E5.

CREATE OR REPLACE TRIGGER d\_dept\_ang-- Doar delete pt dept de la fiecare ang

BEFORE DELETE ON DEPT\_TEST\_DP-- Trigger la nivel de linie care s-ar declansa inainte

FOR EACH ROW

BEGIN

DELETE FROM EMP\_TEST\_DP WHERE department\_id = :OLD.department\_id;

END;

/

CREATE OR REPLACE TRIGGER u\_dept\_ang

BEFORE UPDATE ON DEPT\_TEST\_DP

FOR EACH ROW

BEGIN

IF :NEW.department\_id != :OLD.department\_id THEN

UPDATE EMP\_TEST\_DP

SET department\_id = :NEW.department\_id

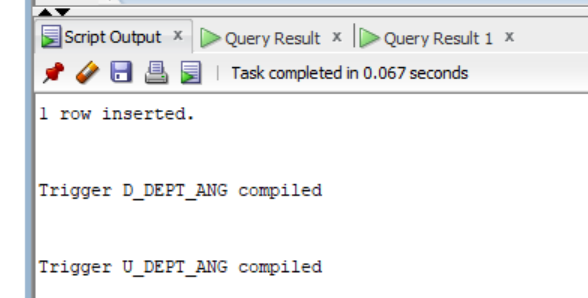
WHERE department\_id = :OLD.department\_id;

END IF;

END;

/

* abia dupa ce fiecare tabel e creat



--Testez fara constrangere de fk in emp referitor la dept

DROP TABLE DEPT\_TEST\_DP CASCADE CONSTRAINTS;

DROP TABLE EMP\_TEST\_DP;

CREATE TABLE DEPT\_TEST\_DP (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50) NOT NULL

);

CREATE TABLE EMP\_TEST\_DP (

employee\_id NUMBER PRIMARY KEY,

last\_name VARCHAR2(50),

first\_name VARCHAR2(50),

department\_id NUMBER-- Pentru testare valori nu il definesc ca fiind foreign key constraint

--Altfel ar fi fost: FOREIGN KEY (department\_id) REFERENCES dept\_test\_dp(department\_id)

);

INSERT INTO DEPT\_TEST\_DP (department\_id, department\_name)

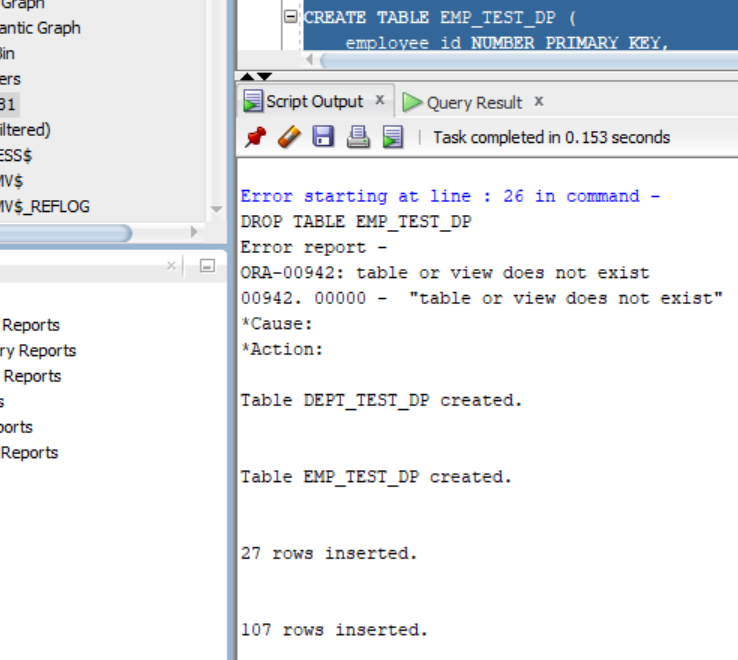
SELECT department\_id, department\_name

FROM departments;

INSERT INTO EMP\_TEST\_DP (employee\_id, last\_name, first\_name, department\_id)

SELECT employee\_id, last\_name, first\_name, department\_id

FROM employees;



--Mai intai identific ce department\_id pot alege dintre ce e inserat in emp

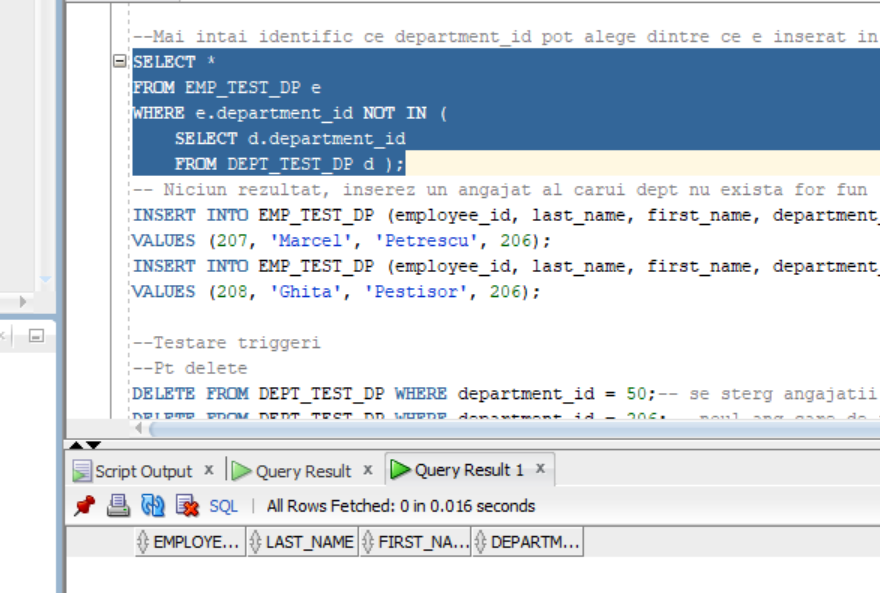
SELECT \*

FROM EMP\_TEST\_DP e

WHERE e.department\_id NOT IN (

SELECT d.department\_id

FROM DEPT\_TEST\_DP d );



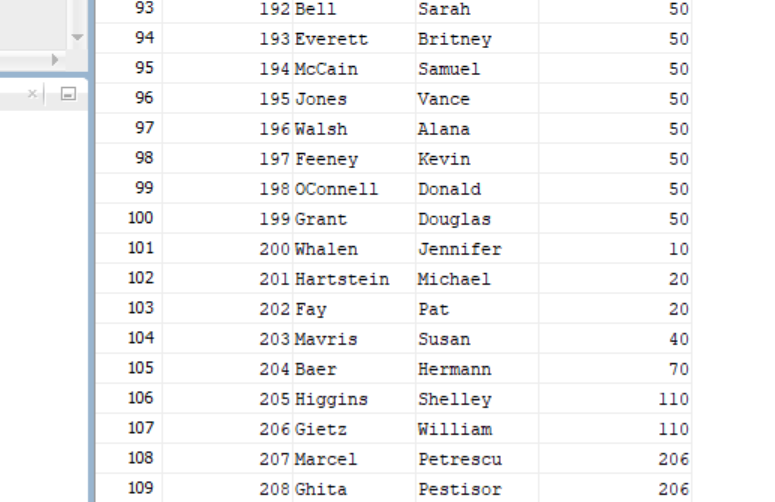
-- Niciun rezultat, inserez un angajat al carui dept nu exista for fun

INSERT INTO EMP\_TEST\_DP (employee\_id, last\_name, first\_name, department\_id)

VALUES (207, 'Marcel', 'Petrescu', 206);

INSERT INTO EMP\_TEST\_DP (employee\_id, last\_name, first\_name, department\_id)

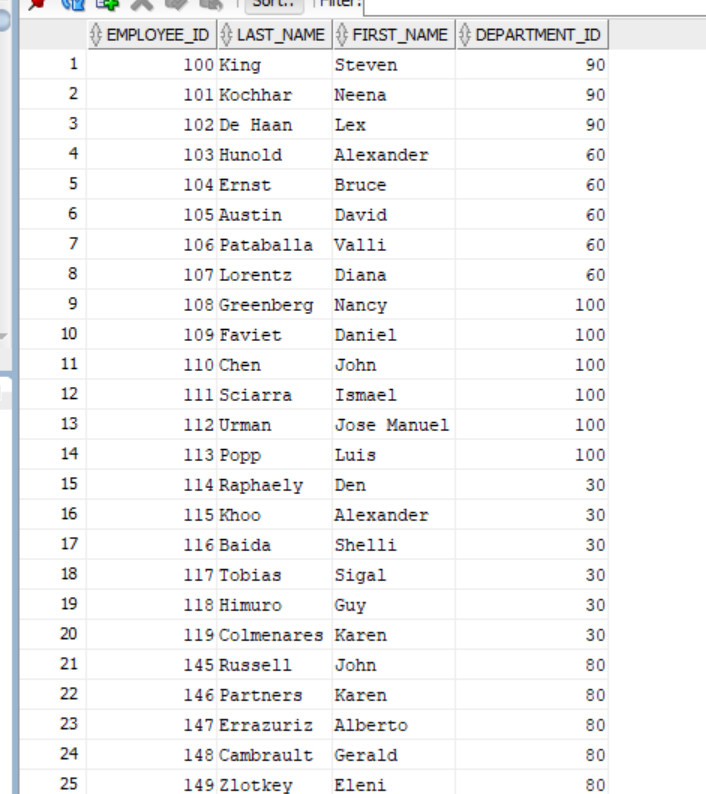
VALUES (208, 'Ghita', 'Pestisor', 206);



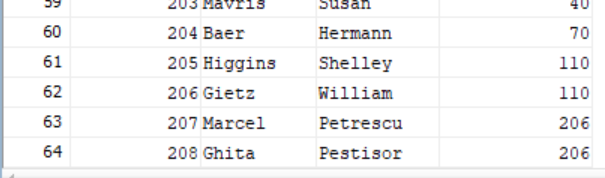
--Testare triggeri

--Pt delete

DELETE FROM DEPT\_TEST\_DP WHERE department\_id = 50;-- se sterg angajatii fara nicio problema cu toat ca deptarment\_id nu e definit ca fk



DELETE FROM DEPT\_TEST\_DP WHERE department\_id = 206;-- noul ang care de fapt are un dept inexistent, nu dispare

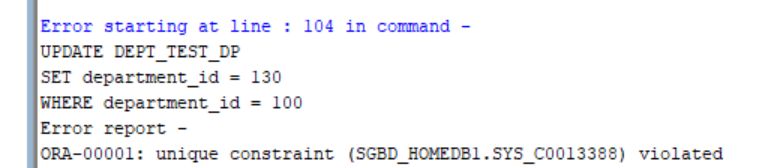


--Pt update

UPDATE DEPT\_TEST\_DP

SET department\_id = 130

WHERE department\_id = 100;-- 130 exista ca si dept, unique constraint violated



--Testez cu constrangere de fk in emp referitor la dept

DROP TABLE DEPT\_TEST\_DP CASCADE CONSTRAINTS;

DROP TABLE EMP\_TEST\_DP;

CREATE TABLE DEPT\_TEST\_DP (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50) NOT NULL

);

CREATE TABLE EMP\_TEST\_DP (

employee\_id NUMBER PRIMARY KEY,

last\_name VARCHAR2(50),

first\_name VARCHAR2(50),

department\_id NUMBER,

CONSTRAINT fk\_department FOREIGN KEY (department\_id) REFERENCES DEPT\_TEST\_DP(department\_id)

);

INSERT INTO DEPT\_TEST\_DP (department\_id, department\_name)

SELECT department\_id, department\_name

FROM departments;

INSERT INTO EMP\_TEST\_DP (employee\_id, last\_name, first\_name, department\_id)

SELECT employee\_id, last\_name, first\_name, department\_id

FROM employees;

--Chestia asta nu merge, cum ma asteptam

INSERT INTO EMP\_TEST\_DP (employee\_id, last\_name, first\_name, department\_id)

VALUES (207, 'Marcel', 'Petrescu', 206);-- parent key not found

--Testare triggeri

--Pt delete

DELETE FROM DEPT\_TEST\_DP WHERE department\_id = 50;-- se sterg angajatii

* aceleasi screenshot-uri

--Pt update

UPDATE DEPT\_TEST\_DP

SET department\_id = 130

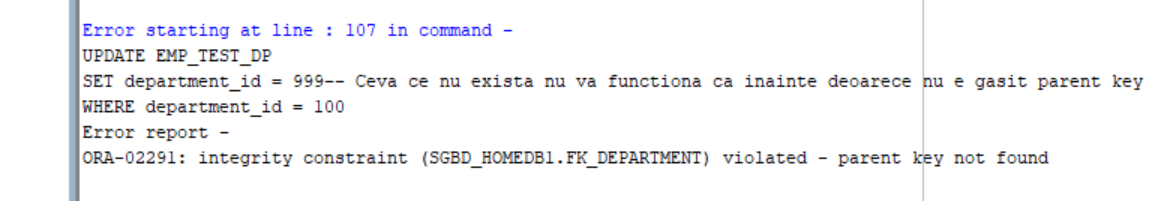
WHERE department\_id = 100;-- 130 exista ca si dept, unique constraint violated

* aceleasi screenshot-uri

UPDATE EMP\_TEST\_DP

SET department\_id = 999-- Ceva ce nu exista nu va functiona ca inainte deoarece nu e gasit parent key

WHERE department\_id = 100;



--Testez cu constrangere de fk in emp referitor la dept on delete cascade

DROP TABLE DEPT\_TEST\_DP CASCADE CONSTRAINTS;

DROP TABLE EMP\_TEST\_DP;

CREATE TABLE DEPT\_TEST\_DP (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50) NOT NULL

);

CREATE TABLE EMP\_TEST\_DP (

employee\_id NUMBER PRIMARY KEY,

last\_name VARCHAR2(50),

first\_name VARCHAR2(50),

department\_id NUMBER,

CONSTRAINT fk\_department FOREIGN KEY (department\_id)

REFERENCES DEPT\_TEST\_DP(department\_id)

ON DELETE CASCADE

);

INSERT INTO DEPT\_TEST\_DP (department\_id, department\_name)

SELECT department\_id, department\_name

FROM departments;

INSERT INTO EMP\_TEST\_DP (employee\_id, last\_name, first\_name, department\_id)

SELECT employee\_id, last\_name, first\_name, department\_id

FROM employees;

--Testare triggeri

--Pt delete

DELETE FROM DEPT\_TEST\_DP WHERE department\_id = 50;-- se sterg angajatii

* aceleasi screenshot-uri

--Pt update

UPDATE DEPT\_TEST\_DP

SET department\_id = 130

WHERE department\_id = 100;-- 130 exista ca si dept, unique constraint violated

UPDATE DEPT\_TEST\_DP

SET department\_id = 999

WHERE department\_id = 100;

* aceleasi screenshot-uri

--Testez cu constrangere de fk in emp referitor la dept on delete set null

DROP TABLE DEPT\_TEST\_DP CASCADE CONSTRAINTS;

DROP TABLE EMP\_TEST\_DP;

CREATE TABLE DEPT\_TEST\_DP (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50) NOT NULL

);

CREATE TABLE EMP\_TEST\_DP (

employee\_id NUMBER PRIMARY KEY,

last\_name VARCHAR2(50),

first\_name VARCHAR2(50),

department\_id NUMBER,

CONSTRAINT fk\_department FOREIGN KEY (department\_id)

REFERENCES DEPT\_TEST\_DP(department\_id)

ON DELETE SET NULL

);

INSERT INTO DEPT\_TEST\_DP (department\_id, department\_name)

SELECT department\_id, department\_name

FROM departments;

INSERT INTO EMP\_TEST\_DP (employee\_id, last\_name, first\_name, department\_id)

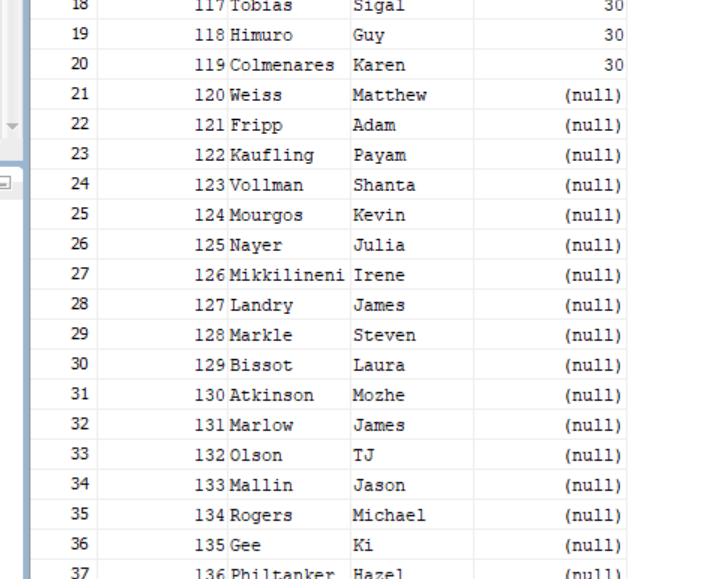
SELECT employee\_id, last\_name, first\_name, department\_id

FROM employees;

--Testare triggeri

--Pt delete

DELETE FROM DEPT\_TEST\_DP WHERE department\_id = 50;-- nu se sterg angajatii, se seteaza department\_id la null



--Pt update

UPDATE DEPT\_TEST\_DP

SET department\_id = 130

WHERE department\_id = 100;-- 130 exista ca si dept, unique constraint violated

UPDATE DEPT\_TEST\_DP

SET department\_id = 999

WHERE department\_id = 100;

--E6 pt baza mea de date

CREATE TABLE EroriMagazinInghetata (

user\_id VARCHAR2(128),

nume\_bd VARCHAR2(128),

erori VARCHAR2(4000),

data TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE OR REPLACE TRIGGER Erori

AFTER SERVERERROR ON DATABASE

BEGIN

INSERT INTO EroriMagazinInghetata (user\_id, nume\_bd, erori)

VALUES (SYS.LOGIN\_USER, SYS.DATABASE\_NAME, DBMS\_UTILITY.FORMAT\_ERROR\_STACK);

END;

/

-- il creez in sys\_homedb1

-- apoi ca sa generez niste erori interesante fac cu contstraint uri interesante in sys\_homedb1 si sgbd\_homedb1

GRANT CREATE TRIGGER TO SGBD\_HOMEDB1;

-- exista deja recenziile 1 si 2

INSERT INTO Recenzii (IDRecenzie , IDClient , Rating , Comentariu , DataRecenzie)

VALUES (1 , 1 , 5 , 'O sa mai trec vara asta' , TO\_DATE('2024-01-03' , 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie , IDClient , Rating , Comentariu , DataRecenzie)

VALUES (2 , 2 , 4 , 'Produse peste asteptari wow' , TO\_DATE('2024-01-04' , 'YYYY-MM-DD'));

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (2, NULL, 4);-- nu pot asocia o comanda cu o aroma care e null

ALTER TABLE AdaugaAroma MODIFY IDAroma null;-- nu merge pt ca AdaugaAroma are o cheie primara compusa care trebuie sa fie unique si not null prin defeinitie

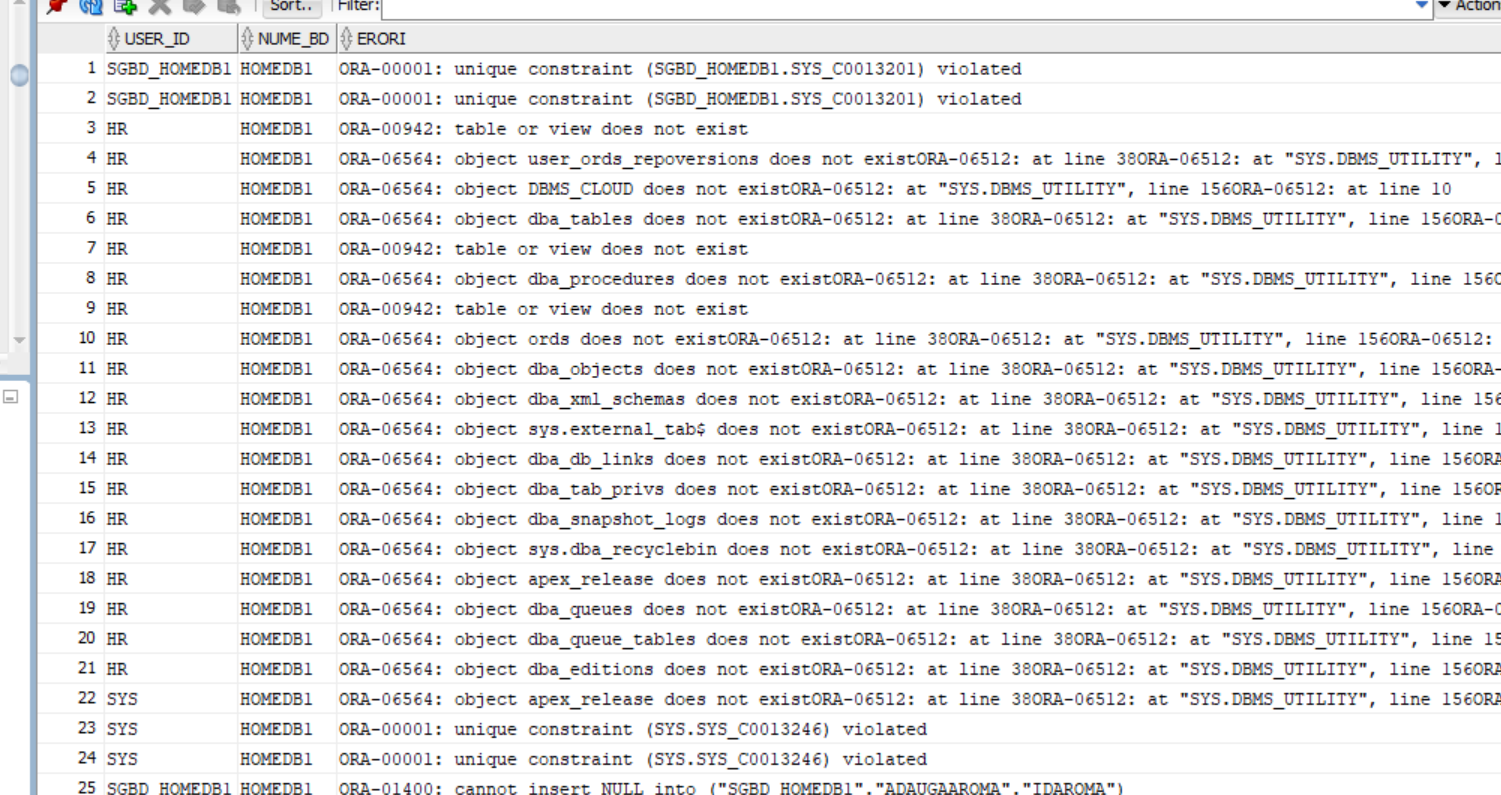
CREATE SEQUENCE SecventaComanda

START WITH 100

INCREMENT BY 5

NOCACHE

NOCYCLE;-- exista deja, eroare name is already used



--E7

DROP TABLE InfoComenzi;

CREATE TABLE InfoComenzi (

IDAdresa NUMBER PRIMARY KEY,

Suma NUMBER,

FOREIGN KEY (IDAdresa) REFERENCES Adresa(IDAdresa)

);

CREATE OR REPLACE PROCEDURE modific\_suma\_adresa

(

v\_id\_adresa InfoComenzi.IDAdresa%TYPE,

v\_suma InfoComenzi.Suma%TYPE

) AS

BEGIN

UPDATE InfoComenzi

SET Suma = NVL(Suma, 0) + v\_suma

WHERE IDAdresa = v\_id\_adresa;

END;

/

CREATE OR REPLACE TRIGGER trig\_update\_suma\_adrese

AFTER INSERT OR DELETE OR UPDATE OF SumaInvestitie ON Sponsor

FOR EACH ROW

BEGIN

IF UPDATING THEN

modific\_suma\_adresa(:OLD.IDAdresa, :NEW.SumaInvestitie-:OLD.SumaInvestitie);

ELSIF DELETING THEN

modific\_suma\_adresa(:OLD.IDAdresa, -1\*:OLD.SumaInvestitie);

ELSIF INSERTING THEN

modific\_suma\_adresa(:OLD.IDAdresa, :NEW.SumaInvestitie);

END IF;

END;

/

DELETE FROM InfoComenzi;

INSERT INTO InfoComenzi (IDAdresa, Suma) VALUES (12, 1000);

--INSERT INTO Sponsor VALUES (21, 12, 'MaxFactoryy', 'max@gmail.com', '0880187170', 200.4, 2);-- nu stiu de ce nu merge

UPDATE Sponsor

SET SumaInvestitie = SumaInvestitie + 1000

WHERE IDSponsor = 20;

DELETE FROM Sponsor WHERE IDSponsor = 20;

s-a dat u[pdate pana la urma