

BAZA DE DATE A UNEI COMUNE

Florete Fabian-Andrei
grupa 134

Facultatea de matematică și informatică,
Universitatea din București

Cerința 1:

Modelul prezentat își propune să modeleze o bază de date corespunzătoare unei comune. Ea are în evidență diferitele familii și cetățeni ce locuiesc în acesta. Primăria satului poate organiza diferite evenimente culturale, precum și oficializa contractele de negoț dintre familii și cumpărători. Pe parcursul timpului, se rețin date precum formele de învățământ absolvite de persoane, precum și casele în care au stat.

Modelul real are ca scop crearea unei baze de date capabilă să gestioneze evenimentele ce se pot petrece într-o comună. El are utilitate în lumea reală, deoarece poate unor entități organizatorice, precum primăriile să-și gestioneze mai ușor cetățenii, urmărind cu ușurință informații precum sumele pe care aceștia le încasează, dar și casele la care locuiesc aceștia.

Cerința 2:

Baza de date proiectată funcționează după următoarele reguli:

- Nu pot exista mai mult de 50 de case pe o singură stradă
- Nu pot exista două case la același număr
- Fiecare locuitor face parte dintr-o familie
- Fiecare casă are asociată o stradă și un număr
- Pe o stradă se pot afla mai multe instituții de învățământ, dar nu și mai multe primării
- Orice informație referitoare la o dată nu poate fi nulă (an de învățământ, data contractului de cumpărare, data de mutare dintr-o casă în alta a unei familii)

Cerința 3:

Entități:

În modelarea acestei baze de date am folosit următoarele entități: PRIMARIE, FIRMA_ORGANIZARE, EVENIMENT_CULTURAL, FORMA_INVATAMANT, LOCUITOR, CUMPARATOR, FAMILIE, CASA, STRADA. Mai jos voi da o descriere detaliată fiecăreia, precizând și cheia primară a acestora:

PRIMARIE: cheie primară: ID_PRIMĂRIE. Ea are rolul de a ține evidența locuitorilor în sat. Oficializează contractele de cumpărare și colaborează cu firmele de organizare pentru evenimente culturale.

FIRMA_ORGANIZARE: cheie primară: ID_FIRMA. Se ocupă de organizarea evenimentelor culturale.

EVENIMENT_CULTURAL: cheie primară: ID_EVENIMENT. Reprezintă o listă cu evenimentele care pot fi organizate de către firme.

CUMPARATOR: cheie primară: ID_CUMPARATOR. Aceasta va fi entitatea cu

care va participa în contractele de vânzare-cumpărare cu familiile din sat. Ele vor fi coordoante de primărie.

FORMA_DE_INVATAMANT: cheie primară: ID_FORMA. Conține toate formele de învățământ la care vor fi înscriși locuitorii (Gimnaziu, Școala Generală, Liceu).

LOCUTOR: cheie primară: ID_LOCUTOR. Entitatea ce ține în evidență oamenii prezenți în sat. Ei fac parte dintr-o familie și urmează/ au urmat o formă de învățământ.

FAMILIE: cheie primară: ID_FAMILIE. Este entitatea în care vor fi incluși mai mulți locuitori. Ea va locui pe parcursul timpului la mai multe case.

CASA: cheie primară compusă din ID_STRADA și NUMAR prin faptul că pe o stradă se poate afla doar o singură casă la acel număr. În casă se vor afla diferite familii pe intervale diferite de timp.

STRADA: cheie primară ID_STRADA. De ea vor depinde cheile primare ale caselor. Totodată, fiecare stradă are un nume specific.

Cerința 4:

Relații:

Mai departe, voi prezenta relațiile dintre entități. Oferind, în continuare, o descriere detaliată a acestora, precum și cardinalitatea.

PRIMARIE_organizeaza_EVENTIMENT_CULTURAL = relație ce leagă entitățile PRIMARIE și EVENTIMENT_CULTURAL, indicând evenimentele organizate de fiecare primărie pe parcursul timpului. Ea are cardinalul $M(0) : M(0)$ (O primărie poate organiza pe parcursul timpului mai multe evenimente culturale, iar fiecare eveniment poate fi organizat de mai multe ori). Cardinalitatea minimă este $0 : 0$, iar cea maximă $n : n$.

FIRMA_ORGANIZARE_se_ocupa_de_EVENTIMENT_CULTURAL = relație ce leagă entitățile FIRMA_ORGANIZARE și EVENTIMENT_CULTURAL, indicând de ce firmă este organizat fiecare eveniment cultural. Ea are cardinalul $1 : M(0)$ (Un eveniment cultural este organizat de o singură firmă, iar o firmă organizează mai multe evenimente. Ea poate totuși să nu organizeze încă evenimente). Cardinalitatea minimă este $1 : 0$, iar cea maximă $1 : n$.

PRIMARIE_are_in_evidenta_FAMILIE = relație ce leagă entitățile PRIMARIE și FAMILIE, indicând primăria care are în evidență fiecare familie. Ea are cardinalul $1 : M(1)$ (O primărie are în evidență mai multe familii, dar trebuie să aibe cel puțin una, iar o familie trebuie să se afle în evidența unei singure primării). Cardinalitatea minimă este $1 : 1$, iar cea maxima $1 : n$.

LOCUTOR_apartine_FAMILIE = relație ce leagă entitățile LOCUTOR și FAMILIE, indicând familia din care face parte fiecare locuitor. Ea are cardinalul $1 : M(1)$ (Un locuitor trebuie să aparțină unei familii, iar o familie este formată din mai

mulți locuitori, dar cel puțin unul). Cardinalitatea minimă este 1 : 1, iar cea maximă 1:n.

LOCUIITOR_urmeaza_FORMA_INVATAMANT = relație ce leagă entitățile FORMA_INVATAMANT și LOCUIITOR, indicând foremele de învățământ urmate de locuitori pe parcursul anilor. Ea are cardinalul $M(0) : M(0)$ (O formă de învățământ poate să fie urmată de mai mulți locuitori, sau de niciun locuitor, iar un locuitor poate urma pe parcursul anilor mai multe forme de învățământ, sau niciuna) Cardinalitatea minimă este 1 : 1, iar cea maximă $n : n$.

FAMILIE_locuieste_la_CASA = relație ce leagă entitățile FAMILIE și CASA, indicând casa la care locuiește o anumită familie, dar și istoricul caselor la care a locuit o familie. Ea are cardinalul $M(0) : M(1)$ (La o casă pot locui, pe parcursul timpului mai multe familii, sau niciuna, iar o familie trebuie să fi locuit la cel puțin o casă, dar poate locui pe parcursul timpului la mai multe). Astfel, cardinalitatea minimă este 0:1, iar cea maximă $n:n$.

CASA_se_afla_pe_STRADA = relație ce leagă entitățile CASA și STRADA, indicând strada pe care se află o casa. Ea are cardinalul 1 : $M(0)$ (Pe o strada se pot afla mai multe, sau nicio casă, iar casa se află pe o singură stradă și trebuie să se afle pe o singură stradă). Cardinalitatea minimă este 1 : 0.

PRIMARIE_situata_STRADA = relație ce leagă entitățile PRIMARIE și STRADA, indicând strada pe care se află primăria. Ea are cardinalul 1 : 1 (Pe o stradă se află o singură primărie, sau niciuna, iar primăria trebuie să se afle pe o singură stradă).

PRIMARIE_FAMILIE_incheie_contract_cu_CUMPARATOR = relație de tip 3 ce leagă entitățile PRIMARIE, FAMILIE și CUMPARATOR, indicând cei trei participanți implicați în încheierea unui contract dintre familiile de negustori și cumpărători.

Cerința 5:

Atribute:

Atribute pentru entitatea *PRIMARIE*:

id_primarie (variabila de tip întreg de maxim 3 cifre. Reprezinta cheia primară, deci nu poate fi nulă).

nume (variabilă de tip caracter, de maxim 100 de caractere. Reprezintă numele primăriei. Nu poate fi nulă).

id_strada (variabilă de tip întreg, de maxim 3 cifre. Reprezintă strada pe care se află primăria, va fi o cheie externă la entitatea STRADA. Nu poate fi nulă)

Atribute pentru entitatea *FIRMA_ORGANIZARE*:

id_firma (variabilă de tip întreg de 3 cifre. Reprezinta cheia primară, deci nu poate fi nulă).

nume (variabilă de tip caracter, de lungime maximă 30. Reprezintă denumirea firmei)

Atribute pentru entitatea *EVENIMENT_CULTURAL*:

id_eveniment (variabilă de tip întreg de 3 cifre. Reprezintă cheia primară, deci nu poate fi nulă)

id_firma (Variabilă ce face referire la id-ul firmei care organizează respectivul eveniment cultural)

nume_eveniment (variabilă de tip caracter, de lungime maximă 60. Reprezintă numele evenimentului)

capacitate (Variabilă de tip întreg, de 7 cifre. Numărul de persoane ce pot participa la eveniment.)

Atribute pentru entitatea *INSTITUTIE_INVATAMANT*:

id_forma (variabilă de tip întreg, de 2 cifre. Reprezintă cheia primară, deci nu poate fi nulă).

nume (variabilă de tip caracter, de lungime maximă 50. Reprezintă numele institutiei de învățământ).

tip_forma_invatamant (variabilă de tip caracter, de lungime maximă 10. Nu poate fi nulă și trebuie să fie una dintre următoarele: „Școala generală”, „Școala gimnazială”, „Liceu”)

id_strada (Numele străzii pe care este situată forma de învățământ, face referire la cheia primară a entității STRADA)

Atribute pentru entitatea *LOCUIITOR*:

id_locuitor (variabilă de tip întreg, de 5 cifre. Reprezintă cheia primară, deci nu poate fi nulă)

id_familie (variabilă ce face referire la familie din care face parte locuitorul.)

prenume (variabilă de tip șir de caractere, de lungime maximă 40. Reprezintă prenumele persoanei)

sex (variabilă de tip întreg, de o cifră: 1 dacă persoana este bărbat, și 2 dacă persoana este femeie).

Atribute pentru entitatea *FAMILIE*:

id_familie (variabilă de tip întreg, de 4 cifre. Reprezintă cheia primară a entității familie, deci nu poate fi nulă).

id_primarie (variabilă ce face referire la entitatea PRIMARIE. Ea va indica primăria ce are în evidență respectiva familie).

nume_familie (variabilă de tip șir de caractere, de lungime maximă 25. Ea va fi utilizată pentru a găsi numele de familie al persoanelor).

Atribute pentru entitatea *CASA*:

id_strada (variabilă ce face referire la entitatea STRADA. Ea, împreună cu numărul vor indica adresa la care se află o casă).

numar (variabilă de tip întreg, de maxim 3 cifre. Standard, ea va fi 1 și va intra în componența cheii primare)

camere (variabilă de tip întreg, de maxim 2 cifre. Va indica numărul de camere ale unei case)

dimensiune (variabilă de tip întreg, de maxim 4 cifre. Va indica dimensiunea în metrii pătrați a unei case).

Atribute pentru entitatea *STRADA*:

id_strada (variabilă de tip întreg, de maxim 3 cifre. Va reprezenta cheia primară a acestei entități, deci nu poate fi nulă)

nume_strada (variabilă de tip întreg de șir de caractere , de lungime maximă 100. Ea reprezintă numele străzii).

Atribute pentru entitatea *CUMPARATOR*:

id_cumparator (variabilă de tip întreg, de maxim 3 cifre. Va reprezenta cheia primară a acestei entități, deci nu poate fi nulă).

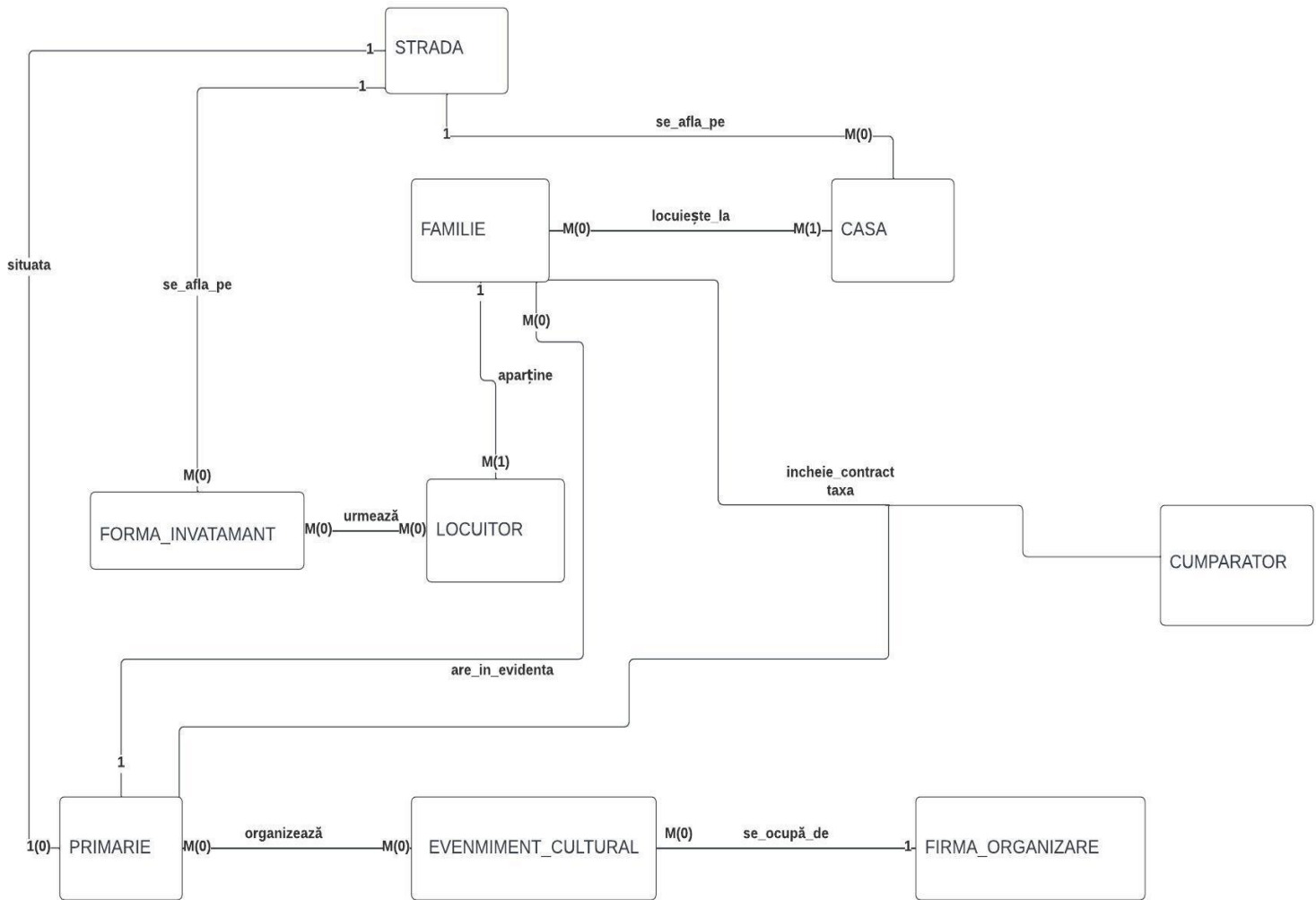
nume (variabilă de tip șir de caractere, de lungime maximă. Va reprezenta numele cumpărătorului. Nu poate fi nulă).

Atribute pentru relația *INCHEIERE_CONTRACT*:

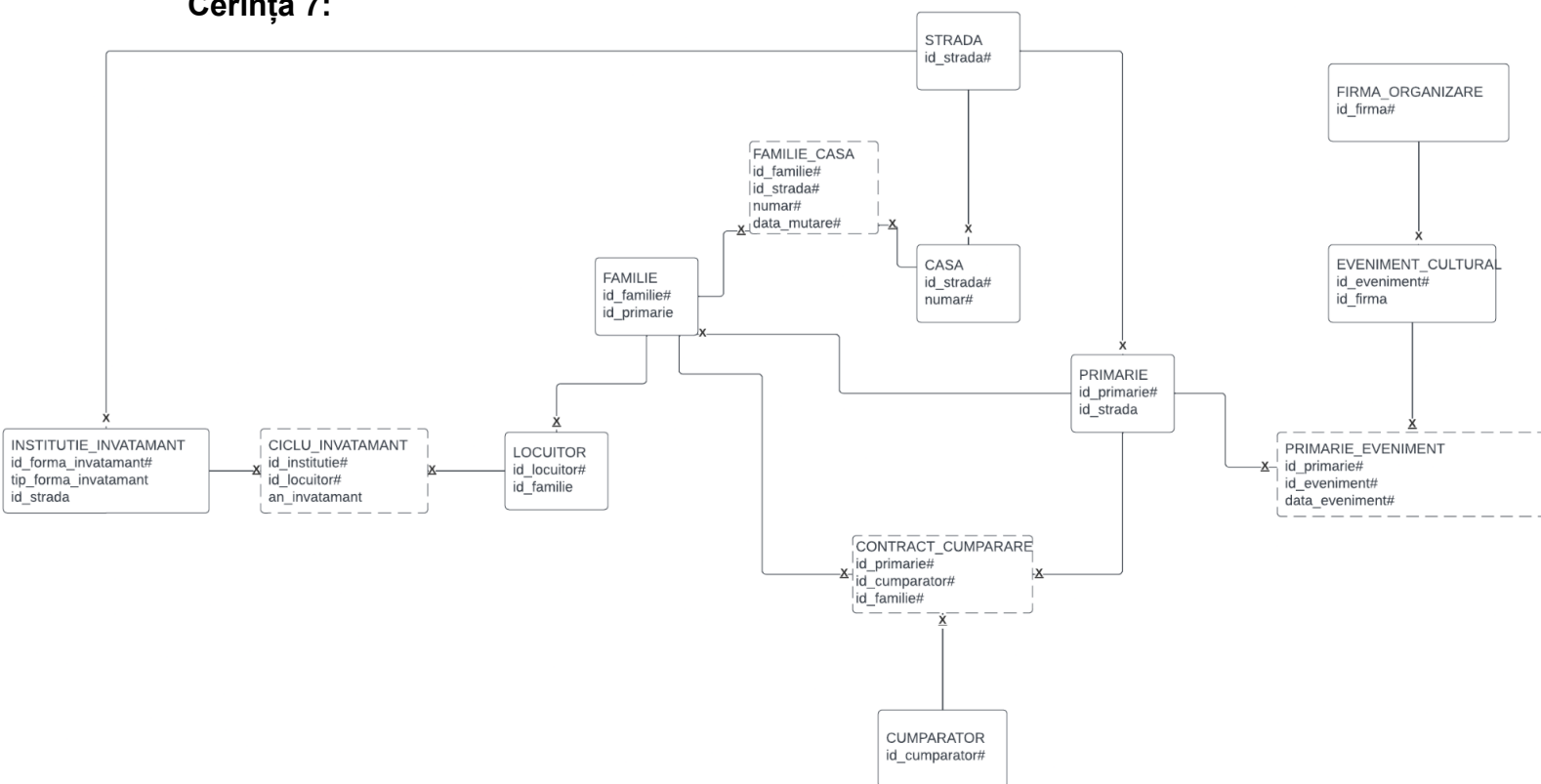
suma (variabila de tip întreg, de lungime 6. Reprezintă suma încasată de familie).

taxa (variabila de tip întreg, de lungime 2. Reprezintă procentul din sumă pe care o ia primăria).

Cerința 6:



Cerința 7:



Cerința 8:

Scheme relaționale:

STRADA (id_strada#, nume_strada)

INSTITUTIE_INVATAMANT (id_forma#, nume, tip_forma_invatamant, id_strada)

CASA (id_strada#, numar#, camere, dimensiune)

PRIMARIE (id_primarie#, nume, id_strada)

FAMILIE (id_familie#, id_primarie, nume_familie)

LOCUIITOR (id_locuitor#, id_familie, prenume, sex)

FIRMA_ORGANIZARE (id_firma#, nume)

EVENIMENT_CULTURAL (id_eveniment#, id_firma, nume_eveniment, capacitate)

CUMPARATOR (id_cumparator#, nume)

CICLU_INVATAMANT (id_forma#, id_locuitor#, an_invatamant#)

CONTRACT_CUMPARARE (id_primarie#, id_cumparator#, id_familie#,
data_cumparare#, taxa, suma)

FAMILIE_CASA (id_familie#, id_strada#, numar#, data_mutare#)

PRIMARIE_EVENTIMENT(id_primarie#, id_eveniment#, data_eveniment#)

Cerința 9:

Baza de date prezentată se află deja în forma normală 3. Așa că, în continuare, voi oferi câteva exemple care încalcă formele normale:

FN1:

Forma normală 1 ar fi încălcată în cazul în care, de exemplu, am avea la tabela FAMILIE o coloană intitulată membru_familie prin care încercăm să reținem prenumele tuturor membrilor din familie. Acest lucru ar încălca forma normală 1 prin faptul că atributul membru_familie nu este singular. Un alt exemplu de încălcare a formei normale 1 este reținerea unităților de învățământ absolvite de un locuitor. Am avea, astfel, o înșiruire de unități în loc de o singură valoare.

FN2:

O încălcare a formei normale 2 s-ar produce în cazul în care un atribut depinde de o parte a cheii primare, dar nu în totalitate de cheia primară. De exemplu, dacă în tabela FAMILIE_CASA am avea nume_familie, atunci nume_familie ar putea fi dedus din id_familie, dar nu are nicio legătură cu id_stradă sau data mutării. Un alt exemplu de încălcare al FN2 poate fi reprezentată de introducerea atributului nume(referitor la numele primăriei) în tabela PRIMARIE_EVENTIMENT. Astfel, numele depinde de id_primărie, dar nu are nicio legătură cu data_eveniment sau id_eveniment

FN3:

Forma normală 3 reprezintă dependența unui atribut non-cheie de un alt atribut non-cheie. De exemplu, această forma normală ar fi încălcată în cazul în care am avea atributul nume_familie în tabela LOCUITOR. În acest caz, numele de familie poate fi dedus direct din cheia externă id_familie, prin urmare datele legate de numele de familie apar duplicate în tabele diferite (LOCUITOR și FAMILIE). Astfel, dacă numele familiei cu id X se schimbă, cel din LOCUITOR va rămâne neschimbat.

Cerința 10:

```
DROP SEQUENCE strada_seq;  
CREATE SEQUENCE strada_seq
```

```

START with 1
INCREMENT BY 5
NOCACHE;
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Teilor');
INSERT INTO STRADA
VALUES(strada_seq.nextval1, 'Libertatii');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Viilor');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Crinului');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Tineretului');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Eroilor');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Singuratiei');

```

```

CREATE SEQUENCE id_prim
  START WITH 1
  INCREMENT BY 1
  MINVALUE 1
  MAXVALUE 5
  CYCLE
  NOCACHE;
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 1, '12-july-1998');
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 5, '6-november-2006');
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 3, '20-september-1993');
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 2, '3-may-2014');
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 1, '18-january-2002');
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 3, '9-october-2022');
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 5, '28-march-2022');
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 4, '14-june-2007');
INSERT INTO PRIMARIE_EVENT
VALUES(id_prim.nextval, 2, '27-august-1992');
INSERT INTO PRIMARIE_EVENT

```

```
VALUES(id_prim.nextval, 1, '01-may-2011');  
INSERT INTO PRIMARIE_EVENIMENT  
VALUES(id_prim.nextval, 2, '05-aug-2022');  
INSERT INTO PRIMARIE_EVENIMENT  
VALUES(id_prim.nextval, 3, '29-sep-2022');
```

Cerința 11:

```
CREATE TABLE STRADA(  
    id_strada NUMBER(3) PRIMARY KEY,  
    nume_strada VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE INSTITUTIE_INVATAMANT (  
    id_forma NUMBER(3) PRIMARY KEY,  
    nume VARCHAR(50) NOT NULL,  
    tip_forma_invatamant CHAR(40) NOT NULL,  
    id_strada REFERENCES STRADA  
);
```

```
CREATE TABLE CASA (  
    id_strada REFERENCES STRADA,  
    numar NUMBER(3) NOT NULL CHECK (numar > 0 AND numar < 50),  
    camere NUMBER(2) DEFAULT 1 NOT NULL,  
    dimensiune NUMBER(4) NOT NULL,  
    PRIMARY KEY(id_strada, numar)  
);
```

```
CREATE TABLE PRIMARIE (  
    id_primarie NUMBER(3) PRIMARY KEY,  
    nume VARCHAR(100) NOT NULL,  
    id_strada REFERENCES STRADA,  
    CONSTRAINT primarie_locatie_unic UNIQUE (id_strada)  
);
```

```
CREATE TABLE FAMILIE(  
    id_familie NUMBER(4) PRIMARY KEY,  
    id_primarie REFERENCES PRIMARIE,  
    nume_familie VARCHAR(25) NOT NULL  
);
```

```
CREATE TABLE LOCUITOR (  
    id_locuitor NUMBER(5) PRIMARY KEY,  
    id_familie NUMBER(4) REFERENCES FAMILIE(id_familie),
```

```
    prenume VARCHAR(50) NOT NULL,  
    sex NUMBER(1) CHECK (sex > 0 AND sex < 3)  
);
```

```
CREATE TABLE FIRMA_ORGANIZARE(  
    id_firma NUMBER(3) PRIMARY KEY,  
    nume VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE EVENIMENT_CULTURAL(  
    id_eveniment NUMBER(3) PRIMARY KEY,  
    id_firma REFERENCES FIRMA_ORGANIZARE,  
    nume_eveniment VARCHAR(25) NOT NULL,  
    capacitate NUMBER(7) NOT NULL  
);
```

```
CREATE TABLE CUMPARATOR(  
    id_cumparator NUMBER(3) PRIMARY KEY,  
    nume VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE CICLU_INVATAMANT(  
    id_forma REFERENCES INSTITUTIE_INVATAMANT,  
    id_locuitor REFERENCES locuitor,  
    an_invatamant DATE NOT NULL,  
    PRIMARY KEY (id_forma, id_locuitor, an_invatamant)  
);
```

```
CREATE TABLE CONTRACT_CUMPARARE(  
    id_primarie REFERENCES PRIMARIE,  
    id_cumparator REFERENCES CUMPARATOR,  
    id_familie REFERENCES FAMILIE,  
    data_cumparare DATE NOT NULL,  
    taxa NUMBER(3) DEFAULT NULL,  
    suma NUMBER(8) NOT NULL,  
    PRIMARY KEY (id_primarie, id_cumparator, id_familie, data_cumparare)  
);
```

```
CREATE TABLE FAMILIE_CASA(  
    id_familie NUMBER(4) NOT NULL,  
    id_strada NUMBER(3),  
    numar NUMBER(3),  
    data_mutare DATE NOT NULL,  
    PRIMARY KEY (id_familie, id_strada, numar, data_mutare),
```

```
FOREIGN KEY (id_strada, numar) REFERENCES CASA(id_strada,
numar),
FOREIGN KEY (id_familie) REFERENCES FAMILIE(id_familie)
);
```

```
CREATE TABLE PRIMARIE_EVENTIMENT(
id_primarie REFERENCES PRIMARIE,
id_eventiment REFERENCES EVENTIMENT_CULTURAL,
data_eventiment DATE NOT NULL,
PRIMARY KEY(id_primarie, id_eventiment, data_eventiment)
);
```

```
DROP SEQUENCE strada_seq;
CREATE SEQUENCE strada_seq
START with 1
INCREMENT BY 5
NOCACHE;
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Teilor');
INSERT INTO STRADA
VALUES(strada_seq.nextval1, 'Libertatii');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Viilor');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Crinului');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Tineretului');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Eroilor');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Singuratatii');
```

```
INSERT INTO CASA
VALUES(1, 1, 5, 420);
INSERT INTO CASA
VALUES(11, 2, 2, 210);
INSERT INTO CASA
VALUES(16, 14, 1, 100);
INSERT INTO CASA
VALUES(21, 4, 3, 250);
INSERT INTO CASA
VALUES(21, 5, 2, 260);
INSERT INTO CASA
VALUES(16, 25, 3, 420);
```

```
INSERT INTO PRIMARIE
VALUES(1, 'Primaria Viilor', 6);
INSERT INTO PRIMARIE
VALUES(2, 'Primaria Tineretului', 16);
INSERT INTO PRIMARIE
VALUES(3, 'Primaria Crinului', 11);
INSERT INTO PRIMARIE
VALUES(4, 'Primaria Eroilor', 21);
INSERT INTO PRIMARIE
VALUES(5, 'Primaria Teilor', 1);
INSERT INTO PRIMARIE
VALUES(6, 'Primaria Degeaba', 26);
```

```
INSERT INTO FAMILIE
VALUES(5, 4, 'Popescu');
INSERT INTO FAMILIE
VALUES(10, 1, 'Ionescu');
INSERT INTO FAMILIE
VALUES(15, 1, 'Popa');
INSERT INTO FAMILIE
VALUES(20, 4, 'Constantinescu');
INSERT INTO FAMILIE
VALUES(25, 5, 'Tupeiescu');
INSERT INTO FAMILIE
VALUES(30, 2, 'Vecinescu');
```

```
INSERT INTO LOCUITOR
VALUES (1, 5, 'Ion', 1);
INSERT INTO LOCUITOR
VALUES (2, 5, 'Maria', 2);
INSERT INTO LOCUITOR
VALUES (3, 5, 'Ana', 2);
INSERT INTO LOCUITOR
VALUES (4, 20, 'Mihai', 1);
INSERT INTO LOCUITOR
VALUES (5, 20, 'Elena', 2);
INSERT INTO LOCUITOR
VALUES(6, 25, 'Costache', 1);
INSERT INTO LOCUITOR
VALUES(7, 25, 'Mihaela', 2);
INSERT INTO LOCUITOR
VALUES(8, 30, 'Ramon', 1);
```

```
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (1, 'Scoala Gimnaziala Nr. 1', 'Scoala Gimnaziala', 1);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (2, 'Liceul Teoretic "Mihai Eminescu"', 'Liceu', 1);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (3, 'Scoala Gimnaziala Nr. 2', 'Scoala Gimnaziala', 6);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (4, 'Colegiul National "Gheorghe Lazar"', 'Colegiu', 16);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (5, 'Liceul Tehnic', 'Liceu', 21);
```

```
INSERT INTO CUMPARATOR
VALUES (1, 'ABC Company');
INSERT INTO CUMPARATOR
VALUES (2, 'XYZ Corporation');
INSERT INTO CUMPARATOR
VALUES (3, '123 Industries');
INSERT INTO CUMPARATOR
VALUES (4, 'Acme Enterprises');
INSERT INTO CUMPARATOR
VALUES (5, 'Global Solutions');
```

```
INSERT INTO FIRMA_ORGANIZARE
VALUES (1, 'Event Management Company');
INSERT INTO FIRMA_ORGANIZARE
VALUES (2, 'Art Agency');
INSERT INTO FIRMA_ORGANIZARE
VALUES (3, 'Production House');
INSERT INTO FIRMA_ORGANIZARE
VALUES (4, 'Entertainment Solutions');
INSERT INTO FIRMA_ORGANIZARE
VALUES (5, 'Creative Events');
```

```
INSERT INTO EVENIMENT_CULTURAL
VALUES (1, 1, 'Concert Live', 1000);
INSERT INTO EVENIMENT_CULTURAL
VALUES (2, 5, 'Expozitie de arta', 500);
INSERT INTO EVENIMENT_CULTURAL
VALUES (3, 5, 'Teatru in aer liber', 300);
INSERT INTO EVENIMENT_CULTURAL
VALUES (4, 1, 'Dans contemporan', 800);
INSERT INTO EVENIMENT_CULTURAL
```

VALUES (5, 4, 'Film in aer liber', 200);

INSERT INTO CICLU_INVATAMANT
VALUES(1, 1, '15-sep-1985');
INSERT INTO CICLU_INVATAMANT
VALUES(5, 1, '20-sep-1993');
INSERT INTO CICLU_INVATAMANT
VALUES(3, 2, '15-sep-1985');
INSERT INTO CICLU_INVATAMANT
VALUES(4, 2, '20-sep-1993');
INSERT INTO CICLU_INVATAMANT
VALUES(3, 3, '29-aug-2003');
INSERT INTO CICLU_INVATAMANT
VALUES(4, 3, '30-sep-2011');
INSERT INTO CICLU_INVATAMANT
VALUES(1, 4, '29-aug-2003');
INSERT INTO CICLU_INVATAMANT
VALUES(4, 4, '30-sep-2011');
INSERT INTO CICLU_INVATAMANT
VALUES(3, 5, '11-sep-2002');
INSERT INTO CICLU_INVATAMANT
VALUES(5, 5, '19-sep-2010');
INSERT INTO CICLU_INVATAMANT
VALUES(4, 6, '20-may-2003');

INSERT INTO FAMILIE_CASA
VALUES(5, 16, 14, '10-may-1980');
INSERT INTO FAMILIE_CASA
VALUES(10, 11, 2, '10-april-1985');
INSERT INTO FAMILIE_CASA
VALUES(5, 21, 4, '9-april-1985');
INSERT INTO FAMILIE_CASA
VALUES(20, 21, 5, '29-aug-2003');
INSERT INTO FAMILIE_CASA
VALUES(15, 21, 4, '15-feb-1995');
INSERT INTO FAMILIE_CASA
VALUES(15, 1, 1, '20-jan-1997');
INSERT INTO FAMILIE_CASA
VALUES(10, 1, 1, '14-jan-2001');
INSERT INTO FAMILIE_CASA
VALUES(15, 11, 2, '10-june-2000');
INSERT INTO FAMILIE_CASA
VALUES(20, 21, 5, '19-jul-2005');
INSERT INTO FAMILIE_CASA


```
VALUES(20, 1, 1, '20-march-2002');
INSERT INTO FAMILIE_CASA
VALUES(10, 21, 4, '15-march-2002');
INSERT INTO FAMILIE_CASA
VALUES(25, 16, 14, '5-june-2001');
INSERT INTO FAMILIE_CASA
VALUES(30, 16, 25, '3-may-2001');
```

```
INSERT INTO CONTRACT_CUMPARARE
VALUES(1, 1, 5, '15-sep-1990', 15, 120000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(1, 2, 5, '20-jan-1985', 20, 17000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(3, 1, 20, '15-aug-2019', 5, 7000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(2, 5, 25, '14-sep-2023', 40, 800000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(3, 4, 20, '19-jul-2021', 14, 75189);
INSERT INTO CONTRACT_CUMPARARE
VALUES(2, 2, 25, '11-june-2021', 10, 10000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(1, 1, 5, '15-sep-2004', 5, 6510);
INSERT INTO CONTRACT_CUMPARARE
VALUES(3, 2, 20, '14-sep-2022', 10, 19000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(2, 1, 25, '19-oct-2022', 4, 7000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(1, 3, 15, '18-nov-2022', 10, 19500);
INSERT INTO CONTRACT_CUMPARARE
VALUES(2, 3, 10, '20-aug-2001', NULL, 10000);
```

```
CREATE SEQUENCE id_prim
  START WITH 1
  INCREMENT BY 1
  MINVALUE 1
  MAXVALUE 5
  CYCLE
  NOCACHE;
INSERT INTO PRIMARIE_EVENIMENT
VALUES(id_prim.nextval, 1, '12-july-1998');
INSERT INTO PRIMARIE_EVENIMENT
VALUES(id_prim.nextval, 5, '6-november-2006');
INSERT INTO PRIMARIE_EVENIMENT
```

```

VALUES(id_prim.nextval, 3, '20-september-1993');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 2, '3-may-2014');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 1, '18-january-2002');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 3, '9-october-2022');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 5, '28-march-2022');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 4, '14-june-2007');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 2, '27-august-1992');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 1, '01-may-2011');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 2, '05-aug-2022');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 3, '29-sep-2022');

```

Poze pentru cerințele 10 + 11:

```

CREATE TABLE STRADA (
    id_strada NUMBER(3) PRIMARY KEY,
    nume_strada VARCHAR(100) NOT NULL
);

CREATE TABLE INSTITUTIE_INVATAMANT (
    id_forma NUMBER(3) PRIMARY KEY,
    nume VARCHAR(50) NOT NULL,
    tip_forma_invatamant CHAR(40) NOT NULL,
    id_strada REFERENCES STRADA
);

CREATE TABLE CASA (
    id_strada REFERENCES STRADA,
    numar NUMBER(3) NOT NULL CHECK (numar > 0 AND numar < 50),
    camere NUMBER(2) DEFAULT 1 NOT NULL,
    dimensiune NUMBER(4) NOT NULL,
    PRIMARY KEY(id_strada, numar)
);

CREATE TABLE PRIMARIE (
    id_primarie NUMBER(3) PRIMARY KEY,

```

Script Output x Query Result x

Task completed in 0.019 seconds

Table PRIMARIE dropped.

Table STRADA dropped.

Table STRADA created.

Activate Windows
Go to Settings to activate Windows.

```

);

CREATE TABLE INSTITUTIE_INVATAMANT (
  id_forma NUMBER(3) PRIMARY KEY,
  nume VARCHAR(50) NOT NULL,
  tip_forma_invatamant CHAR(40) NOT NULL,
  id_strada REFERENCES STRADA
);

CREATE TABLE CASA (
  id_strada REFERENCES STRADA,
  numar NUMBER(3) NOT NULL CHECK (numar > 0 AND numar < 50),
  camere NUMBER(2) DEFAULT 1 NOT NULL,
  dimensiune NUMBER(4) NOT NULL,
  PRIMARY KEY(id_strada, numar)
);

CREATE TABLE PRIMARIE (
  id primarie NUMBER(3) PRIMARY KEY,

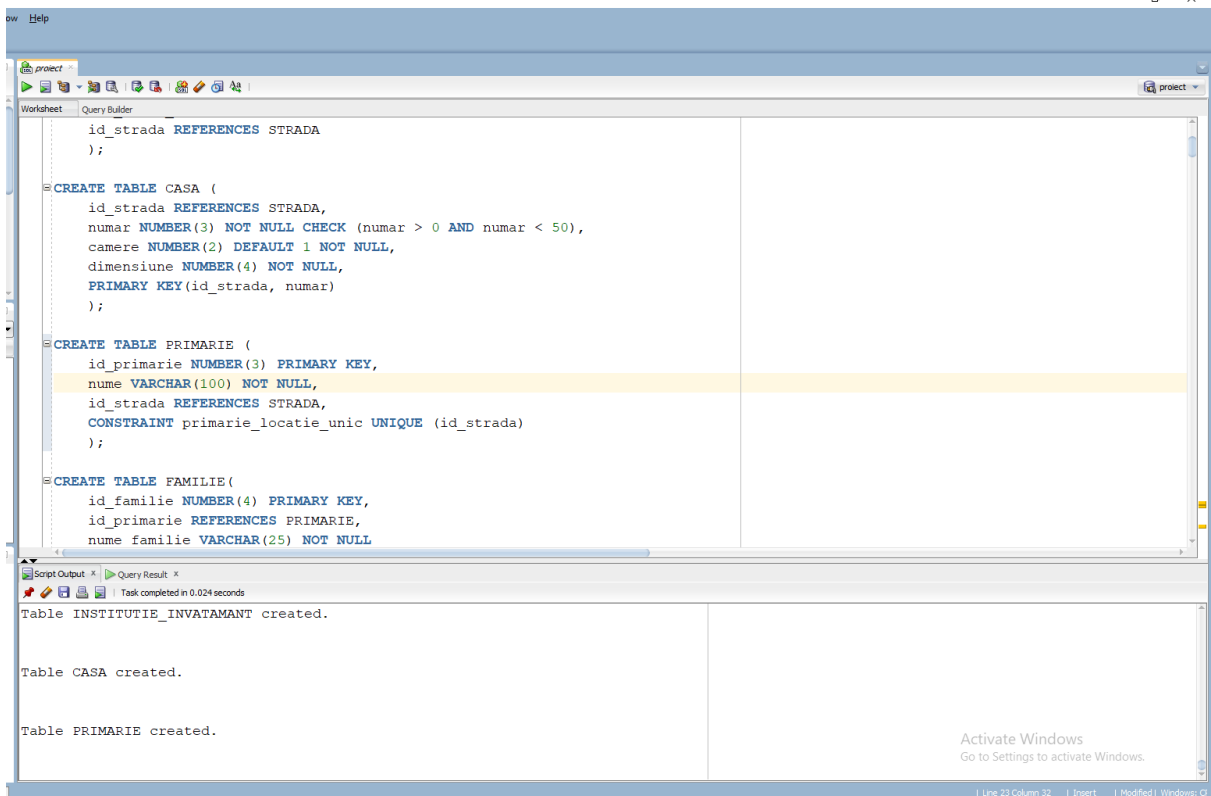
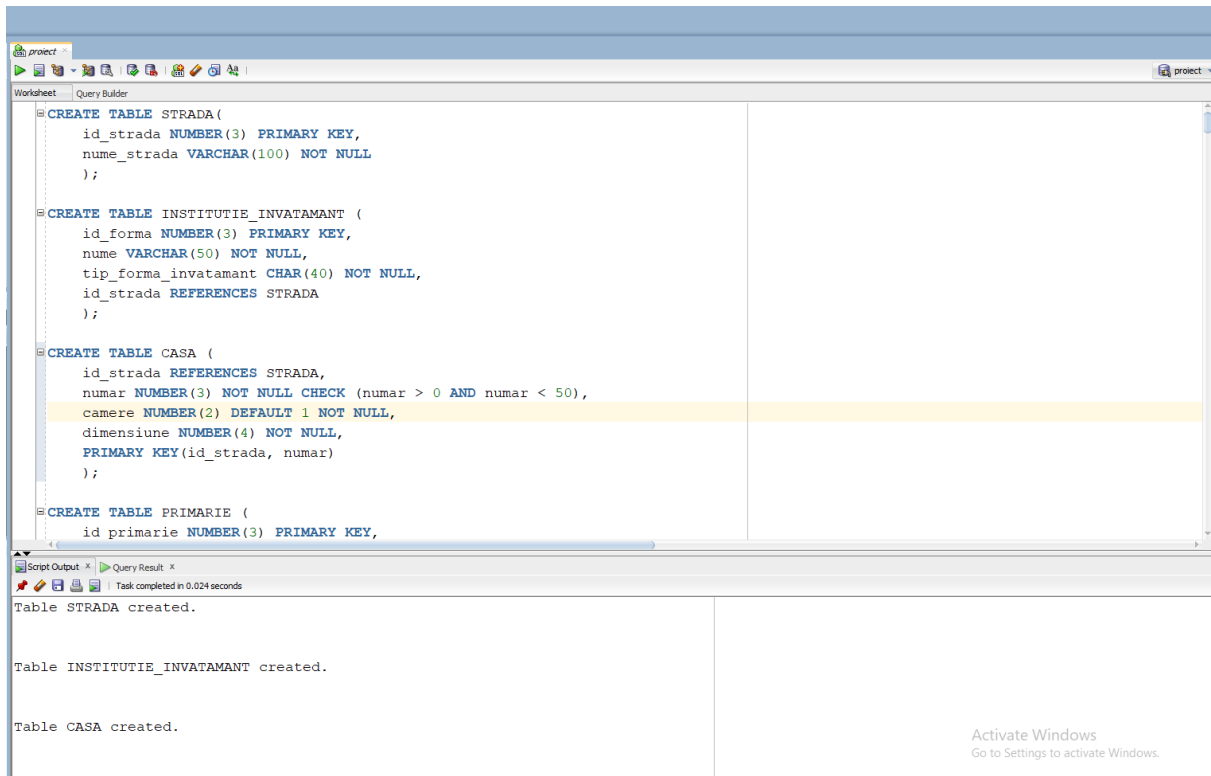
```

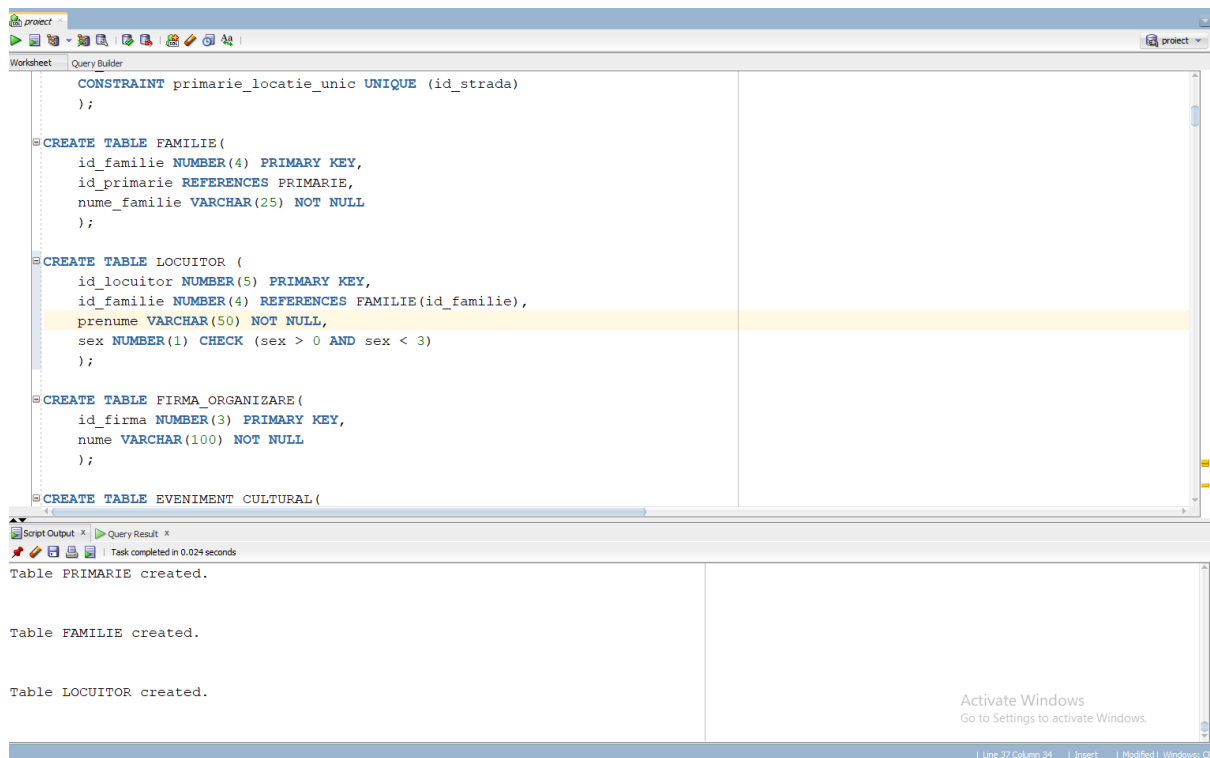
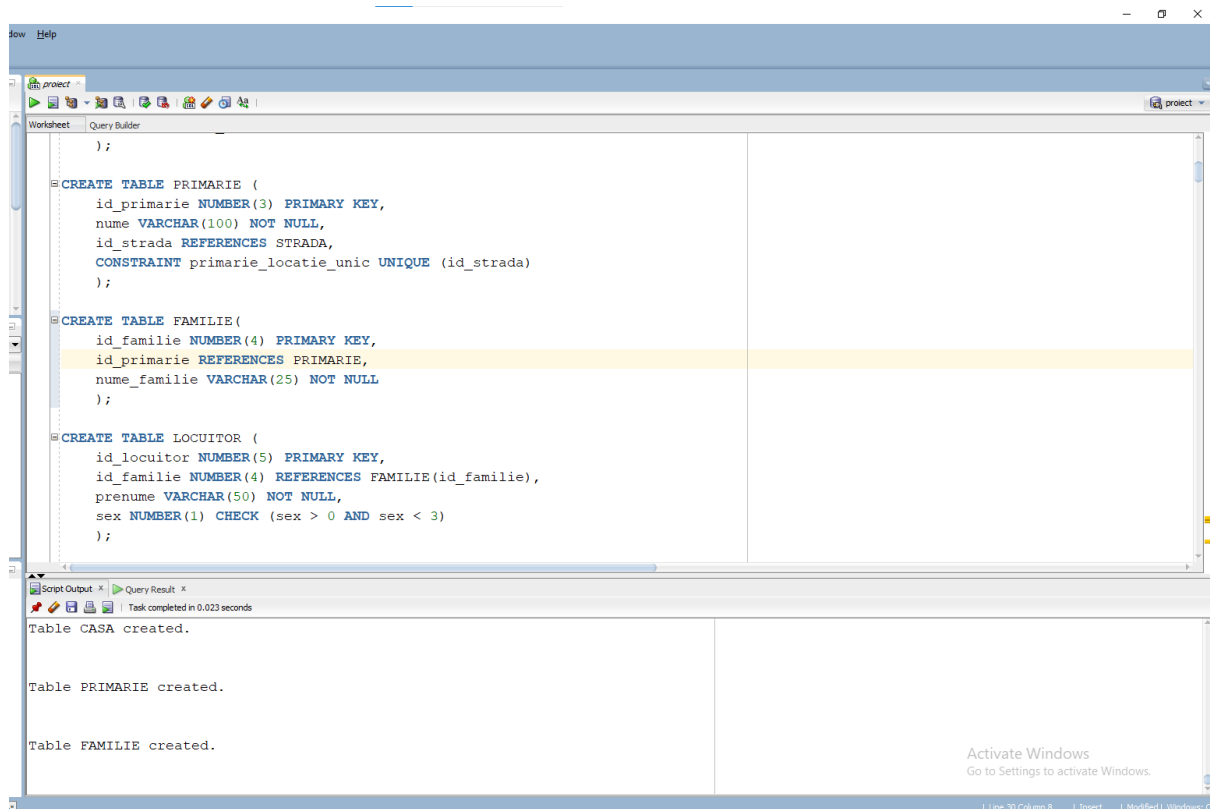
Script Output x Query Result x
 Task completed in 0.022 seconds

Table STRADA dropped.

Table STRADA created.

Table INSTITUTIE_INVATAMANT created.





```

CREATE TABLE FAMILIE(
    id_familie NUMBER(4) PRIMARY KEY,
    id_primarie REFERENCES PRIMARIE,
    nume_familie VARCHAR(25) NOT NULL
);

CREATE TABLE LOCUITOR (
    id_locutor NUMBER(5) PRIMARY KEY,
    id_familie NUMBER(4) REFERENCES FAMILIE(id_familie),
    prenume VARCHAR(50) NOT NULL,
    sex NUMBER(1) CHECK (sex > 0 AND sex < 3)
);

CREATE TABLE FIRMA_ORGANIZARE(
    id_firma NUMBER(3) PRIMARY KEY,
    nume VARCHAR(100) NOT NULL
);

CREATE TABLE EVENIMENT_CULTURAL(
    id_eveniment NUMBER(3) PRIMARY KEY,
    id_firma REFERENCES FIRMA_ORGANIZARE,
    nume_eveniment VARCHAR(25) NOT NULL,

```

Script Output x Query Result x
Task completed in 0.025 seconds

Table FAMILIE created.

Table LOCUITOR created.

Table FIRMA_ORGANIZARE created.

| Line 43 Column 31 | Insert | Modified | Windows: O

```

    prenume VARCHAR(50) NOT NULL,
    sex NUMBER(1) CHECK (sex > 0 AND sex < 3)
);

CREATE TABLE FIRMA_ORGANIZARE(
    id_firma NUMBER(3) PRIMARY KEY,
    nume VARCHAR(100) NOT NULL
);

CREATE TABLE EVENIMENT_CULTURAL(
    id_eveniment NUMBER(3) PRIMARY KEY,
    id_firma REFERENCES FIRMA_ORGANIZARE,
    nume_eveniment VARCHAR(25) NOT NULL,
    capacitate NUMBER(7) NOT NULL
);

CREATE TABLE CUMPARATOR(
    id_cumparator NUMBER(3) PRIMARY KEY,
    nume VARCHAR(100) NOT NULL
);

CREATE TABLE CICLU_INVATAMANT(

```

Script Output x Query Result x
Task completed in 0.023 seconds

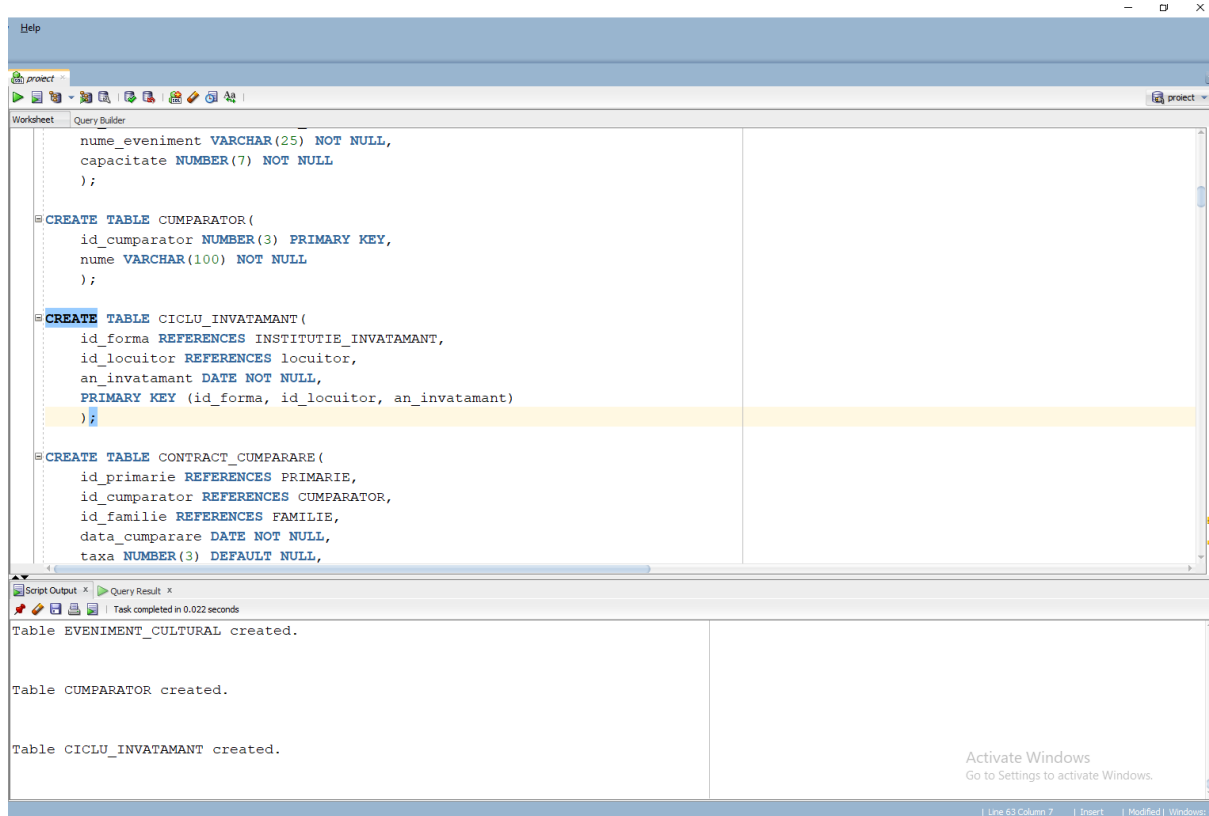
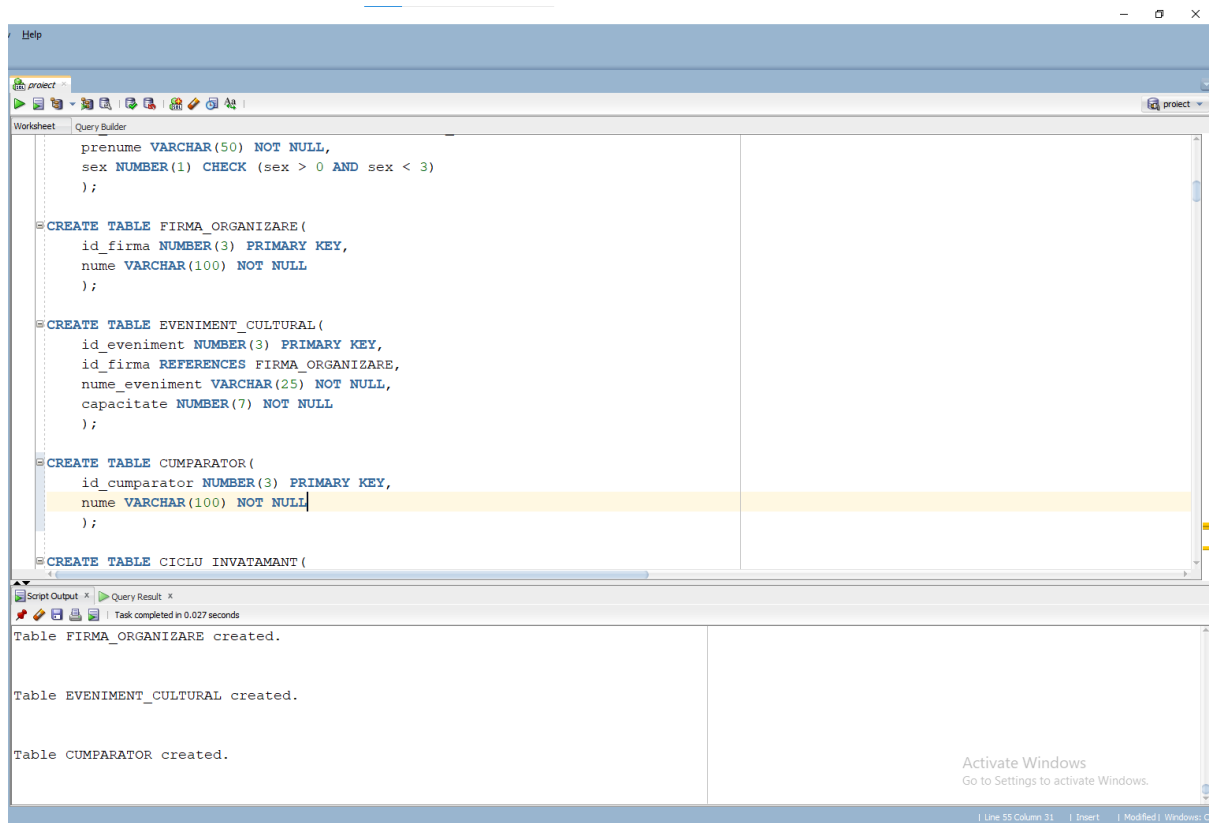
Table LOCUITOR created.

Table FIRMA_ORGANIZARE created.

Table EVENIMENT_CULTURAL created.

Activate Windows
Go to Settings to activate Windows.

| Line 47 Column 41 | Insert | Modified | Windows: O



Help

project

Worksheet Query Builder

```

CREATE TABLE CICLU_INVATAMANT (
  id_forma REFERENCES INSTITUTIE_INVATAMANT,
  id_locutor REFERENCES locutor,
  an_invatamant DATE NOT NULL,
  PRIMARY KEY (id_forma, id_locutor, an_invatamant)
);

CREATE TABLE CONTRACT_CUMPARARE (
  id_primarie REFERENCES PRIMARIE,
  id_cumparator REFERENCES CUMPARATOR,
  id_familie REFERENCES FAMILIE,
  data_cumparare DATE NOT NULL,
  taxa NUMBER(3) DEFAULT NULL,
  suma NUMBER(8) NOT NULL,
  PRIMARY KEY (id_primarie, id_cumparator, id_familie, data_cumparare)
);

CREATE TABLE FAMILIE_CASA (
  id_familie NUMBER(4) NOT NULL,
  id_strada NUMBER(3),
  numar NUMBER(3),
  data_mutare DATE NOT NULL,

```

Script Output x Query Result x

Task completed in 0.023 seconds

Table CUMPARATOR created.

Table CICLU_INVATAMANT created.

Table CONTRACT_CUMPARARE created.

| Line 66 Column 37 | Insert | Modified | Windows: G

Help

project

Worksheet Query Builder

```

CREATE TABLE FAMILIE_CASA (
  id_familie NUMBER(4) NOT NULL,
  id_strada NUMBER(3),
  numar NUMBER(3),
  data_mutare DATE NOT NULL,
  PRIMARY KEY (id_familie, id_strada, numar, data_mutare),
  FOREIGN KEY (id_strada, numar) REFERENCES CASA(id_strada, numar),
  FOREIGN KEY (id_familie) REFERENCES FAMILIE(id_familie)
);

CREATE TABLE PRIMARIE_EVENIMENT (
  id_primarie REFERENCES PRIMARIE,
  id_eveniment REFERENCES EVENIMENT_CULTURAL,
  data_eveniment DATE NOT NULL,
  PRIMARY KEY (id_primarie, id_eveniment, data_eveniment)
);

DROP TABLE CONTRACT_CUMPARARE;
DROP TABLE CUMPARATOR;
DROP TABLE PRIMARIE_EVENIMENT;

```

Script Output x Query Result x

Task completed in 0.021 seconds

Table CICLU_INVATAMANT created.

Table CONTRACT_CUMPARARE created.

Table FAMILIE_CASA created.

| Line 77 Column 25 | Insert | Modified | Windows: G

Help

Worksheet Query Builder

```
id_familie NUMBER(4) NOT NULL,  
id_strada NUMBER(3),  
numar NUMBER(3),  
data_mutare DATE NOT NULL,  
PRIMARY KEY (id_familie, id_strada, numar, data_mutare),  
FOREIGN KEY (id_strada, numar) REFERENCES CASA(id_strada, numar),  
FOREIGN KEY (id_familie) REFERENCES FAMILIE(id_familie)  
);  
  
CREATE TABLE PRIMARIE_EVENTIMENT(  
id_primarie REFERENCES PRIMARIE,  
id_eveniment REFERENCES EVENIMENT_CULTURAL,  
data_eveniment DATE NOT NULL,  
PRIMARY KEY(id_primarie, id_eveniment, data_eveniment)  
);  
  
DROP TABLE CONTRACT_CUMPARARE;  
DROP TABLE CUMPARATOR;  
DROP TABLE PRIMARIE_EVENTIMENT;  
DROP TABLE EVENIMENT_CULTURAL;  
DROP TABLE FIRMA_ORGANIZARE;  
DROP TABLE CICLU_INVATAMANT;  
DROP TABLE INSCRIERILE_INVATAMANT;
```

Script Output x Query Result x

Task completed in 0.024 seconds

Table CONTRACT_CUMPARARE created.

Table FAMILIE_CASA created.

Table PRIMARIE_EVENTIMENT created.

Activate Windows
Go to Settings to activate Windows.

Line 87 Column 44 | Insert | Modified | Windows

Help

Worksheet Query Builder

```
CREATE SEQUENCE strada_seq  
START with 1  
INCREMENT BY 5  
NOCACHE;  
INSERT INTO STRADA  
VALUES(strada_seq.nextval, 'Teilor');  
INSERT INTO STRADA  
VALUES(strada_seq.nextval, 'Libertatii');  
INSERT INTO STRADA  
VALUES(strada_seq.nextval, 'Viilor');  
INSERT INTO STRADA  
VALUES(strada_seq.nextval, 'Crinului');  
INSERT INTO STRADA  
VALUES(strada_seq.nextval, 'Tineretului');  
INSERT INTO STRADA  
VALUES(strada_seq.nextval, 'Eroilor');  
INSERT INTO STRADA  
VALUES(strada_seq.nextval, 'Singuratiei');  
  
SELECT * FROM strada;  
  
INSERT INTO CASA
```

Script Output x Query Result x

All Rows Fetched: 6 in 0.001 seconds

ID_STRADA	NUME_STRADA
1	Teilor
6	Viilor
11	Crinului
16	Tineretului
21	Eroilor
26	Singuratiei

Activate Windows
Go to Settings to activate Windows.

Help

project

WorksheetQuery Builder

VALUES(strada_seq.nextval, 'Eroilor');
INSERT INTO STRADA
VALUES(strada_seq.nextval, 'Singuratatii');

INSERT INTO CASA
VALUES(1, 1, 5, 420);
INSERT INTO CASA
VALUES(11, 2, 2, 210);
INSERT INTO CASA
VALUES(16, 14, 1, 100);
INSERT INTO CASA
VALUES(21, 4, 3, 250);
INSERT INTO CASA
VALUES(21, 5, 2, 260);
INSERT INTO CASA
VALUES(16, 25, 3, 420);

SELECT * FROM CASA;

INSERT INTO PRIMARIE
VALUES(1, 'Primaria Viilor', 6);
INSERT INTO PRIMARIE

Script OutputQuery Result

SQLAll Rows Fetched: 6 in 0.002 seconds

ID_STRADA	NUMAR	CAMERE	DIMENSIONE
1	1	5	420
11	2	2	210
16	14	1	100
21	4	3	250
21	5	2	260
16	25	3	420

Activate Windows
Go to Settings to activate Windows.

Line 139 Column 20InsertModifiedWindows

Help

project

Worksheet Query Builder

```
INSERT INTO CASA
VALUES(16, 25, 3, 420);

INSERT INTO PRIMARIE
VALUES(1, 'Primaria Viilor', 6);
INSERT INTO PRIMARIE
VALUES(2, 'Primaria Tineretului', 16);
INSERT INTO PRIMARIE
VALUES(3, 'Primaria Crinului', 11);
INSERT INTO PRIMARIE
VALUES(4, 'Primaria Eroilor', 21);
INSERT INTO PRIMARIE
VALUES(5, 'Primaria Teilor', 1);
INSERT INTO PRIMARIE
VALUES(6, 'Primaria Degeaba', 26);

SELECT * FROM PRIMARIE;

INSERT INTO FAMILIE
VALUES(5, 4, 'Popescu');
INSERT INTO FAMILIE
VALUES(10, 1, 'Ionescu');
INSERT INTO FAMILIE
```

Script Output x Query Result x

All Rows Fetched: 6 in 0 seconds

ID_PRIMARIE	NUME	ID_STRADA
1	1Primaria Viilor	6
2	2Primaria Tineretului	16
3	3Primaria Crinului	11
4	4Primaria Eroilor	21
5	5Primaria Teilor	1
6	6Primaria Degeaba	26

Activate Windows
Go to Settings to activate Windows.

Line 152 Column 24 | Insert | Modified | Windows: C

ow Help

project

Worksheet Query Builder

```
VALUES(6, 'Primaria Degeaba', 26);

INSERT INTO FAMILIE
VALUES(5, 4, 'Popescu');
INSERT INTO FAMILIE
VALUES(10, 1, 'Ionescu');
INSERT INTO FAMILIE
VALUES(15, 1, 'Popa');
INSERT INTO FAMILIE
VALUES(20, 4, 'Constantinescu');
INSERT INTO FAMILIE
VALUES(25, 5, 'Tupeiescu');
INSERT INTO FAMILIE
VALUES(30, 2, 'Vecinescu');

SELECT * FROM FAMILIE;

INSERT INTO LOCUITOR
VALUES(1, 5, 'Ion', 1);
INSERT INTO LOCUITOR
VALUES(2, 5, 'Maria', 2);
INSERT INTO LOCUITOR
```

Script Output x Query Result x

All Rows Fetched: 6 in 0.002 seconds

ID_FAMILIE	ID_PRIMARIE	NUME_FAMILIE
1	5	4Popescu
2	10	1Ionescu
3	15	1Popa
4	20	4Constantinescu
5	25	5Tupeiescu
6	30	2Vecinescu

Activate Windows
Go to Settings to activate Windows.

Line 165 Column 23 | Insert | Modified | Windows: C

Help

project

Worksheet Query Builder

```
VALUES (30, 2, 'Vecinescu');

INSERT INTO LOCUTOR
VALUES (1, 5, 'Ion', 1);
INSERT INTO LOCUTOR
VALUES (2, 5, 'Maria', 2);
INSERT INTO LOCUTOR
VALUES (3, 5, 'Ana', 2);
INSERT INTO LOCUTOR
VALUES (4, 20, 'Mihai', 1);
INSERT INTO LOCUTOR
VALUES (5, 20, 'Elena', 2);
INSERT INTO LOCUTOR
VALUES (6, 25, 'Costache', 1);
INSERT INTO LOCUTOR
VALUES (7, 25, 'Mihaela', 2);
INSERT INTO LOCUTOR
VALUES (8, 30, 'Ramon', 1);

SELECT * FROM LOCUTOR;
```

Script Output x Query Result x

SQL All Rows Fetched: 8 in 0.002 seconds

	ID_LOCUTOR	ID_FAMILIE	PRENUME	SEX
1	1	5	Ion	1
2	2	5	Maria	2
3	3	5	Ana	2
4	4	20	Mihai	1
5	5	20	Elena	2
6	6	25	Costache	1
7	7	25	Mihaela	2
8	8	30	Ramon	1

Activate Windows
Go to Settings to activate Windows.

Line 182 Column 24 | Insert | Modified | Windows: C

Help

project

Worksheet Query Builder

```
VALUES (7, 25, 'Mihaela', 2);
INSERT INTO LOCUTOR
VALUES (8, 30, 'Ramon', 1);

INSERT INTO INSTITUTIE_INVATAMANT
VALUES (1, 'Scoala Gimnaziala Nr. 1', 'Scoala Gimnaziala', 1);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (2, 'Liceul Teoretic "Mihai Eminescu"', 'Liceu', 1);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (3, 'Scoala Gimnaziala Nr. 2', 'Scoala Gimnaziala', 6);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (4, 'Colegiul National "Gheorghe Lazar"', 'Colegiu', 16);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (5, 'Liceul Tehnic', 'Liceu', 21);

SELECT * FROM INSTITUTIE_INVATAMANT;

INSERT INTO CUMPARATOR
VALUES (1, 'ABC Company');
INSERT INTO CUMPARATOR
VALUES (2, 'XYZ Corporation');
INSERT INTO CUMPARATOR
```

Script Output x Query Result x

SQL All Rows Fetched: 5 in 0.003 seconds

	ID_FORMA	NUME	TIPI_FORMA_INVATAMANT	ID_STRADA
1	1	Scoala Gimnaziala Nr. 1	Scoala Gimnaziala	1
2	2	Liceul Teoretic "Mihai Eminescu"	Liceu	1
3	3	Scoala Gimnaziala Nr. 2	Scoala Gimnaziala	6
4	4	Colegiul National "Gheorghe Lazar"	Colegiu	16
5	5	Liceul Tehnic	Liceu	21

Activate Windows
Go to Settings to activate Windows.

Line 193 Column 37 | Insert | Modified | Windows: C

Worksheet | Query Builder

```
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (4, 'Colegiul National "Gheorghe Lazar"', 'Colegiu', 16);
INSERT INTO INSTITUTIE_INVATAMANT
VALUES (5, 'Liceul Tehnic', 'Liceu', 21);

INSERT INTO CUMPARATOR
VALUES (1, 'ABC Company');
INSERT INTO CUMPARATOR
VALUES (2, 'XYZ Corporation');
INSERT INTO CUMPARATOR
VALUES (3, '123 Industries');
INSERT INTO CUMPARATOR
VALUES (4, 'Acme Enterprises');
INSERT INTO CUMPARATOR
VALUES (5, 'Global Solutions');

SELECT * FROM CUMPARATOR;

INSERT INTO FIRMA_ORGANIZARE
VALUES (1, 'Event Management Company');
INSERT INTO FIRMA_ORGANIZARE
VALUES (2, 'Art Agency');
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0 seconds

ID_CUMPARATOR	NUME
1	1 ABC Company
2	2 XYZ Corporation
3	3 123 Industries
4	4 Acme Enterprises
5	5 Global Solutions

Activate Windows
Go to Settings to activate Windows.

| Line 204 Column 26 | Insert | Modified | Windows: C

Help

Worksheet | Query Builder

```
INSERT INTO CUMPARATOR
VALUES (3, '123 Industries');
INSERT INTO CUMPARATOR
VALUES (4, 'Acme Enterprises');
INSERT INTO CUMPARATOR
VALUES (5, 'Global Solutions');

INSERT INTO FIRMA_ORGANIZARE
VALUES (1, 'Event Management Company');
INSERT INTO FIRMA_ORGANIZARE
VALUES (2, 'Art Agency');
INSERT INTO FIRMA_ORGANIZARE
VALUES (3, 'Production House');
INSERT INTO FIRMA_ORGANIZARE
VALUES (4, 'Entertainment Solutions');
INSERT INTO FIRMA_ORGANIZARE
VALUES (5, 'Creative Events');

SELECT * FROM FIRMA_ORGANIZARE;

INSERT INTO EVENIMENT_CULTURAL
VALUES (1, 1, 'Concert Live', 1000);
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.002 seconds

ID_FIRMA	NUME
1	1 Event Management Company
2	2 Art Agency
3	3 Production House
4	4 Entertainment Solutions
5	5 Creative Events

Activate Windows
Go to Settings to activate Windows.

| Line 215 Column 32 | Insert | Modified | Windows: C

project

Worksheet Query Builder

```
VALUES (3, 'Production House');
INSERT INTO FIRMA_ORGANIZARE
VALUES (4, 'Entertainment Solutions');
INSERT INTO FIRMA_ORGANIZARE
VALUES (5, 'Creative Events');

INSERT INTO EVENIMENT_CULTURAL
VALUES (1, 1, 'Concert Live', 1000);
INSERT INTO EVENIMENT_CULTURAL
VALUES (2, 5, 'Expozitie de arta', 500);
INSERT INTO EVENIMENT_CULTURAL
VALUES (3, 5, 'Teatru in aer liber', 300);
INSERT INTO EVENIMENT_CULTURAL
VALUES (4, 1, 'Dans contemporan', 800);
INSERT INTO EVENIMENT_CULTURAL
VALUES (5, 4, 'Film in aer liber', 200);

SELECT * FROM EVENIMENT_CULTURAL;

INSERT INTO CICLU_INVATAMANT
VALUES(1, 1, '15-sep-1985');
INSERT INTO CICLU_INVATAMANT
```

Script Output x Query Result x

SQL All Rows Fetched: 5 in 0.002 seconds

ID_EVENIMENT	ID_FIRMA	NUME_EVENIMENT	CAPACITATE
1	1	1Concert Live	1000
2	5	5Expozitie de arta	500
3	5	5Teatru in aer liber	300
4	1	1Dans contemporan	800
5	4	4Film in aer liber	200

Activate Windows
Go to Settings to activate Windows.

| Line 226 Column 34 | Insert | Modified | Windows: C

project

Worksheet Query Builder

```
INSERT INTO CICLU_INVATAMANT
VALUES(3, 2, '15-sep-1985');
INSERT INTO CICLU_INVATAMANT
VALUES(4, 2, '20-sep-1993');
INSERT INTO CICLU_INVATAMANT
VALUES(3, 3, '29-aug-2003');
INSERT INTO CICLU_INVATAMANT
VALUES(4, 3, '30-sep-2011');
INSERT INTO CICLU_INVATAMANT
VALUES(1, 4, '29-aug-2003');
INSERT INTO CICLU_INVATAMANT
VALUES(4, 4, '30-sep-2011');
INSERT INTO CICLU_INVATAMANT
VALUES(3, 5, '11-sep-2002');
INSERT INTO CICLU_INVATAMANT
VALUES(5, 5, '19-sep-2010');
INSERT INTO CICLU_INVATAMANT
VALUES(4, 6, '20-may-2003');

SELECT * FROM CICLU_INVATAMANT;

INSERT INTO FAMILIE_CASA
```

Script Output x Query Result x

SQL All Rows Fetched: 11 in 0.003 seconds

ID_FORMA	ID_LOCUTOR	AN_INVATAMANT
1	1	1 15-SEP-85
2	5	1 20-SEP-93
3	3	2 15-SEP-85
4	4	2 20-SEP-93
5	3	3 29-AUG-03
6	4	3 30-SEP-11
7	1	4 29-AUG-03
8	4	4 30-SEP-11
9	3	5 11-SEP-02
10	5	5 19-SEP-10

Activate Windows
Go to Settings to activate Windows.

| Line 249 Column 32 | Insert | Modified | Windows: C

project

Worksheet | Query Builder

```
VALUES(10, 11, 2, '10-june-2000');  
INSERT INTO FAMILIE_CASA  
VALUES(20, 21, 5, '19-jul-2005');  
INSERT INTO FAMILIE_CASA  
VALUES(20, 1, 1, '20-march-2002');  
INSERT INTO FAMILIE_CASA  
VALUES(10, 21, 4, '15-march-2002');  
INSERT INTO FAMILIE_CASA  
VALUES(25, 16, 14, '5-june-2001');  
INSERT INTO FAMILIE_CASA  
VALUES(30, 16, 25, '3-may-2001');  
  
SELECT * FROM FAMILIE_CASA;  
  
INSERT INTO CONTRACT_CUMPARARE  
VALUES(1, 1, 5, '15-sep-1990', 15, 120000);  
INSERT INTO CONTRACT_CUMPARARE  
VALUES(1, 2, 5, '20-jan-1985', 20, 17000);  
INSERT INTO CONTRACT_CUMPARARE  
VALUES(3, 1, 20, '15-aug-2019', 5, 7000);  
INSERT INTO CONTRACT_CUMPARARE  
VALUES(2, 5, 25, '14-sep-2023', 40, 800000);  
INSERT INTO CONTRACT_CUMPARARE
```

Script Output x | Query Result x

All Rows Fetched: 13 in 0.003 seconds

	ID_FAMILIE	ID_STRADA	NUMAR	DATA_MUTARE
1	5	16	14	10-MAY-80
2	10	11	2	10-APR-85
3	5	21	4	09-APR-85
4	20	21	5	29-AUG-03
5	15	21	4	15-FEB-95
6	15	1	1	20-JAN-97
7	10	1	1	14-JAN-01
8	15	11	2	10-JUN-00
9	20	21	5	19-JUL-05
10	20	1	1	20-MAR-02

Activate Windows
Go to Settings to activate Windows.

| Line 276 Column 28 | Insert | Modified | Windows: CI

project

Worksheet Query Builder

```

INSERT INTO CONTRACT_CUMPARARE
VALUES(3, 1, 20, '15-aug-2019', 5, 7000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(2, 5, 25, '14-sep-2023', 40, 800000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(3, 4, 20, '19-jul-2021', 14, 75189);
INSERT INTO CONTRACT_CUMPARARE
VALUES(2, 2, 25, '11-june-2021', 10, 10000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(1, 1, 5, '15-sep-2004', 5, 6510);
INSERT INTO CONTRACT_CUMPARARE
VALUES(3, 2, 20, '14-sep-2022', 10, 19000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(2, 1, 25, '19-oct-2022', 4, 7000);
INSERT INTO CONTRACT_CUMPARARE
VALUES(1, 3, 15, '18-nov-2022', 10, 19500);
INSERT INTO CONTRACT_CUMPARARE
VALUES(2, 3, 10, '20-aug-2001', NULL, 10000);

SELECT * FROM CONTRACT_CUMPARARE;

CREATE SEQUENCE id_prim

```

Script Output x Query Result x

SQL All Rows Fetched: 11 in 0.002 seconds

ID_PRIMARIE	ID_CUMPARATOR	ID_FAMILIE	DATA_CUMPARARE	TAXA	SUMA
1	1	1	5 15-SEP-90	15	120000
2	1	2	5 20-JAN-85	20	17000
3	3	1	20 15-AUG-19	5	7000
4	3	5	25 14-SEP-23	40	800000
5	2	4	20 19-JUL-21	14	75189
6	3	2	25 11-JUN-21	10	10000
7	1	1	5 15-SEP-04	5	6510
8	3	2	20 14-SEP-22	10	19000
9	2	1	25 19-OCT-22	4	7000
10	1	3	15 18-NOV-22	10	19500

Activate Windows
Go to Settings to activate Windows.

| Line 300 Column 34 | Insert | Modified | Windows: C

project

Worksheet Query Builder

```

VALUES(id_prim.nextval, 3, '28-march-2022',
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 4, '14-june-2007');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 2, '27-august-1992');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 1, '01-may-2011');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 2, '05-aug-2022');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 3, '29-sep-2022');

SELECT * FROM PRIMARIE_EVENTIMENT;

-----CERERI-----
--1. Afisati numele institutiilor de invatamant a persoanelor din familia care a avut cel mai scump contract de vanzare,
--precum si luna in care au inceput studiul
WITH max_vanzare AS( --bloc cerere cu with
SELECT max(suma)
FROM contract_cumparare)
SELECT f.num_familie, l.prenume, DECODE(nvl(i.id_forma,0),0,'Nu a terminat o forma de invatamant',i.num) AS "Nume institutie de
FROM familie f JOIN locuitor l ON f.id_familie = l.id_familie FULL OUTER JOIN ciclu_invatamant c ON c.id_locuitor = l.id_locuitor
FULL OUTER JOIN institutie_invatamant i ON i.id_forma = l.id_forma

```

Script Output x Query Result x

SQL All Rows Fetched: 12 in 0.002 seconds

ID_PRIMARIE	ID_EVENTIMENT	DATA_EVENTIMENT
1	1	1 12-JUL-98
2	2	5 06-NOV-06
3	3	3 20-SEP-93
4	4	2 03-MAY-14
5	5	1 18-JAN-02
6	1	3 09-OCT-22
7	2	5 28-MAR-22
8	3	4 14-JUN-07
9	4	2 27-AUG-92
10	5	1 01-MAY-11

Activate Windows
Go to Settings to activate Windows.

| Line 332 Column 34 | Insert | Modified | Windows: C

Cerința 12:

- 1. Afisati numele institutiilor de invatamant a persoanelor din familia care a avut cel mai scump contract de vanzare,
- precum si luna in care au inceput studiul

--elemente folosite: bloc de cerere cu with, functie pe data calendaristica, decode + nvl

```
WITH max_vanzare AS( --bloc cerere cu with
    SELECT max(suma)
    FROM contract_cumparare)
SELECT f.ume_familie, l.prenume, DECODE(nvl(i.id_forma,0),0,'Nu a terminat o
forma de invatamant',i.ume) AS "Nume institutie de Invatamant",
to_char(c.an_invatamant, 'Month') luna_incepere --folosire functie pe data
calendaristica + NVL si DECODE
FROM familie f JOIN locuitor l ON f.id_familie = l.id_familie FULL OUTER JOIN
ciclu_invatamant c ON c.id_locuitor = l.id_locuitor
    FULL OUTER JOIN institutie_invatamant i ON i.id_forma = c.id_forma
WHERE f.id_familie = (SELECT id_familie
    FROM contract_cumparare
    WHERE suma = (SELECT * FROM max_vanzare));
```

--2. Gasiti casa la care locuieste familia care a facut cei mai multi bani din vanzari de pe o strada care are

--prima litera din numele acestei familii drept nume

--with + functie group + filtru la group by + subcerere nesincronizata cu cel putin 3 tabele + subcerere nesincronizata in from

--+ functie pe sir de caractere

```
WITH max_vanzare_familie AS( --clausa with
    SELECT max(sum(suma)) --functie grup
    FROM contract_cumparare cc JOIN familie f ON cc.id_familie = f.id_familie
--group by
    GROUP BY cc.id_familie, f.ume_familie
    HAVING lower(f.ume_familie) = 'tupeiescu') --filtru la group by
SELECT f.id_familie, s.ume_strada, fc.numar, f.ume_familie
FROM familie f JOIN (SELECT id_familie FROM contract_cumparare GROUP BY
id_familie HAVING sum(suma) = (SELECT * FROM max_vanzare_familie)) aux on
aux.id_familie = f.id_familie --subcerere in FROM
    JOIN familie_casa fc ON fc.id_familie = f.id_familie JOIN strada s ON s.id_strada
= fc.id_strada
WHERE fc.data_mutare = (SELECT max(data_mutare) --functie grup + subcerere
nesincronizata cu cel putin 3 tabele
    FROM familie_casa fc2 JOIN strada s ON fc2.id_strada = s.id_strada
    WHERE fc2.id_familie = f.id_familie
    AND s.id_strada
    IN (SELECT ss.id_strada FROM strada ss WHERE
SUBSTR(lower(ss.ume_strada),1, 1) = SUBSTR(lower(f.ume_familie),1, 1)));
```

--3. Afisati numele(avand toate literele mari), prenumele(eliminand eventualele spatii), numarul de persoane din familie

```

--si un mesaj in functie de sumele contractelor de vanzare
-- pentru contractele trecut de anul 2000 si pentru familiile care au prefixul pop
--subcerere necorelata in clauza from + functie pe data calendaristica + grupare de
date + functie pe siruri de caractere
-- + case
WITH vanzari_familie_2000 AS(
    SELECT fam.ume_familie nume_familie, contr.suma suma,
nr_loc_fam.nr_membrii
    FROM (SELECT * FROM familie WHERE trim(lower(ume_familie)) like ('pop%'))
fam JOIN --2 functii pe siruri de caractere + subcereri nesincronizate in from
    (SELECT * FROM contract_cumparare WHERE to_char(data_cumparare, 'YYYY')
>= 2000) contr --functie pe data calendaristica
    ON fam.id_familie = contr.id_familie
    LEFT OUTER JOIN(SELECT id_familie, count(*) nr_membrii FROM locuitor
GROUP BY id_familie) nr_loc_fam -- grupare de date
    ON nr_loc_fam.id_familie = fam.id_familie
)
SELECT trim(ume_familie), suma, DECODE(nvl(nr_membrii,0),0,'Nu au fost gasiti
membrii', nr_membrii), --functie pe sir de caractere + NVL si DECODE
CASE
    WHEN suma < 3000 THEN 'Suma mica'
    WHEN (suma >= 3000 AND suma <= 10000) THEN 'Suma medie'
    WHEN suma > 10000 AND suma < 30000 THEN 'Suma mare' ---Utilizare CASE
    WHEN suma >= 30000 THEN 'Suma foarte mare'
    ELSE 'Alta suma'
END AS "Tip suma"
FROM vanzari_familie_2000;

```

```

--4. Afisati cate evenimente culturale au avut loc in anul 2022
--sunt luate in considerare doar cele care au capacitatea > minimul capacitatii din
randul tuturor evenimentelor conduse de firma cu id 5
--functie pe data calendaristica + functie group + grupari de date cu subcereri
nesincronizate cu 3 tabele + filtrare la nivel de grupuri
WITH an_eveniment AS(
    SELECT to_char(data_eveniment, 'YYYY') data_ev, id_eveniment id_ev
--functie pe data calendaristica
    FROM primarie_eveniment
)
SELECT sum(aux.nr) "Totalul evenimentelor"
FROM (SELECT count(ume_eveniment) nr FROM an_eveniment ae JOIN
(SELECT * FROM eveniment_cultural) ec ON ae.id_ev = ec.id_eveniment
GROUP BY data_ev, ec.ume_eveniment, ec.capacitate HAVING (data_ev) =
'2022' --grupari de date, subcereri nesincronizate cu 3 tabele

```

```
AND ec.capacitate > (SELECT MIN(capacitate) FROM eveniment_cultural  
GROUP BY id_firma HAVING id_firma = 5)) aux; --filtrare la nivel de grupuri + functie  
grup
```

--5. Afisati numele tuturor evenimentelor, impreuna cu capacitatea si primaria care le
a organizat, in ordine descrescatoare

--in functie de capacitate. Se vor afisa si primariile care nu au organizat evenimente

--ordonare + decode + nvl

```
SELECT DECODE(nvl(ec.capacitate, 0), 0, 'Primaria nu a organizat niciun  
eveniment', ec.nume_eveniment), DECODE(nvl(ec.capacitate, 0), 0, 0,  
ec.capacitate), p.nume --decode + nvl
```

```
FROM eveniment_cultural ec JOIN primarie_eveniment pe ON ec.id_eveniment =  
pe.id_eveniment RIGHT OUTER JOIN primarie p ON p.id_primarie = pe.id_primarie  
ORDER BY 2 desc; --ordonari
```

proect

Worksheet Query Builder

```

VALUES(id_prim.nextval, 1, '01-may-2011',
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 2, '05-aug-2022');
INSERT INTO PRIMARIE_EVENTIMENT
VALUES(id_prim.nextval, 3, '29-sep-2022');

-----CERERI-----
--1. Afisati numele institutiilor de invatamant a persoanelor din familia care a avut cel mai scump contract de vanzare,
--precum si luna in care au inceput studiul
WITH max_vanzare AS( --bloc cerere cu with
SELECT max(suma)
FROM contract_cumparare)
SELECT f.ume_familie, l.prenume, DECODE(nvl(i.id_forma,0),0,'Nu a terminat o forma de invatamant',i.ume) AS "Nume institutie de
FROM familie f JOIN locuitor l ON f.id_familie = l.id_familie FULL OUTER JOIN ciclu_invatamant c ON c.id_locuitor = l.id_locuitor
FULL OUTER JOIN institutie_invatamant i ON i.id_forma = c.id_forma
WHERE f.id_familie = (SELECT id_familie
FROM contract_cumparare
WHERE suma = (SELECT * FROM max_vanzare));

--2. Gasiti casa la care locuieste familia care a facut cei mai multi bani din vanzari de pe o strada care are
--prima litera din numele acestei familii drept nume
WITH max_vanzare_familie AS( --clausa with
SELECT max(sum(suma)) --functie grup
FROM contract_cumparare cc JOIN familie f ON cc.id_familie = f.id_familie --group by
GROUP BY cc.id_familie, f.ume_familie
HAVING lower(f.ume_familie) = 'tupeiescu' --filtru la group by
SELECT f.id_familie, s.ume_strada, fc.numar, f.ume_familie
FROM familie f JOIN (SELECT id_familie FROM contract_cumparare GROUP BY id_familie HAVING sum(suma) = (SELECT * FROM max_vanzare_fam
JOIN familie_casa fc ON fc.id_familie = f.id_familie JOIN strada s ON s.id_strada = fc.id_strada
WHERE fc.data_mutare = (SELECT max(data_mutare) --functie grup + subcerere nesincronizata cu cel putin 3 tabele
FROM familie_casa fc2 JOIN strada s ON fc2.id_strada = s.id_strada
WHERE fc2.id_familie = f.id_familie
AND s.id_strada
IN (SELECT ss.id_strada FROM strada ss WHERE SUBSTR(lower(ss.ume_strada),1, 1) = SUBSTR(lower(f.ume_familie),1, 1)))
--3. Afisati numele (avand toate literele mari), prenumele (eliminand eventualele spatii), numarul de persoane din familie

```

Script Output x Query Result x

SQL All Rows Fetched: 2 in 0.004 seconds

NUME_FAMILIE	PRENUME	Nume institutie de Invatamant	LUNA_INCEPERE
1	Tupeiescu Costache	Colegiul National "Gheorghe Lazar"	May
2	Tupeiescu Mihaela	Nu a terminat o forma de invatamant (null)	

| Line 342 Column 44 | Insert | Modified | Windows: C

proect

Worksheet Query Builder

```

FULL OUTER JOIN institutie_invatamant i ON i.id_forma = c.id_forma
WHERE f.id_familie = (SELECT id_familie
FROM contract_cumparare
WHERE suma = (SELECT * FROM max_vanzare));

--2. Gasiti casa la care locuieste familia care a facut cei mai multi bani din vanzari de pe o strada care are
--prima litera din numele acestei familii drept nume
WITH max_vanzare_familie AS( --clausa with
SELECT max(sum(suma)) --functie grup
FROM contract_cumparare cc JOIN familie f ON cc.id_familie = f.id_familie --group by
GROUP BY cc.id_familie, f.ume_familie
HAVING lower(f.ume_familie) = 'tupeiescu' --filtru la group by
SELECT f.id_familie, s.ume_strada, fc.numar, f.ume_familie
FROM familie f JOIN (SELECT id_familie FROM contract_cumparare GROUP BY id_familie HAVING sum(suma) = (SELECT * FROM max_vanzare_fam
JOIN familie_casa fc ON fc.id_familie = f.id_familie JOIN strada s ON s.id_strada = fc.id_strada
WHERE fc.data_mutare = (SELECT max(data_mutare) --functie grup + subcerere nesincronizata cu cel putin 3 tabele
FROM familie_casa fc2 JOIN strada s ON fc2.id_strada = s.id_strada
WHERE fc2.id_familie = f.id_familie
AND s.id_strada
IN (SELECT ss.id_strada FROM strada ss WHERE SUBSTR(lower(ss.ume_strada),1, 1) = SUBSTR(lower(f.ume_familie),1, 1)))
--3. Afisati numele (avand toate literele mari), prenumele (eliminand eventualele spatii), numarul de persoane din familie

```

Script Output x Query Result x

SQL All Rows Fetched: 1 in 0.019 seconds

ID_FAMILIE	NUME_STRADA	NUMAR	NUME_FAMILIE
1	25 Tineretului	14	Tupeiescu

Activate Windows
Go to Settings to activate Windows.

| Line 355 Column 83 | Insert | Modified | Windows: C

Worksheet Query Builder

```
IN (SELECT ss.id_strada FROM strada ss WHERE SUBSTR(lower(ss.num_e_strada),1,1) = SUBSTR(lower(f.num_e_familie),1,1))

--3. Afisati numele (avand toate literele mari), prenumele (eliminand eventualele spatii), numarul de persoane din familie
--si un mesaj in functie de sumele contractelor de vanzare
-- pentru contractele trecut de anul 2000 si pentru familiile care au prefixul pop

WITH vanzari_familie_2000 AS (
    SELECT fam.num_e_familie num_e_familie, contr.suma suma, nr_loc_fam.nr_membrii nr_membrii
    FROM (SELECT * FROM familie WHERE trim(lower(num_e_familie)) like ('pop%')) fam JOIN --2 functii pe siruri de caractere + subcere
    (SELECT * FROM contract_cumparare WHERE to_char(data_cumparare, 'YYYY') >= 2000) contr --functie pe data calendaristica
    ON fam.id_familie = contr.id_familie
    LEFT OUTER JOIN (SELECT id_familie, count(*) nr_membrii FROM locuitor GROUP BY id_familie) nr_loc_fam -- grupare de date
    ON nr_loc_fam.id_familie = fam.id_familie
)
SELECT trim(num_e_familie), suma, DECODE(nvl(nr_membrii,0),0,'Nu au fost gasiti membrii', nr_membrii), --functie pe sir de caractere
CASE
    WHEN suma < 3000 THEN 'Suma mica'
    WHEN (suma >= 3000 AND suma <= 10000) THEN 'Suma medie'
    WHEN suma > 10000 AND suma < 30000 THEN 'Suma mare' ---Utilizare CASE
    WHEN suma >= 30000 THEN 'Suma foarte mare'
    ELSE 'Alta suma'
END AS "Tip suma"
FROM vanzari_familie_2000;
```

Script Output x Query Result x

SQL All Rows Fetched: 2 in 0.004 seconds

	1	2	3	4	5
	TRIM(NUM_E_FAMILIE)	SUMA	DECODE(NVL(NR_MEMBRII,0),0,'NU AU FOST GASITI MEMBRII',NR_MEMBRII)	TIP SUMA	
1	Popescu	6510	3	Suma medie	
2	Popa	19500	Nu au fost gasiti membrii	Suma mare	

Activate Windows
Go to Settings to activate Windows.

Line 369 Column 39 | Insert | Modified | Windows: C

```

--4. Afisati cate evenimente culturale au avut loc in anul 2022
--sunt luate in considerare doar cele care au capacitatea > minimul capacitatii din randul tuturor evenimentelor conduse de firma cu
WITH an_eveniment AS(
    SELECT to_char(data_eveniment, 'YYYY') data_ev, id_eveniment id_ev --functie pe data calendaristica
    FROM primarie_eveniment
)
SELECT sum(aux.nr) "Totalul evenimentelor"
FROM (SELECT count(nume_eveniment) nr FROM an_eveniment ae JOIN (SELECT * FROM eveniment_cultural) ec ON ae.id_ev = ec.id_eveniment
GROUP BY data_ev, ec.num_eveniment, ec.capacitate HAVING (data_ev) = '2022' --grupari de date, subcereri nesincronizate cu
AND ec.capacitate > (SELECT MIN(capacitate) FROM eveniment_cultural GROUP BY id_firma HAVING id_firma = 5)) aux; --filtrare

--5. Afisati numele tuturor evenimentelor, impreuna cu capacitatea si primaria care le a organizat, in ordine descrescatoare
--in functie de capacitate. Se vor afisa si primariile care nu au organizat evenimente
SELECT DECODE(nvl(ec.capacitate, 0), 0, 'Primaria nu a organizat niciun eveniment', ec.num_eveniment), DECODE(nvl(ec.capacitate, 0)

```

```

--sunt luate in considerare doar cele care au capacitatea > minimul capacitatii din randul tuturor evenimentelor conduse de firma cu
WITH an_eveniment AS(
    SELECT to_char(data_eveniment, 'YYYY') data_ev, id_eveniment id_ev --functie pe data calendaristica
    FROM primarie_eveniment
)
SELECT sum(aux.nr) "Totalul evenimentelor"
FROM (SELECT count(nume_eveniment) nr FROM an_eveniment ae JOIN (SELECT * FROM eveniment_cultural) ec ON ae.id_ev = ec.id_eveniment
GROUP BY data_ev, ec.num_eveniment, ec.capacitate HAVING (data_ev) = '2022' --grupari de date, subcereri nesincronizate cu
AND ec.capacitate > (SELECT MIN(capacitate) FROM eveniment_cultural GROUP BY id_firma HAVING id_firma = 5)) aux; --filtrare

--5. Afisati numele tuturor evenimentelor, impreuna cu capacitatea si primaria care le a organizat, in ordine descrescatoare
--in functie de capacitate. Se vor afisa si primariile care nu au organizat evenimente
SELECT DECODE(nvl(ec.capacitate, 0), 0, 'Primaria nu a organizat niciun eveniment', ec.num_eveniment), DECODE(nvl(ec.capacitate, 0)
FROM eveniment_cultural ec JOIN primarie_eveniment pe ON ec.id_eveniment = pe.id_eveniment RIGHT OUTER JOIN primarie p ON p.id_prima
ORDER BY 2 desc; --ordonari:

--13.
--1. Cresteti capacitatea tuturor evenimentelor cu id > 3 cu 150 de locuri fata de media tuturor evenimentelor cu id < 3
UPDATE eveniment_cultural
SET capacitate = capacitate + 150
WHERE id_eveniment IN (SELECT id_eveniment FROM eveniment_cultural WHERE id_eveniment > 3);

```

Cerința 13:

--1. Cresteti capacitatea tuturor evenimentelor cu id > 3 cu 150 de locuri fata de media tuturor evenimentelor cu id < 3

UPDATE eveniment_cultural

SET capacitate = capacitate + 150

```
WHERE id_eveniment IN (SELECT id_eveniment FROM eveniment_cultural
WHERE id_eveniment > 3);
```

--2. Cresteti taxa cu 2 procente tuturor contractelor pentru familiile ce au '%escu%' in nume;

```
UPDATE contract_cumparare
```

```
SET taxa = taxa + 2
```

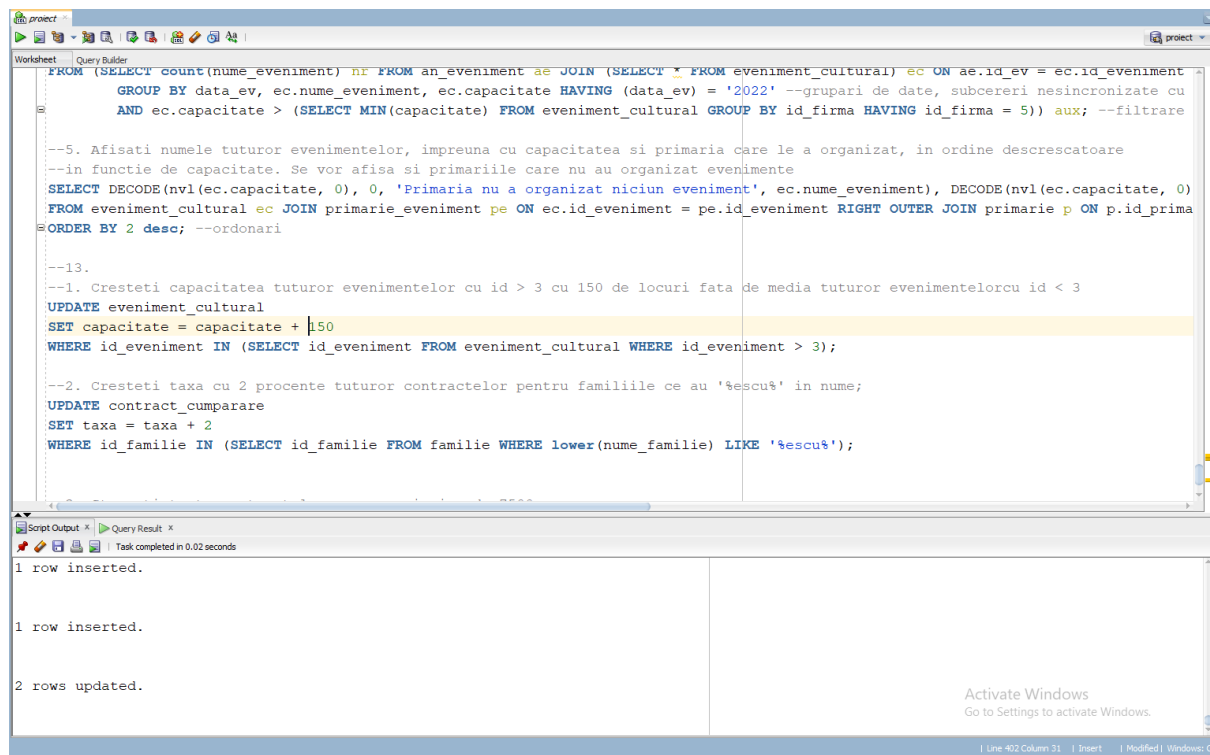
```
WHERE id_familie IN (SELECT id_familie FROM familie WHERE
lower(nume_familie) LIKE '%escu%');
```

--3. Stergeti toate contractele cu suma mai mica de 7500

```
DELETE
```

```
FROM contract_cumparare
```

```
WHERE (id_primarie, id_cumparator, id_familie) IN (SELECT id_primarie,
id_cumparator, id_familie FROM contract_cumparare WHERE suma <= 7500);
```



The screenshot shows a SQL IDE window titled 'proiect'. The main editor displays a script with several SQL queries and comments. The queries include a complex JOIN, an UPDATE on 'eveniment_cultural', and an UPDATE on 'contract_cumparare'. The bottom panel shows the 'Script Output' with the following results:

```
1 row inserted.

1 row inserted.

2 rows updated.
```

At the bottom right, there is a watermark that says 'Activate Windows Go to Settings to activate Windows.'

```
SELECT DECODE(nvl(ec.capacitate, 0), 0, 'Primaria nu a organizat niciun eveniment', ec.num_eveniment), DECODE(nvl(ec.capacitate, 0)
FROM eveniment_cultural ec JOIN primarie_eveniment pe ON ec.id_eveniment = pe.id_eveniment RIGHT OUTER JOIN primarie p ON p.id_prima
= ORDER BY 2 desc; --ordonari

--13.
--1. Cresteti capacitatea tuturor evenimentelor cu id > 3 cu 150 de locuri fata de media tuturor evenimentelor cu id < 3
UPDATE eveniment_cultural
SET capacitate = capacitate + 150
WHERE id_eveniment IN (SELECT id_eveniment FROM eveniment_cultural WHERE id_eveniment > 3);

--2. Cresteti taxa cu 2 procente tuturor contractelor pentru familiile ce au '%escu%' in nume;
UPDATE contract_cumparare
SET taxa = taxa + 2
WHERE id_familie IN (SELECT id_familie FROM familie WHERE lower(nume_familie) LIKE '%escu%');

--3. Stergeti toate contractele cu suma mai mica de 7500
DELETE
FROM contract_cumparare
WHERE (id_primarie, id_cumparator, id_familie) IN (SELECT id_primarie, id_cumparator, id_familie FROM contract_cumparare WHERE suma
```

Script Output x Query Result x

Task completed in 0.025 seconds

1 row inserted.

2 rows updated.

10 rows updated.

Activate Windows
Go to Settings to activate Windows.

Line 407 Column 15 | Insert | Modified | Windows: G

```
SELECT DECODE(nvl(ec.capacitate, 0), 0, 'Primaria nu a organizat niciun eveniment', ec.num_eveniment), DECODE(nvl(ec.capacitate, 0)
FROM eveniment_cultural ec JOIN primarie_eveniment pe ON ec.id_eveniment = pe.id_eveniment RIGHT OUTER JOIN primarie p ON p.id_prima
= ORDER BY 2 desc; --ordonari

--13.
--1. Cresteti capacitatea tuturor evenimentelor cu id > 3 cu 150 de locuri fata de media tuturor evenimentelor cu id < 3
UPDATE eveniment_cultural
SET capacitate = capacitate + 150
WHERE id_eveniment IN (SELECT id_eveniment FROM eveniment_cultural WHERE id_eveniment > 3);

--2. Cresteti taxa cu 2 procente tuturor contractelor pentru familiile ce au '%escu%' in nume;
UPDATE contract_cumparare
SET taxa = taxa + 2
WHERE id_familie IN (SELECT id_familie FROM familie WHERE lower(nume_familie) LIKE '%escu%');

--3. Stergeti toate contractele cu suma mai mica de 7500
DELETE
FROM contract_cumparare
WHERE (id_primarie, id_cumparator, id_familie) IN (SELECT id_primarie, id_cumparator, id_familie FROM contract_cumparare WHERE suma
```

Script Output x Query Result x

Task completed in 0.016 seconds

2 rows updated.

10 rows updated.

4 rows deleted.

Activate Windows
Go to Settings to activate Windows.

Line 412 Column 7 | Insert | Modified | Windows: G

Cerința 15:

--Cerere cu outer join

--afisati istoricul tuturor caselor la care au stat familii, inclusiv cele la care nu au stat nimeni, precum si primariile acestora,

--iar daca nu au primarie afisati 'In constructie'. De asemenea, afisati institutiile de invatamant prin care au locuitorii

--daca nu au trecut afisati 'Nu are studii la acest nivel'

```
SELECT UNIQUE DECODE(nvl(id_forma, 0), 0, 'Nu are studii la acest nivel',
id_forma) id_studii,
      fc.id_familie familie, c.id_strada strada, c.numar numar,
DECODE(nvl(p.id_primarie, 0), 0, 'In constructie', p.numa) primarie
FROM CICLU_INVATAMANT ci RIGHT OUTER JOIN LOCUITOR l ON ci.id_locuitor
= id_familie
      JOIN FAMILIE_CASA fc ON l.id_familie = fc.id_familie
      RIGHT OUTER JOIN CASA c ON (fc.id_strada = c.id_strada AND fc.numar =
c.numar)
      LEFT OUTER JOIN PRIMARIE p ON c.id_strada = p.id_strada;
```

--Cerere cu division

--Afisati primariile care au organizat toate tipurile de evenimente

```
SELECT p.numa
FROM (SELECT DISTINCT id_eveniment, id_primarie FROM
PRIMARIE_EVENTIMENT) pe JOIN PRIMARIE p ON pe.id_primarie = p.id_primarie
GROUP BY pe.id_primarie, p.numa
HAVING COUNT(pe.id_eveniment) = (SELECT COUNT(*) FROM
EVENTIMENT_CULTURAL);
```

--Cerere cu analiza top-n

--afisati numele familiilor cu primele 5 cele mai scumpe contracte, precum si primaria de care apartin

```
WITH sume_sortate AS(
      SELECT * FROM contract_cumparare
      ORDER BY SUMA desc)
SELECT f.numa_familie, p.numa, suma
FROM sume_sortate ss JOIN familie f on ss.id_familie = f.id_familie JOIN primarie p
on f.id_primarie = p.id_primarie
WHERE ROWNUM <= 5;
```

proiect1.sql Welcome Page proiect

Worksheet Query Builder

```
--15
--Cerere cu outer join
--afisati istoricul tuturor caselor la care au stat familii, inclusiv cele la care nu au stat nimeni, precum si primariile acestora,
--iar daca nu au primarie afisati 'In constructie'. De asemenea, afisati institutiile de invatamant prin care au locuitorii
--daca nu au trecut afisati 'Nu are studii la acest nivel'
SELECT UNIQUE DECODE(nvl(id_forma, 0), 0, 'Nu are studii la acest nivel', id_forma) id_studii,
       fc.id_familie familie, c.id_strada strada, c.numar numar, DECODE(nvl(p.id_primarie, 0), 0, 'In constructie', p.nume) primarie
FROM CICLU_INVATAMANT ci RIGHT OUTER JOIN LOCUITOR l ON ci.id_locuitor = id_familie
JOIN FAMILIE_CASA fc ON l.id_familie = fc.id_familie
RIGHT OUTER JOIN CASA c ON (fc.id_strada = c.id_strada AND fc.numar = c.numar)
LEFT OUTER JOIN PRIMARIE p ON c.id_strada = p.id_strada;

--Cerere cu division
--Afisati primariile care au organizat toate tipurile de evenimente
SELECT p.nume
```

Query Result

All Rows Fetched: 10 in 0.032 seconds

ID_STUDII	FAMILIE	STRADA	NUMAR	PRIMARIE
1 Nu are studii la acest nivel	(null)	11	2	Primaria Crinului
2 5	5	16	14	Primaria Tineretului
3 3	5	21	4	Primaria Eroilor
4 Nu are studii la acest nivel	20	21	5	Primaria Eroilor
5 Nu are studii la acest nivel	(null)	64	13	In constructie
6 Nu are studii la acest nivel	25	16	14	Primaria Tineretului
7 5	5	21	4	Primaria Eroilor
8 Nu are studii la acest nivel	30	16	25	Primaria Tineretului
9 3	5	16	14	Primaria Tineretului
10 Nu are studii la acest nivel	20	1	1	Primaria Teilor

Activate Windows
Go to Settings to activate Windows.

Line 11 Column 59 | Insert | Modified | Windows: O

proiect1.sql Welcome Page proiect

Worksheet Query Builder

```
fc.id_familie familie, c.id_strada strada, c.numar numar, DECODE(nvl(p.id_primarie, 0), 0, 'In constructie', p.nume) primarie
FROM CICLU_INVATAMANT ci RIGHT OUTER JOIN LOCUITOR l ON ci.id_locuitor = id_familie
JOIN FAMILIE_CASA fc ON l.id_familie = fc.id_familie
RIGHT OUTER JOIN CASA c ON (fc.id_strada = c.id_strada AND fc.numar = c.numar)
LEFT OUTER JOIN PRIMARIE p ON c.id_strada = p.id_strada;

--Cerere cu division
--Afisati primariile care au organizat toate tipurile de evenimente
SELECT p.nume
FROM (SELECT DISTINCT id_eveniment, id_primarie FROM PRIMARIE_EVENTIMENT) pe JOIN PRIMARIE p ON pe.id_primarie = p.id_primarie
GROUP BY pe.id_primarie, p.nume
HAVING COUNT(pe.id_eveniment) = (SELECT COUNT(*) FROM EVENIMENT_CULTURAL);

--Cerere cu analiza top-n
--afisati numele familiilor cu primele 5 cele mai scumpe contracte, precum si primaria de care apartin
```

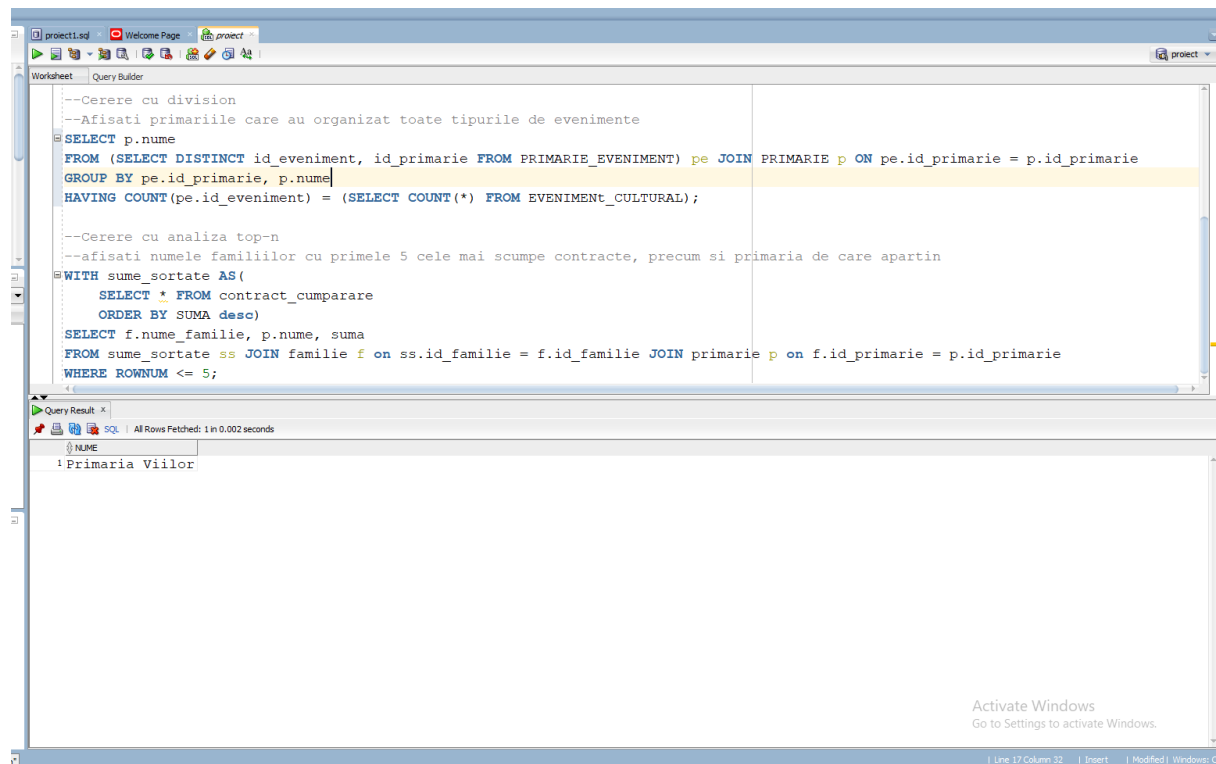
Query Result

All Rows Fetched: 1 in 0.002 seconds

NUME
Primaria Viilor

Activate Windows
Go to Settings to activate Windows.

Line 17 Column 32 | Insert | Modified | Windows: O



Cerința 16:

Cererea înainte de optimizare:

Expresie algebrică:

$$\Pi_{sum(nr)}(\sigma_{data_eveniment = '2022, capacitate > \sigma_{min}(\Pi_{capac}(\sigma_{id_firma = 5(eventiment_cultural)}(JOIN(\Pi_{count(num_eveniment)}(an_eveniment), eveniment_cultural), unde an_eveniment = \Pi_{data_eveniment, id_eveniment}(primarie_eveniment)$$

Arbore algebric:



Cererea în limbaj SQL: (Cerința de la exercițiul 12, exemplul 4)

```

WITH an_eveniment AS(
SELECT to_char(data_eveniment, 'YYYY') data_ev, id_eveniment id_ev
FROM primarie_eveniment
)
SELECT sum(aux.nr) "Totalul evenimentelor"
FROM (SELECT count(ume_eveniment) nr FROM an_eveniment ae JOIN
(SELECT * FROM eveniment_cultural) ec ON ae.id_ev = ec.id_eveniment
GROUP BY data_ev, ec.ume_eveniment, ec.capacitate HAVING (data_ev) =
'2022'
AND ec.capacitate > (SELECT MIN(capacitate) FROM eveniment_cultural
GROUP BY id_firma HAVING id_firma = 5)) aux;

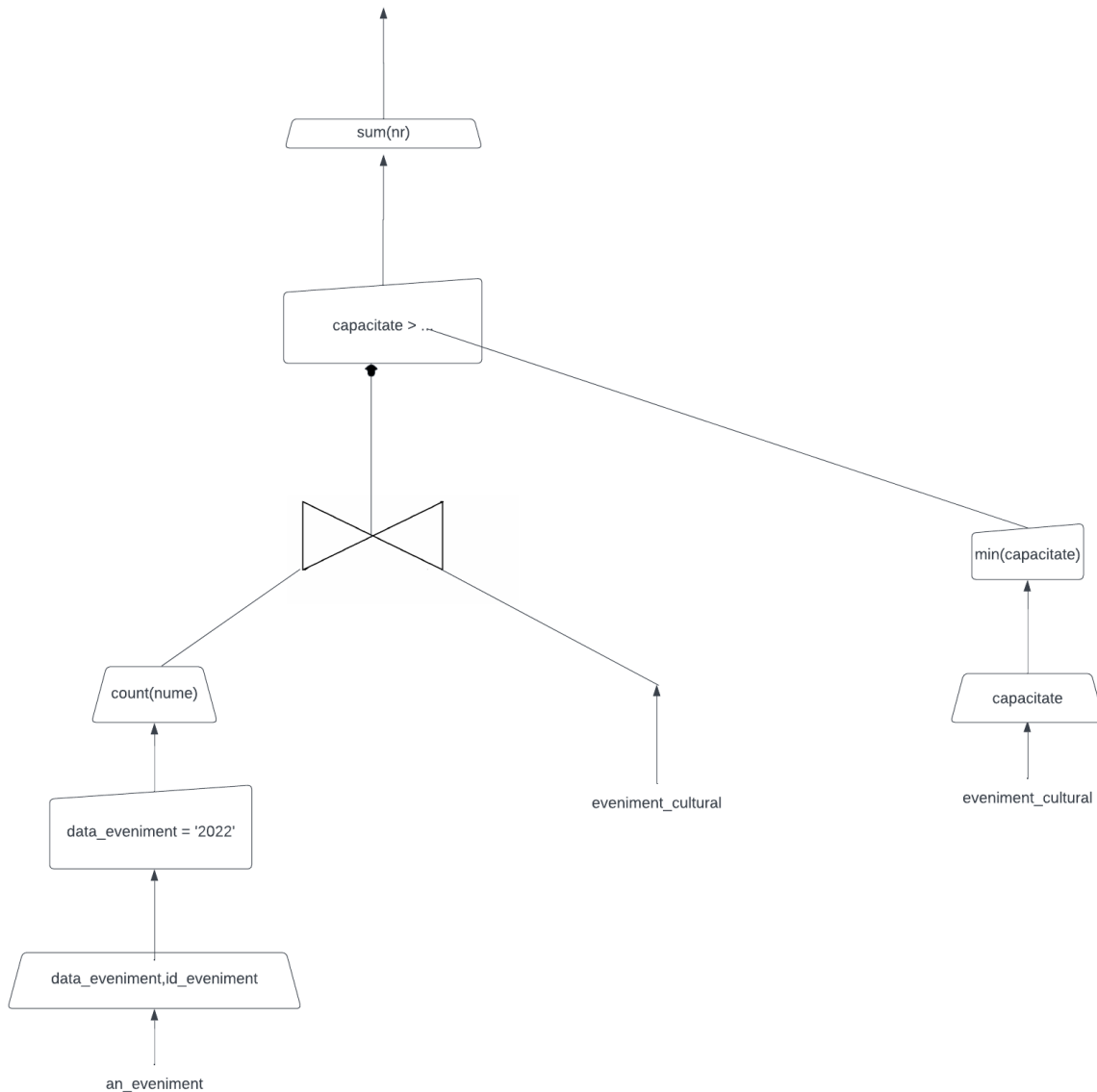
```

Cererea după optimizare:

Expresie algebrică:

$\Pi_{suma(nr)}(\Pi_{capac > (\sigma_{\min(\Pi_{capac}(\sigma_{id_firma=5}(eveniment_cultural)))} JOIN(\Pi_{count(ume_eveniment)}(an_eveniment), eveniment_cultural))$

Arbore algebric:



Cererea în limbaj SQL:

```
WITH an_eveniment AS(  
    SELECT to_char(data_eveniment, 'YYYY') data_ev, id_eveniment id_ev  
    --functie pe data calendaristica  
    FROM primarie_eveniment  
    WHERE to_char(data_eveniment, 'YYYY') = '2022'
```

```

)
SELECT sum(aux.nr) "Totalul evenimentelor"
FROM (SELECT count(ume_eveniment) nr FROM an_eveniment ae JOIN
eveniment_cultural ec ON ae.id_ev = ec.id_eveniment
GROUP BY data_ev, ec.ume_eveniment, ec.capacitate --grupari de date,
subcereri nesincronizate cu 3 tabele
HAVING ec.capacitate > (SELECT MIN(capacitate) FROM eveniment_cultural
GROUP BY id_firma HAVING id_firma = 5)) aux;

```

Cerința 17:

BCNF:

Incalcarea formei normale BCNF se produce atunci când dintr-un atribut non-cheie se poate deduce un atribut cheie. Un exemplu din diagramă ce contravine acestei convenții ar putea fi, de exemplu, ca în tabela FAMILIE, id-ul familiei să depindă de nume_familie. Pentru a rezolva această problemă am putea crea un nou tabel în care numele de familie este cheie primară, iar din el s-ar deduce id_familie.

FN4:

Forma normală 4 reprezintă apariția unor perechi de cheie-atribut duplicate. De exemplu, acest lucru s-ar întâmpla în diagrama proiectată dacă, am avea în tabela PRIMARIE atributul facilitati (Ex: (31, Primaria X, 3, 'Clădire iluminată'), (31, Primaria X, 3, 'Spatiu verde')). Pentru a rezolva această problemă cel mai bine este să creăm un tabel nou, intitulat, de exemplu, Primarie_facilitati în care să notăm pentru fiecare primărie facilitățile (Ex: (31, 'Clădire Iluminată'), (31, 'Spatiu verde'))

FN5:

Forma normală 5 spune că dacă un tabel, în urma descompunerii în alte tabele mai mici își pierde din informații, atunci acesta nu ar trebui descompus.

Un exemplu de astfel de tabel este CONTRACT_CUMPARARE. Dacă acesta ar fi descompus în alte două tabele de exemplu PRIMARIE_CUMPARATOR și CUMPARATOR_FAMILIE s-ar pierde informația de primărie care a asistat la cumpărător în tranzacția cu familia.