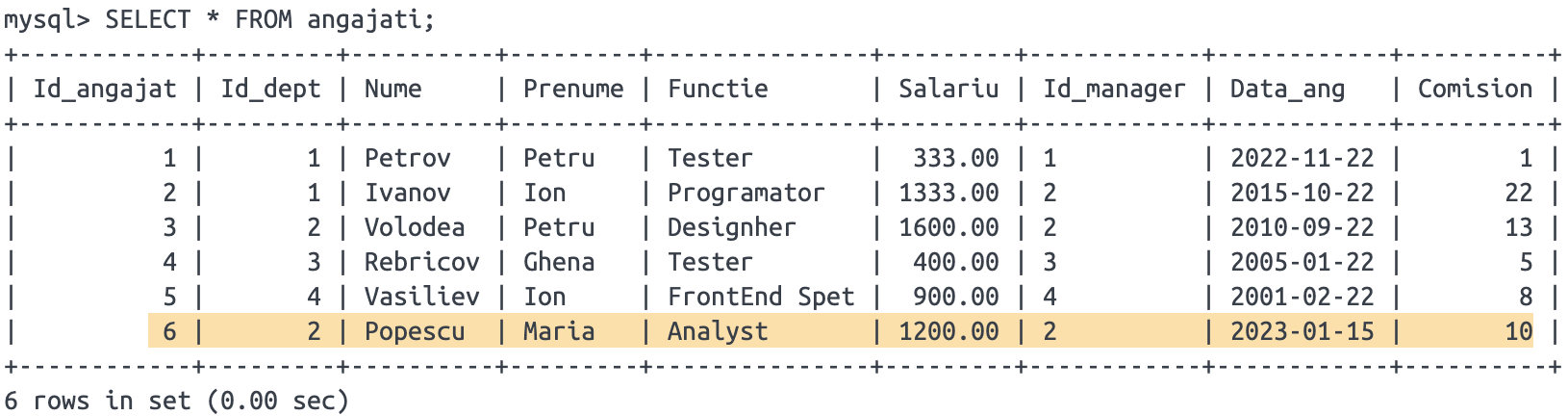
****

1. **Insereaza un angajat nou in departamentul cu Id\_dept = 2.**

**INSERT INTO angajati (Id\_dept, Nume, Prenume, Functie, Salariu, Id\_manager, Data\_ang, Comision)**

**VALUES (2, 'Popescu', 'Maria', 'Analyst', 1200.00, 2, '2023-01-15', 10);**

****

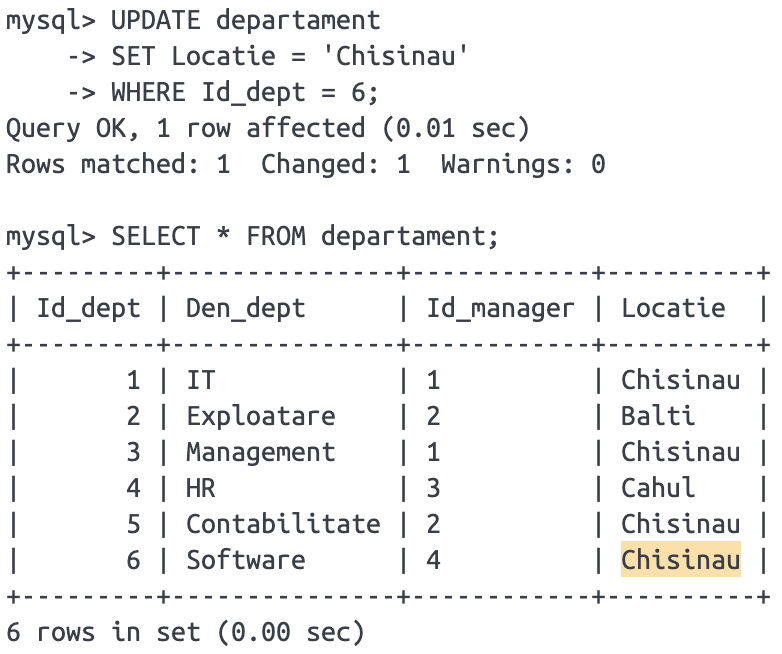
****

1. **Actualizeaza locatia departamentului cu Id\_dept = 6 la "Chisinau".**

**UPDATE departament**

**SET Locatie = 'Chisinau'**

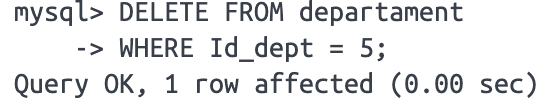
**WHERE Id\_dept = 6;**

****

1. **Sterge departamentul cu Id\_dept = 5.**

**DELETE FROM departament**

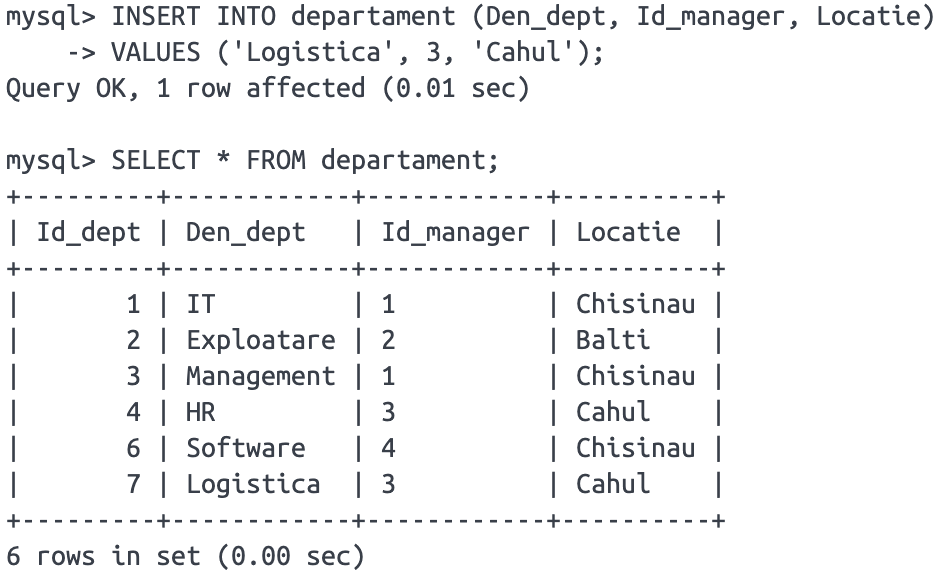
**WHERE Id\_dept = 5;**

****

1. **Insereaza un departament nou cu denumirea "Logistica" si locatia "Cahul".**

**INSERT INTO departament (Den\_dept, Id\_manager, Locatie)**

**VALUES ('Logistica', 3, 'Cahul');**

****

SARCINI CU DECLANSATOARE , PROCEDURI SI FUNCTII STOCATE PENTRU BAZA DE DATE intreprindere

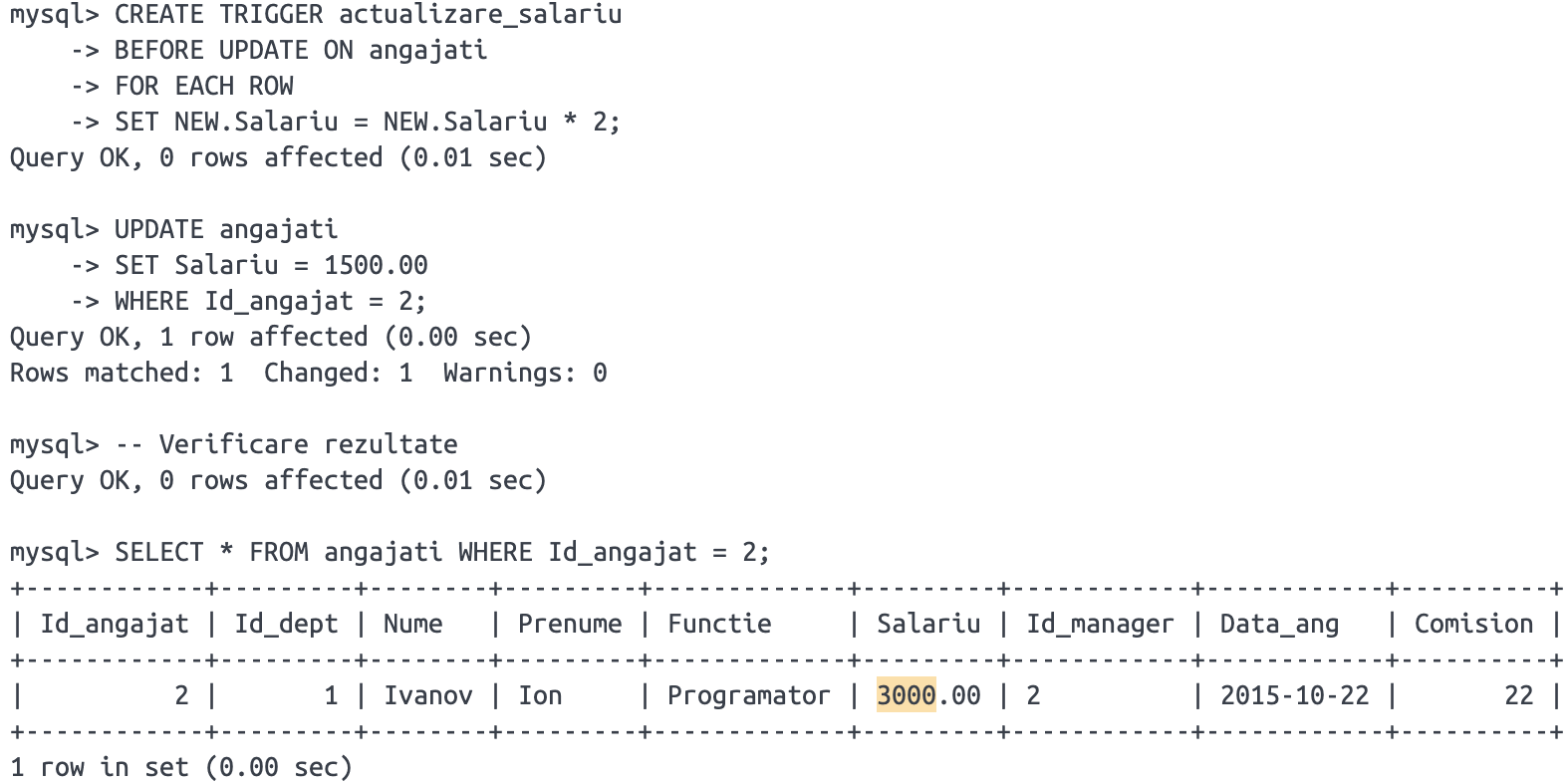
1. **Declanșator pentru actualizare automată a salariului.**

**CREATE TRIGGER actualizare\_salariu**

**BEFORE UPDATE ON angajati**

**FOR EACH ROW**

**SET NEW.Salariu = NEW.Salariu \* 2;**

****

1. **Procedură pentru actualizarea salariului unui angajat**

**DELIMITER //**

**CREATE PROCEDURE actualizare\_salariu\_angajat(IN angajat\_id INT, IN nou\_salariu DOUBLE)**

**BEGIN**

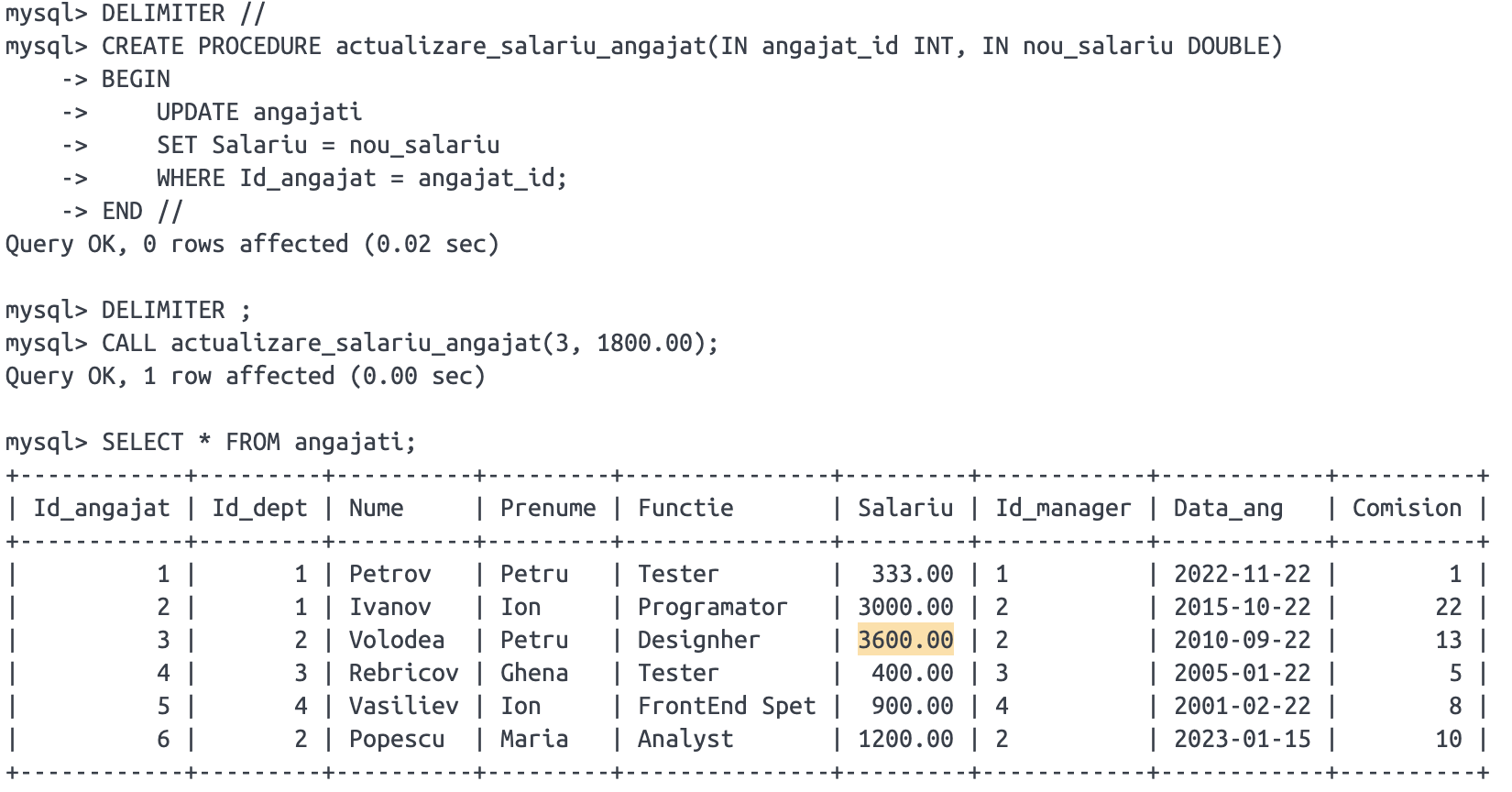
**UPDATE angajati**

**SET Salariu = nou\_salariu**

**WHERE Id\_angajat = angajat\_id;**

**END //**

**DELIMITER ;**

****

**3600 din deoarece s-a apelat TRIGGER actualizare\_salariu care a facut salariu x2.**

1. **Procedură pentru inserarea unui nou departament.**

**DELIMITER //**

**CREATE PROCEDURE inserare\_departament(IN denumire VARCHAR(20), IN manager\_id INT, IN locatie VARCHAR(50))**

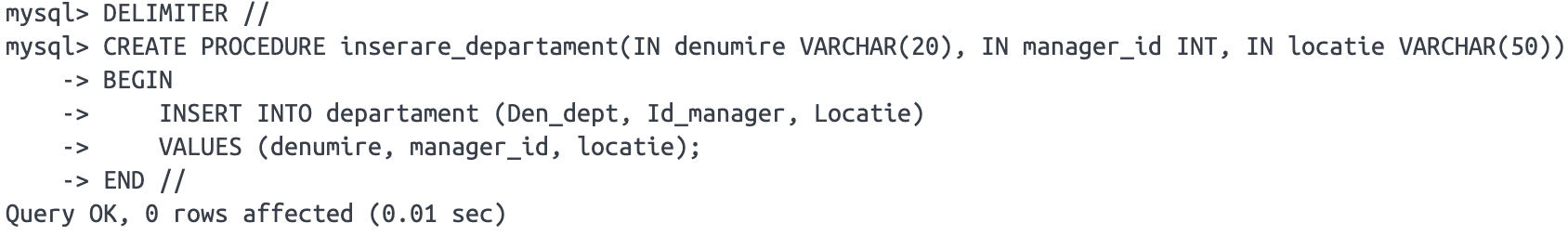
**BEGIN**

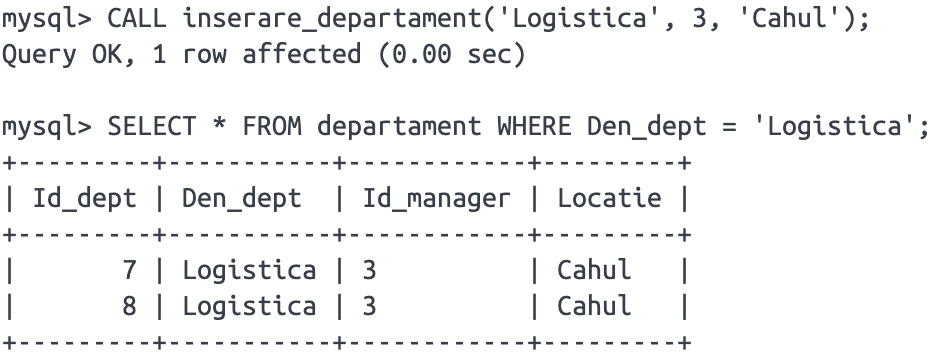
**INSERT INTO departament (Den\_dept, Id\_manager, Locatie)**

**VALUES (denumire, manager\_id, locatie);**

**END //**

**DELIMITER ;**

****

****

1. **Funcție pentru calculul vechimii unui angajat.**

**DELIMITER //**

**CREATE FUNCTION vechime\_angajat(angajat\_id INT)**

**RETURNS INT**

**DETERMINISTIC**

**BEGIN**

**DECLARE data\_angajare DATE;**

**DECLARE vechime INT;**

**SELECT Data\_ang INTO data\_angajare**

**FROM angajati**

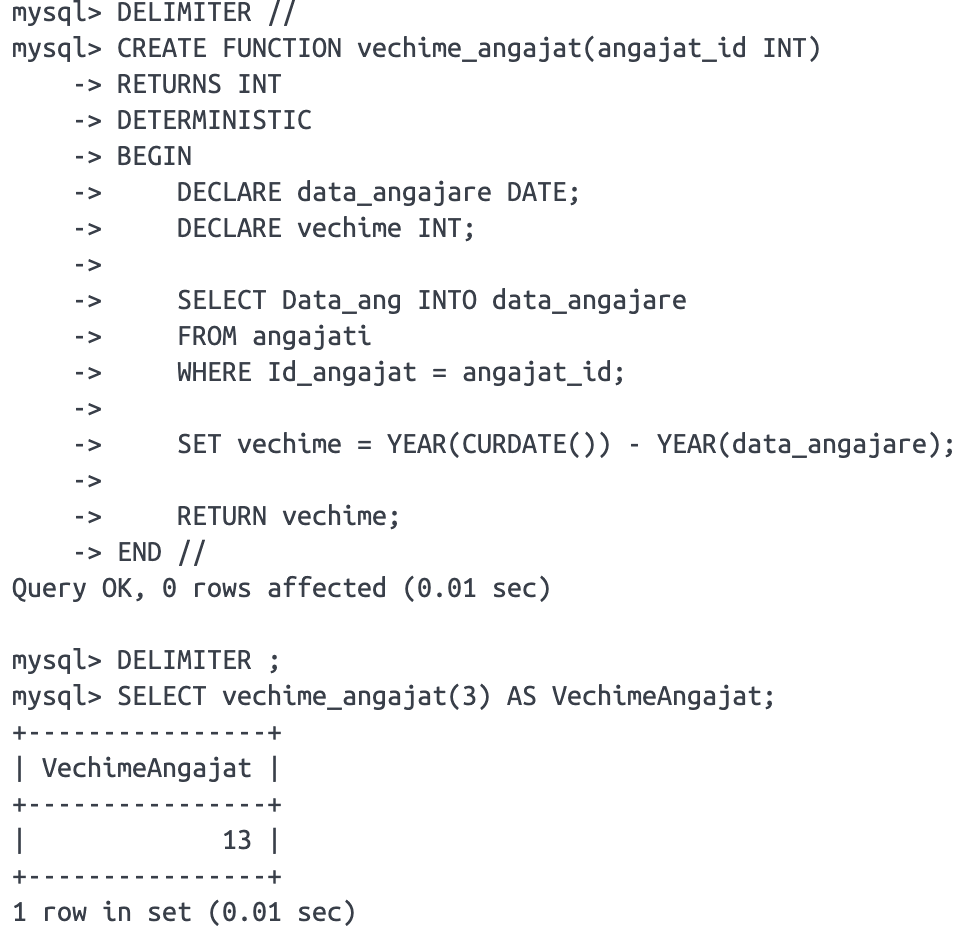
**WHERE Id\_angajat = angajat\_id;**

**SET vechime = YEAR(CURDATE()) - YEAR(data\_angajare);**

**RETURN vechime;**

**END //**

**DELIMITER ;**

****

1. **Declansator pentru a adauga un nou angajat în tabelul "angajati" atunci când un departament nou este adăugat.**

**DELIMITER //**

**CREATE TRIGGER AdaugareAngajatDupaDepartament**

**AFTER INSERT ON departament**

**FOR EACH ROW**

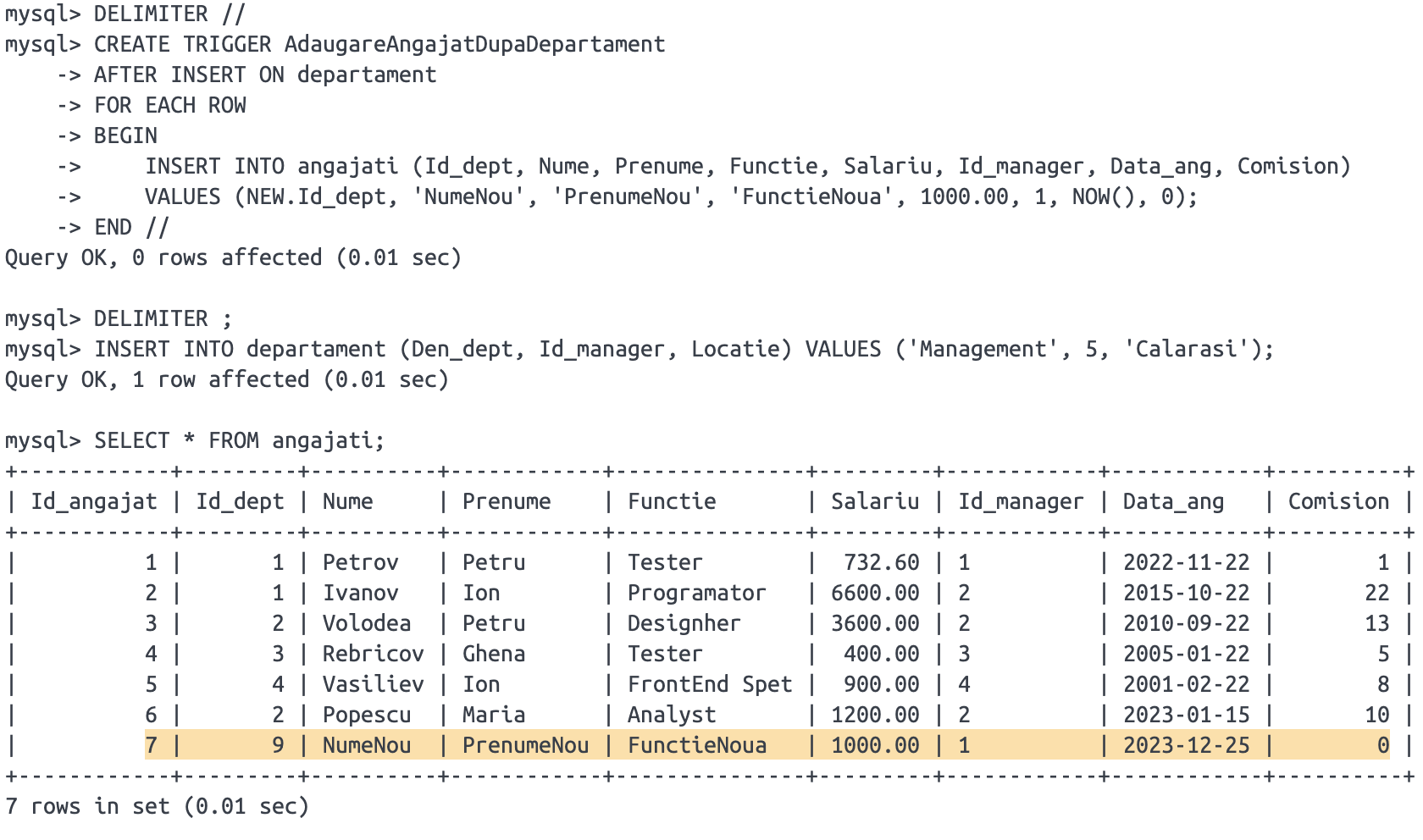
**BEGIN**

**INSERT INTO angajati (Id\_dept, Nume, Prenume, Functie, Salariu, Id\_manager, Data\_ang, Comision)**

**VALUES (NEW.Id\_dept, 'NumeNou', 'PrenumeNou', 'FunctieNoua', 1000.00, 1, NOW(), 0);**

**END //**

**DELIMITER ;**

****

1. **Funcție stocată pentru a calcula salariul total al unui departament.**

**DELIMITER //**

**CREATE FUNCTION CalculSalariuTotalDepartament(dept\_id INT) RETURNS DECIMAL(10,2)**

**DETERMINISTIC**

**BEGIN**

**DECLARE total\_salariu DECIMAL(10,2);**

**SELECT SUM(Salariu) INTO total\_salariu**

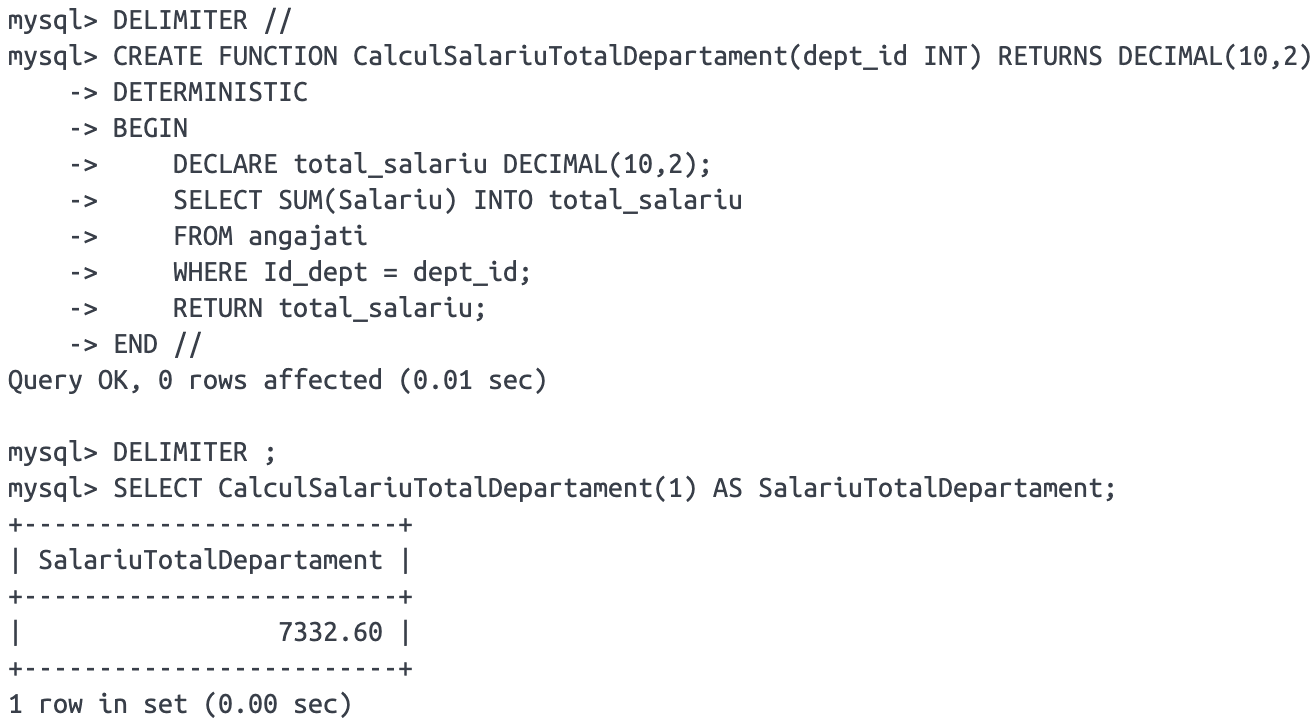
**FROM angajati**

**WHERE Id\_dept = dept\_id;**

**RETURN total\_salariu;**

**END //**

**DELIMITER ;**

****

1. **Procedura pentru a adăuga un nou angajat.**

**DELIMITER //**

**CREATE PROCEDURE AdaugaAngajat(**

**IN Nume\_param varchar(40),**

**IN Prenume\_param varchar(40),**

**IN Functie\_param varchar(25),**

**IN Salariu\_param double,**

**IN Id\_manager\_param varchar(3),**

**IN Data\_ang\_param date,**

**IN Comision\_param int,**

**IN Id\_dept\_param int**

**)**

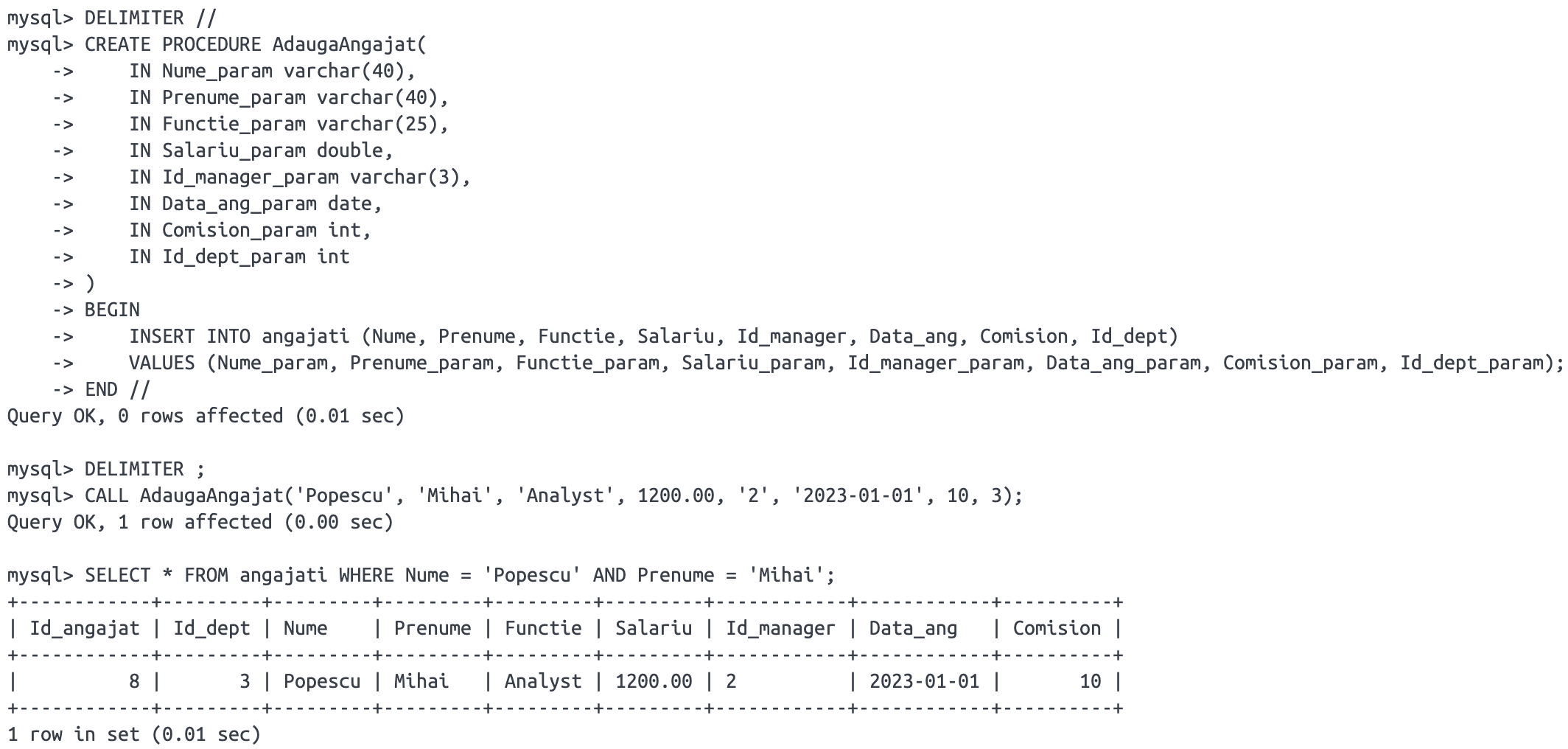
**BEGIN**

**INSERT INTO angajati (Nume, Prenume, Functie, Salariu, Id\_manager, Data\_ang, Comision, Id\_dept)**

**VALUES (Nume\_param, Prenume\_param, Functie\_param, Salariu\_param, Id\_manager\_param, Data\_ang\_param, Comision\_param, Id\_dept\_param);**

**END //**

**DELIMITER ;**

****

1. **Procedura pentru a actualiza informațiile unui angajat.**

**DELIMITER //**

**CREATE PROCEDURE ActualizeazaAngajat(**

**IN Id\_angajat\_param INT,**

**IN Nume\_param varchar(40),**

**IN Prenume\_param varchar(40),**

**IN Functie\_param varchar(25),**

**IN Salariu\_param double,**

**IN Id\_manager\_param varchar(3),**

**IN Data\_ang\_param date,**

**IN Comision\_param int,**

**IN Id\_dept\_param int**

**)**

**BEGIN**

**UPDATE angajati**

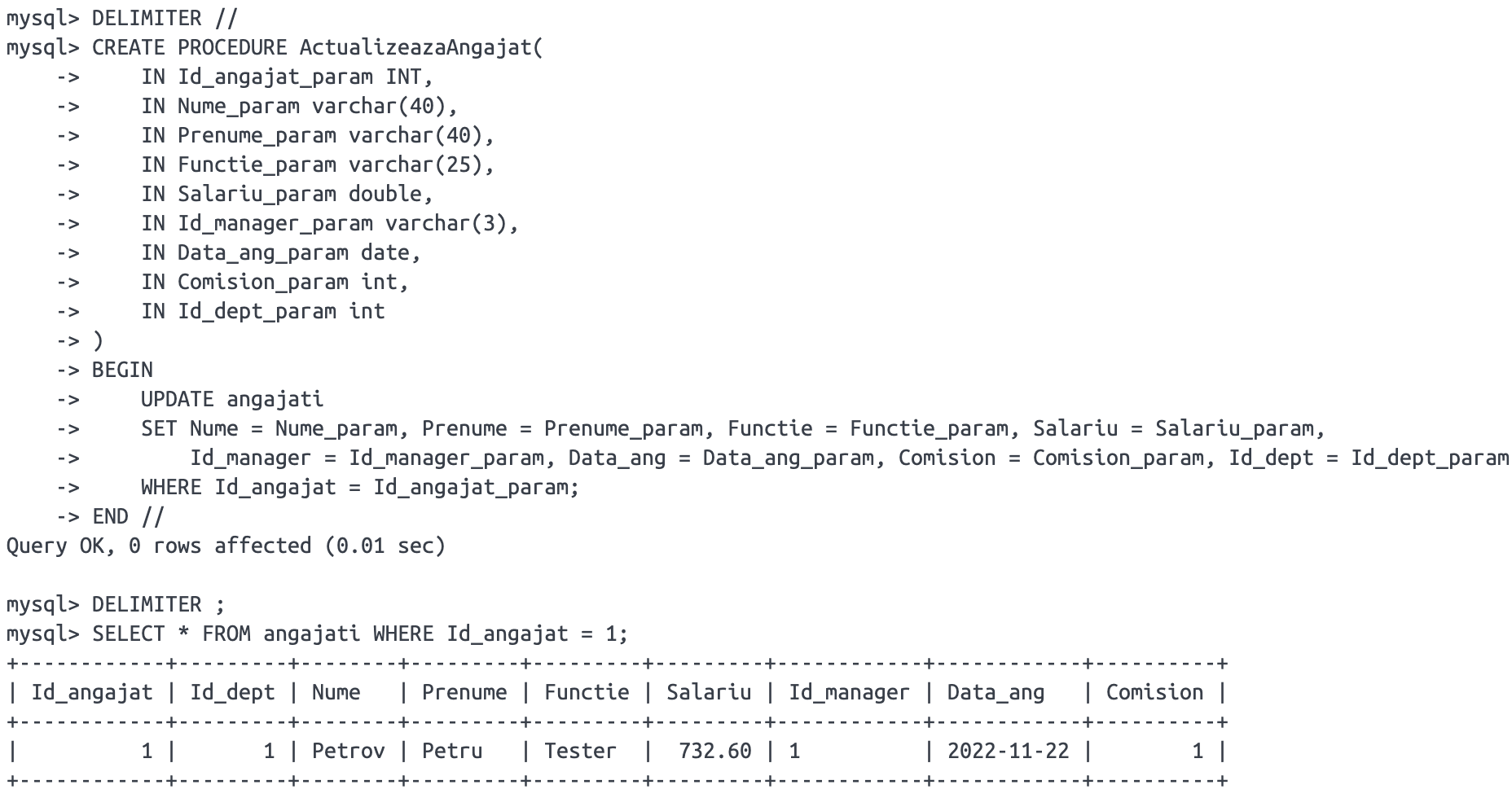
**SET Nume = Nume\_param, Prenume = Prenume\_param, Functie = Functie\_param, Salariu = Salariu\_param,**

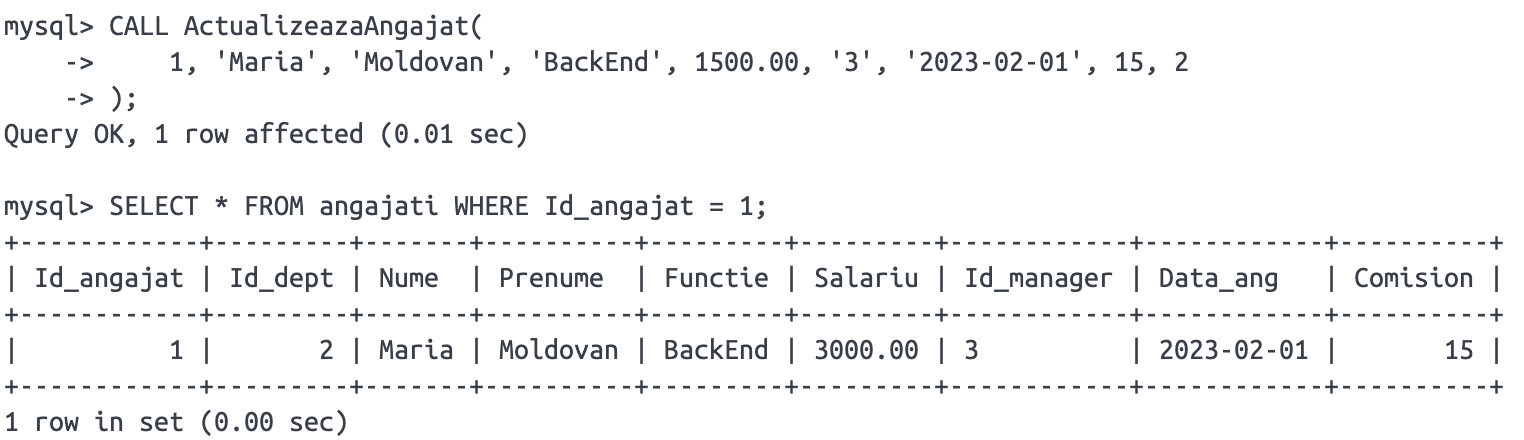
**Id\_manager = Id\_manager\_param, Data\_ang = Data\_ang\_param, Comision = Comision\_param, Id\_dept = Id\_dept\_param**

**WHERE Id\_angajat = Id\_angajat\_param;**

**END //**

**DELIMITER ;**

****

****

1. **Procedura pentru a obține informații despre un angajat specific.**

**DELIMITER //**

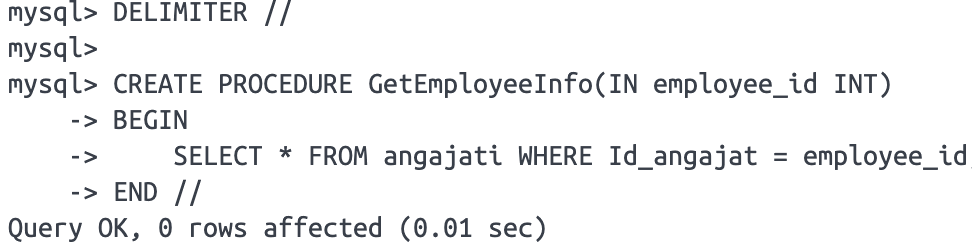
**CREATE PROCEDURE GetEmployeeInfo(IN employee\_id INT)**

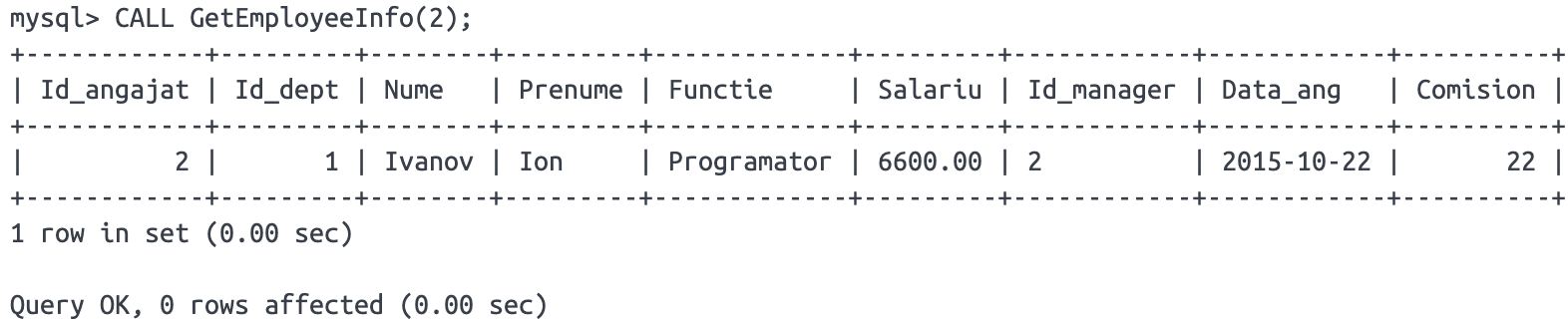
**BEGIN**

**SELECT \* FROM angajati WHERE Id\_angajat = employee\_id;**

**END //**

**DELIMITER ;**

****

****

1. **Funcția pentru a obține numărul total de angajați într-un departament.**

**DELIMITER //**

**CREATE FUNCTION GetEmployeeCountInDepartment(department\_id INT)**

**RETURNS INT**

**READS SQL DATA**

**BEGIN**

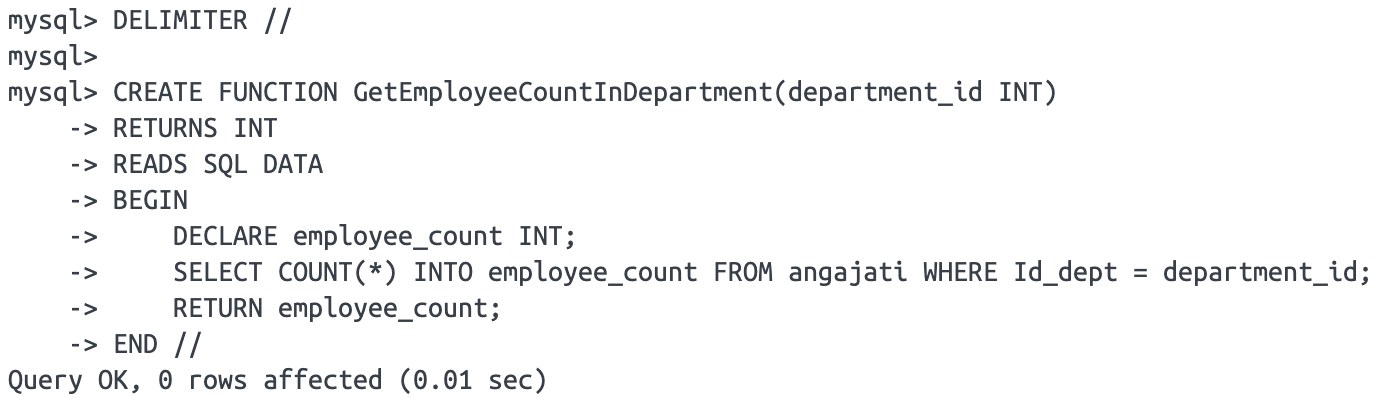
**DECLARE employee\_count INT;**

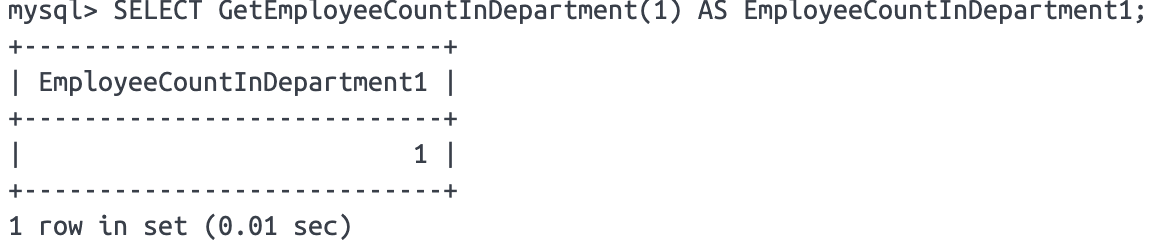
**SELECT COUNT(\*) INTO employee\_count FROM angajati WHERE Id\_dept = department\_id;**

**RETURN employee\_count;**

**END //**

**DELIMITER ;**

****

****