Universitatea din București

Facultatea de Matematică și Informatică

Specializarea Informatică

GESTIUNEA ACTIVITATILOR UNEI GRADINITA STOCHEAZA INFORMATII SI DESPRE INCADRAREA COPIILOR PE GRUPE DESPRE SALILE IN CARE SE TIN LECTIILE.

Proiect realizat de:

Allanur Yakubov

Grupa 132

# **1.** Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

# Descrierea modelului real și a utilității acestuia:

Proiectul își propune să modernizeze modul în care grădinițele gestionează datele, cu accent pe siguranța copiilor și îmbunătățirea comunicării între părinți și educatori. Părinții pot vedea cu ușurință ce fac copiii lor în fiecare zi, inclusiv programul lor, clasele și cine sunt profesorii lor. Educatorii pot accesa detalii importante despre fiecare copil, cum ar fi informațiile de contact și cum să îi contacteze pe părinți.

Fiecare părinte are adresa, numărul de telefon și adresa de e-mail. Acest lucru facilitează partajarea actualizărilor pentru educatori și menținerea legăturii cu familiile. Sistemul nostru este conceput pentru a simplifica sarcinile pentru educatori, oferindu-le mai mult timp pentru a se concentra pe ajutarea copiilor să învețe și să crească.

Regulile de funcționare ale modelului:

Fiecare copil are parent(tata sau mama.una dintre ei.Nu avem nevoie ambele).

Avem singur adres pentru fiecare parent. Fiecare copil dintre un grup.

Copii face lectia cu angajat diferit .

Angajat preda de la diferit subictele.

2.Modelul de date respectă anumite restricţii de funcţionare**:**

1.Un parinte are cel putin un copil, un copi are un parent.

2.Un parinte locieste in singur adresă, la locatia se afla o singur parinte.

3.Un copil face parte dintr-un grup (one child belongs to one group) iar un grup are cel putin 1 copi.( but group has at least one kid.)

5.Multe angajati preda lectia de la multe subiect la multe grupuri. (one or many).

**3.** Descrierea entităților, incluzând precizarea cheii primare

Entitățile foloti în model sunt: Parents, Adresă, Copi, Grupa, Angajati, Subiect, Lectia

## **1.Parinte** = Entitatea părinte descrie părintele, numele acestuia, adresa de e-mail, numărul de telefon. Aceștia sunt primii oameni cu care profesorii ar trebui să țină legătura în legătură cu situația copiilor. Cheia primara a acestei entități e id\_parinte.

**2.Adresă** = Această tabelă oferă mai multe informații despre locația părintelui, respectiv orașul în care locuiește, strada, codul poștal și apartamentul. Cheia primara a acestei entități e id\_adress.

**3.Copi** = Aceasta este entitatea principală din modelul nostru. Această entitate oferă informații despre copii, cum ar fi numele, prenumele și data nașterii. Cheia primara a acestei entități e id\_copi.

**4.Grupa** = Această entitate e esentială in modelul nostru de date gradiniță. Reprezintă numele grupului de copii, oferind o denumire distinctivă pentru fiecare grupă din grădiniță si vârstă al copiilor din grupă, reprezintă capacitatea maximă a grupului, adică numărul maxim de copii pe care grupa îi poate primi.Cheia primara a acestei entități e id\_grupa.

**5. Angajat** = Această structură simplă a tabelului "angajati" permite înregistrarea eficientă a datelor despre personalul grădiniței, facilitând managementul resurselor umane și comunicarea internă. Cheia primara a acestei entități e id\_angajat.

**6. Subiect** = Această tabelă, numită "subiect" (subject), servește ca un depozit pentru diferitele materii predate în cadrul curriculumului grădiniței. Fiecare materie este identificată în mod unic printr-un ID de materie și este însoțită de numele său. Cheia primara a acestei entități e id\_subiect.

**7.Lectia** = Această structură de tabel este associative si e folosită pentru a gestiona informațiile despre lecții în cadrul grădiniței. Fiecare lecție este identificată printr-un ID unic și este asociată cu o anumită grupă, subiect, angajat responsabil în care are loc. Acest tabel ajută la organizarea și urmărirea activităților de predare și învățare din cadrul grădiniței. Cheia primară compusă este formată din id\_lectia,id\_profesor, id\_subiect, id\_grupa.

**4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.**

**1.Parinte\_locuește\_în\_adresa** = relație care leagă entitățile parent și Adressa, reflectând legătura dintre acestea( adressa are o anumita parent). **Ea are cardinalitatea minimă 1:1**( Parent trebuie sa aiba o singur adressa, si adressa apartine unui singur părinte.) si **cardinalitatea maxima 1:1.**

**2.Parente\_are\_copi** = Relație care leagă entitățile Parent și Copi, , reflectând legătura dintre acestea(Care e copilul lui parent, si cine e parent copilul ) **Ea cardinalitatea minimă 1:1 si cardinalitatea maximă 1:n.**

**3.Copi\_e\_în\_grupa** = Relație de tip one-to-many care leagă entitățile Copi și Grupa.Relație are **cardinalitatea minimă 1:1 si cardinalitatea maximă 1:n.**

**Angajat\_preda\_subiect\_la\_grupa** =Relație de tip *many\_to\_many* care leagă Angajat, Subiect, Grupa reflectând cine a preda un subiect, la ce grupa. Denumirea acestei e ***Lecție.***

**5.Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.**

1.Atributele entității Parinte:

Id\_parinte : variabilă de tip număr, de lungime maximă 5,care reprezintă id\_ul(codul) parent; E cheia primară a entități și are valoare obligatorie și unică.

Nume: variablă de tip caracter, de lungime maxima 30,care reprezintă nume parintei;valoare acestui atribut trebuie sa fie diferite de NULL.

Prenume : variablă de tip caracter, de lungime maxima 30, care reprezintă prenume parintei;valoare acestui atribut trebuie sa fie diferite de NULL.

Mail : variabilă de tip caracter, de lungime maximă 50, care reprezintă adresa de mail a unei parinte;

Telefon :variabilă de tip numar, de lungime 10, care reprezintă numărul de telefon al parinte; valoare acestui atribut trebuie sa fie diferite de NULL.

Id\_adresă : variable de tip numar,de lungime 5,care reprezintă idul adressei.Atributul trebuie sa corespundă la o valoare a cheii primare din tabelul Adress.

2.Atributele entității Adresă:

Id\_adress:variabilă de tip întreg, de lungime maximă 5,care reprezintă id\_ul(codul)adresă; E cheia primară a entități și are valoare obligatorie și unică

Oraș: variablă de tip caracter, de lungime maxima 30, care reprezintă nume orașlui;valoare acestui atribut trebuie sa fie diferite de NULL.

Strada: variablă de tip caracter, de lungime maxima 30, care reprezintă nume stradei;valoare acestui atribut trebuie sa fie diferite de NULL.

Zip\_code: variabilă de tip numar, de lungime maximă 6, care reprezintă codul postal al unei adresă

Appartment:variablă de tip caracter, de lungime maximă 30,care reprezintă nume blocului.

3. Atributele entității Copi:

Id\_copi: variabilă de tip number, de lungime maximă 5,care reprezintă id\_ul(codul)copil; E cheia primară a entități și are valoare obligatorie și unică.

Nume: variablă de tip caracter, de lungime maxima 30,care reprezintă nume copilui;valoare acestui atribut trebuie sa fie diferite de NULL.

Prenume: variablă de tip caracter, de lungime maxima 30,care reprezintă prenume copilui;valoare acestui atribut trebuie sa fie diferite de NULL.

Data\_naștere: variabilă de tip dată calendaristică, care reprezintă data de naștere a copilui

Id\_grupa: variabilă de tip number, de lungime maximă 5, care reprezintă codul grupei din care face parte grupa. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul\_grupa.

Id\_parinte: variabilă de tip number, de lungime maximă 5, care reprezintă codul parintei din care face parte parinte. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul\_Parinte.

4. Atributele entității Grupa:

Id\_grupa: variabilă de tip number, de lungime maximă 5, care reprezintă id\_ul(codul)grupei; E cheia primară a entități și are valoare obligatorie și unică.

Nume\_grupa: variabilă de tip caracter, de lungime maximă 30, care reprezintă numele de grupei; valoarea acestui atribut trebuie să fie diferită de NULL si UNICĂ deoarece este o informație esențială pentru funcționarea corectă;

vârsta\_copi: variabilă de tip numar, de lungime maximă 2,care reprezintă vârsta copiilor pentru grupa(de exemplu avem grupa pentru vârsta 6,si pentru vârsta 3și altele) ; valoarea acestui atribut trebuie să fie diferită de NULL

Capacitate: variabilă de tip numar, de lungime maximă 2, care reprezintă capacitatea grupei. ; valoarea acestui atribut trebuie să fie diferită de NULL.

5. Atributele entității Angajat:

Id\_angajat: variabilă de tip numar , de lungime maximă 5, care reprezintă id\_ul(codul)activitatei; E cheia primară a entități și are valoare obligatorie și unică.

Nume: variablă de tip caracter, de lungime maxima 30,care reprezintă nume angajatului;valoare acestui atribut trebuie sa fie diferite de NULL

Prenume: variablă de tip caracter, de lungime maxima 30,care reprezintă prenume angajatlui;valoare acestui atribut trebuie sa fie diferite de NULL

Mail: : variabilă de tip caracter, de lungime maximă 50, care reprezintă adresa de mail a unei angajatului; valoare acestui atribut trebuie sa fie diferite de NULL

Telefon: variabilă de tip numar, de lungime 10, care reprezintă numărul de telefon al anagjatlui; valoare acestui atribut trebuie sa fie diferite de NULL

Poziția:variablă de tip caracter, de lungime 30,care reprezintă poziția angajatului, valoare acestui atribut trebuie sa fie diferite de NULL

6. Atributele entității Subiect:

Id\_subiect: variabilă de tip numar, de lungime maximă 5, care reprezintă id\_ul(codul) subiectei; E cheia primară a entități și are valoare obligatorie și unică

Nume: : variablă de tip caracter, de lungime maxima 30, care reprezintă nume subiectei; valoare acestui atribut trebuie sa fie diferite de NULL

7. Relație Angajat\_preda\_subiect\_la\_grupa\_(Entitatea Lecție)are ca atribute:

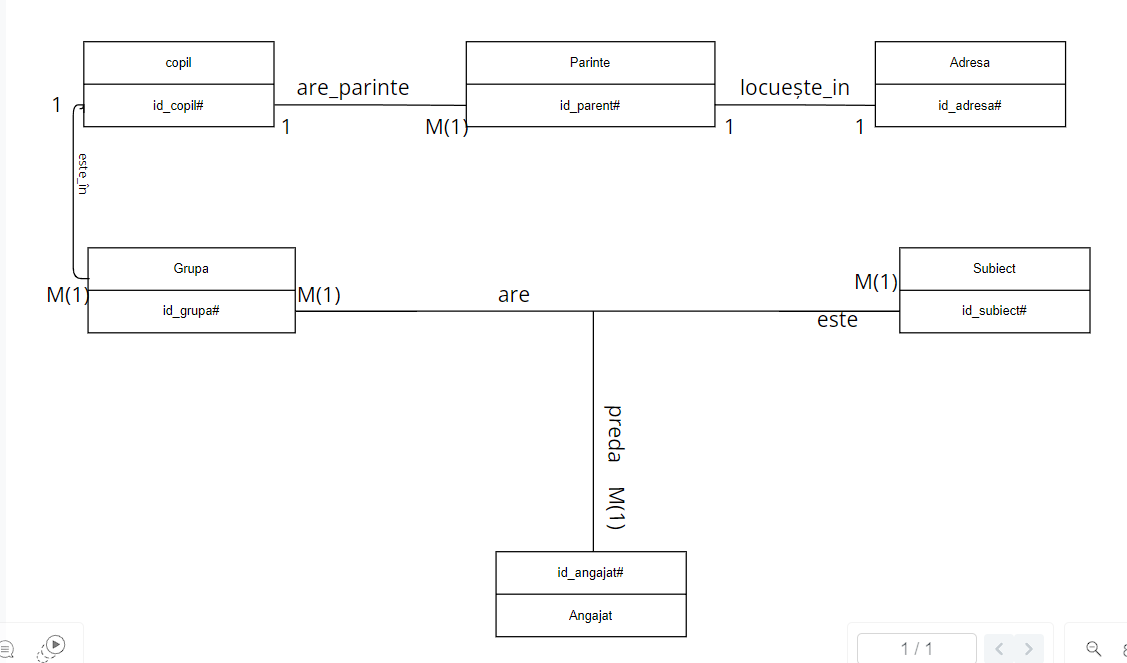
id\_lectia:variable de tip number,de lungime maximă 5,care reprezintă id-ul angajatului.

Id\_angajat: variabilă de tip number, de lungime maximă 5, care reprezintă angajatlui care predă subiect. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ANGAJAT

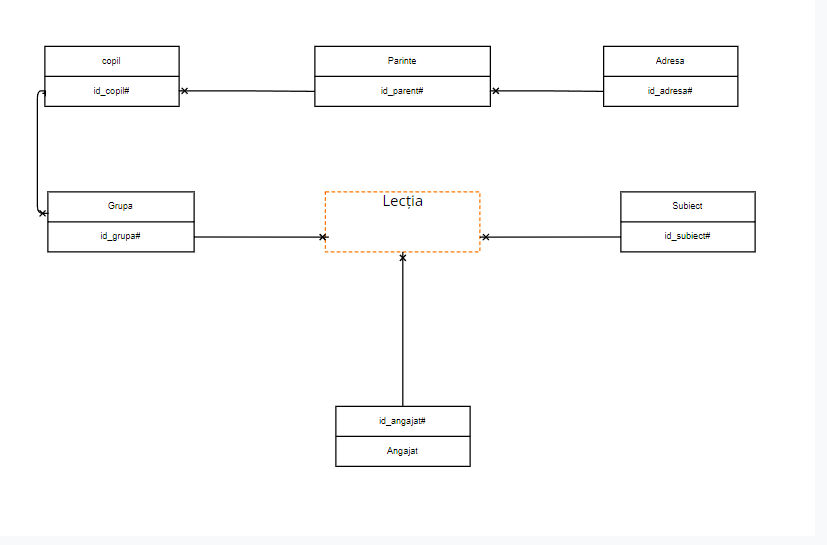
Id\_subiect: variabilă de tip number, de lungime maximă 5, care reprezintă codul subiectlui care e predat. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SUBIECT.

Id\_grupa: variabilă de tip number, de lungime maximă 5, care reprezintă grupei care participa la lectia. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul GRUPA.

# 6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



# 7. **.** Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



# 8. **.** Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

PARINTE (id\_parinte #, nume, prenume, mail, telefon, id\_adress #)

ADRESĂ (Id\_adresă#, oraș, strada, zip\_code, appartment)

COPI(Id\_copi#,nume, prenume, date\_naștere,id\_grupă,id\_parent#)

GRUPĂ (id\_grupă #, nume\_grupă,vârsta\_copi,capacitate)

ANGAJAT(id\_angajat#, nume, prenume,mail, telefon, id\_istoria#)

SUBIECT(id\_subiect#, nume)

LECȚIA(id\_lectia#,id\_angajat#, id\_subiect#,id\_grupă#,id\_sala#)

**9**. **Forma normală 1 (FN1)**

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o

valoare indivizabilă.

Forma normală 1 este și cea care impune și faptul că fiecare înregistrare să fie definită astfel încât să fie identificată unic prin intermediul unei chei primare.

În cadrul bazei de date pe care o construiesc, voi utiliza un exemplu pentru a prezenta aflarea acesteia in forma normal Voi lua entitățile **Lectie**.

|  |  |  |
| --- | --- | --- |
| Id\_angajat | Id\_subiect | Id\_grupa |
| 1 | 2 | 1,4 |
| 2 | 3 | 1,2 |
| 3 | 1 | 3,4 |

|  |  |  |
| --- | --- | --- |
| Id\_angajat | Id\_subiect | Id\_grupa |
| 1 | 2 | 1 |
| 1 | 2 | 4 |
| 2 | 3 | 1 |
| 2 | 3 | 2 |
| 3 | 1 | 3 |
| 3 | 1 | 4 |

**Forma Normală 2 (FN2)**

Știm că o relație este în a doua formă normală dacă și numai dacă ea este în FN1 și fiecare atribut care nu este cheie este dependent de întreaga cheie primară.

UN exmplu non-FN2 ar fi entitate **Lecția.**

|  |  |  |  |
| --- | --- | --- | --- |
| id\_angajat# | Nume\_angajat | Id\_grupa# | Id\_subiect# |
| 1 | Ionescu | 3 | 4 |
| 2 | Popescu | 2 | 3 |
| 3 | Georgescu | 1 | 2 |

Fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară – in cazul nostru atributul Nume\_angajat, nu este cheie si trebuie sa depinda direct de intreaga cheie primara id\_angajat #.Aceste atribute nu depind direct de intreaga cheie primara deoarece se observa dependenta directa dintre Nume\_angajat si id\_angajat, insemnand ca nume\_angajat depinde direct doar de o parte a cheii primare, si anume doar de id\_angajat, relatia nu se afla in FN2. Numele angajatilor trebui sa fie doar în entitatea activitate.

|  |  |  |
| --- | --- | --- |
| id\_angajat# | Id\_grupa# | Id\_subiect# |
| 1 | 3 | 4 |
| 2 | 2 | 3 |
| 3 | 1 | 2 |

**Forma Normală 3 (FN3)**

O relație este în a treia formă normală dacă și numai dacă este in FN2 și fiecare atribut care nu este cheie depinde direct de cheia primară.

De exemplu:student și parinte ar fi forma:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id\_copil# | Nume | Prenume | Date\_nastere | Nume\_parinte | Telefon | mail |
| 1 | Popescu | Maria | 10-Mar-21 | Alex | 5555555555 | alexbrown@gmail.com |
| 2 | Ionescu | Andrei | 25-Jul-20 | John | 6666666666 | John12@gmail.com |
| 3 | Dumitrescu | Ana | 15-Nov-21 | Alice | 7777777777 | alice@gmail.com |

Se poate observa că atributelule telefon și mail depinde de atributul nume\_Parinte care depinde la randul sau de cheia primară id\_copil, astfel nefiind in NF3.Pentru facem FN3,separ atributele nume\_parinte,telefon,mail din copil și inlocuim aceste cu chei id\_parinte.

FN3 reduce redundanța, îmbunătățește integritatea datelor, simplifică întreținerea bazei de date și optimizarea interogărilor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id\_copil# | Nume | Prenume | Date\_nastere | Id\_parinte# |
| 1 | Popescu | Maria | 10-Mar-21 | 1 |
| 2 | Ionescu | Andrei | 25-Jul-20 | 2 |
| 3 | Dumitrescu | Ana | 15-Nov-21 | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_parinte# | Nume\_parinte | Telefon | mail |
| 1 | Alex | 5555555555 | alexbrown@gmail.com |
| 2 | John | 6666666666 | alexbrown@gmail.com |
| 3 | Alice | 7777777777 | alice@gmail.com |

# 10.Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

# 

# **11.** Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

1)Creare tabelelor

**1.Parinte**

create table parinte(

id\_parinte number(5) CONSTRAINT pk\_parents primary key,

nume varchar(30) not null,

prenume varchar(30) not null,

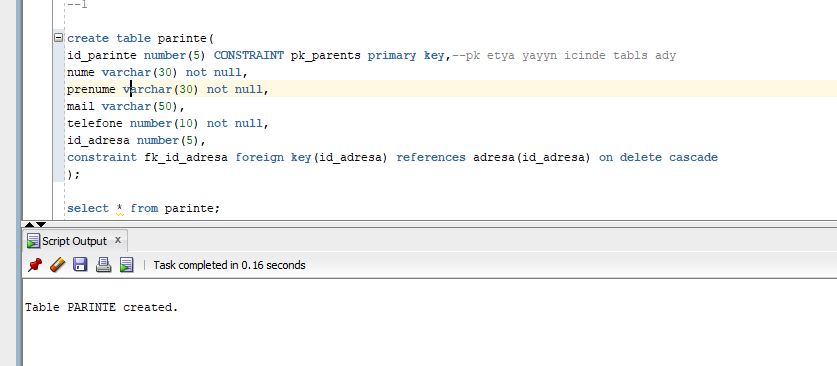
mail varchar(50) not null,

telefone number(10) not null,

adress\_id number(5),

constraint fk\_id\_adress foreign key(adress\_id) references adress(adress\_id) on delete cascade

);



# **2.Adresa**

# create table adresa(

# id\_adresa number (5) constraint pk\_adresa primary key,

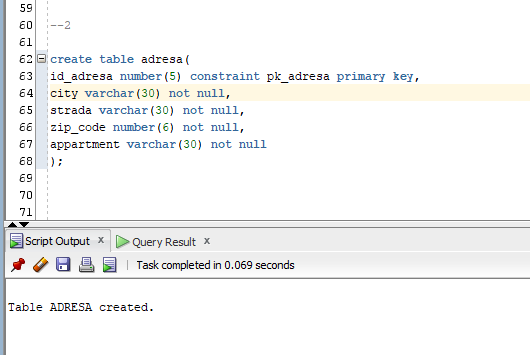
# city varchar (30) not null,

# strada varchar (30) not null,

# zip\_code number (6) not null,

# appartment varchar (30) not null

# );



**3.COPIL**

create table copil(

copil\_id number(5) constraint pk\_copil primary key,

nume varchar(30) not null,

prenume varchar(30) not null,

date\_nastere date,

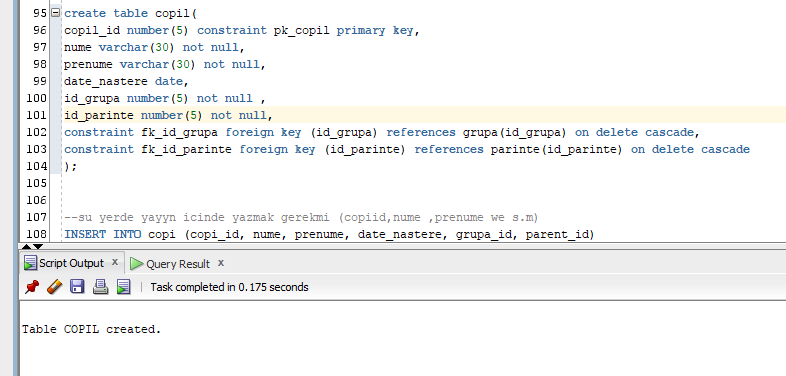
id\_grupa number(5) not null ,

id\_parinte number(5) not null,

constraint fk\_id\_grupa foreign key (id\_grupa) references grupa(id\_grupa) on delete cascade,

constraint fk\_id\_parinte foreign key (id\_parinte) references parinte(id\_parinte) on delete cascade

);



**4. Grupa**

create table grupa(

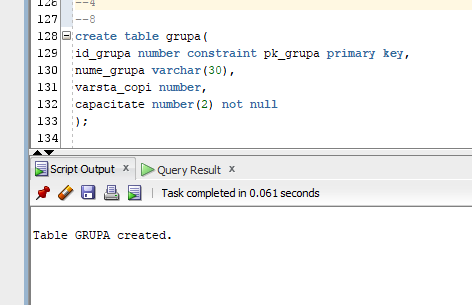
id\_grupa number constraint pk\_grupa primary key,

nume\_grupa varchar(30),

varsta\_copi number,

capacitate number(2) not null

);



**5.Angajati**

create table angajati(

id\_angajat number(5) CONSTRAINT pk\_professor primary key,

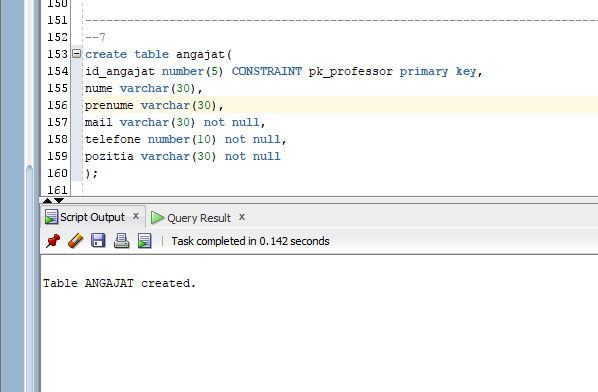
nume varchar(30),

prenume varchar(30),

mail varchar(30) not null,

telefone number(10) not null,

pozitia varchar(30)



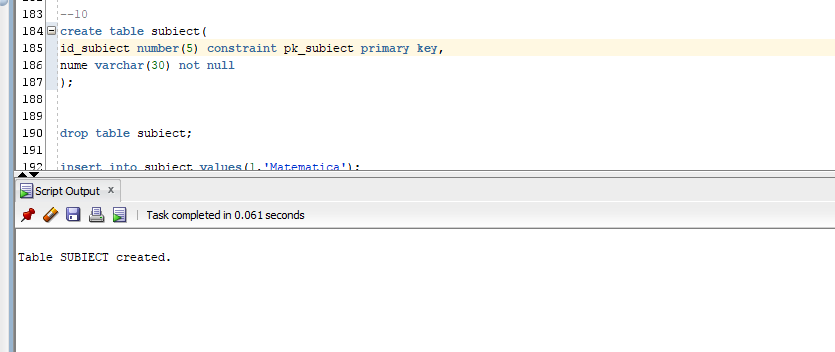
**6.Subiect**

create table subiect(

subiect\_id number(5) constraint pk\_subiect primary key,

nume varchar(30) not null

);



**7.Lectia**

create table lectia(

id\_lectia number(5)constraint pk\_lectia primary key,

id\_angajat number(5),

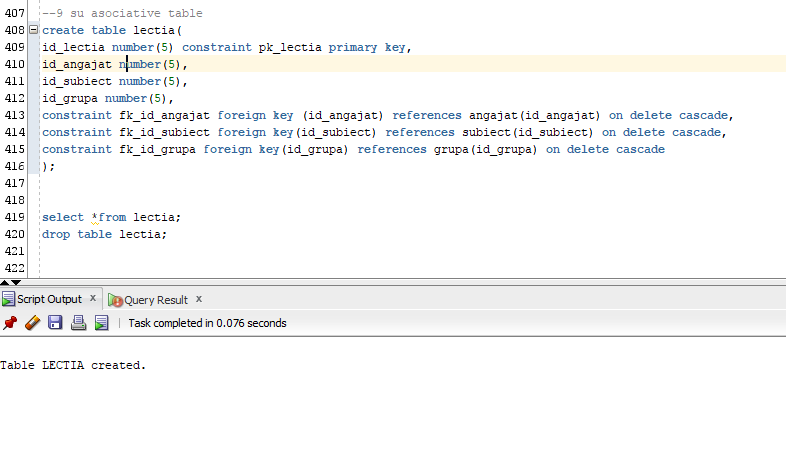
id\_subiect number(5),

id\_grupa number(5),

constraint fk\_id\_angajat foreign key (id\_angajat) references angajat(id\_angajat) on delete cascade,

constraint fk\_id\_subiect foreign key(id\_subiect) references subiect(id\_subiect) on delete cascade,

constraint fk\_id\_grupa foreign key(id\_grupa) references grupa(id\_grupa) on delete cascade



**Incerarea datelor**

**Parinte**

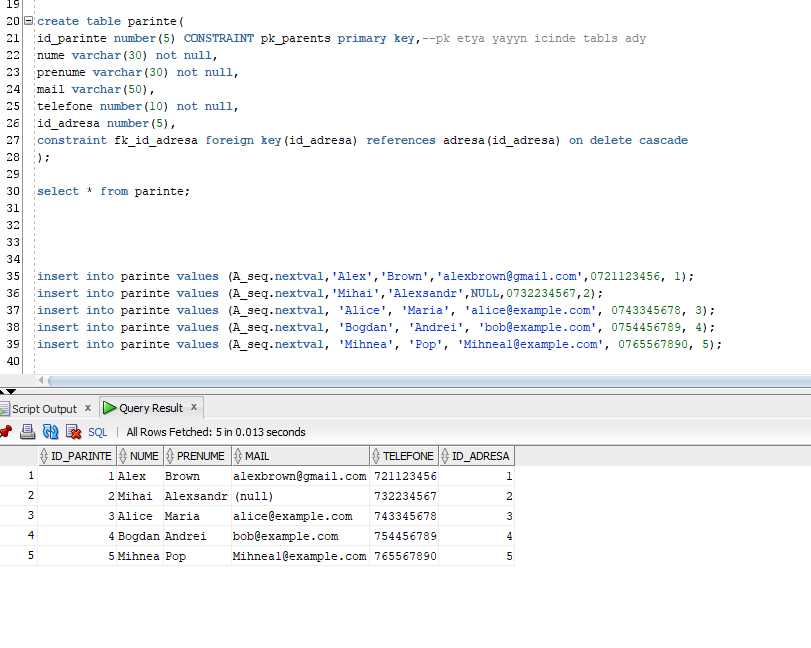
insert into parinte values (A\_seq.nextval,'Alex','Brown','alexbrown@gmail.com',0721123456, 1);

insert into parinte values (A\_seq.nextval,'Mihai','Alexsandr',NULL,0732234567,2);

insert into parinte values (A\_seq.nextval, 'Alice', 'Maria', 'alice@example.com', 0743345678, 3);

insert into parinte values (A\_seq.nextval, 'Bogdan', 'Andrei', 'bob@example.com', 0754456789, 4);

insert into parinte values (A\_seq.nextval, 'Mihnea', 'Pop', 'Mihnea1@example.com', 0765567890, 5);

****

**Adresa**

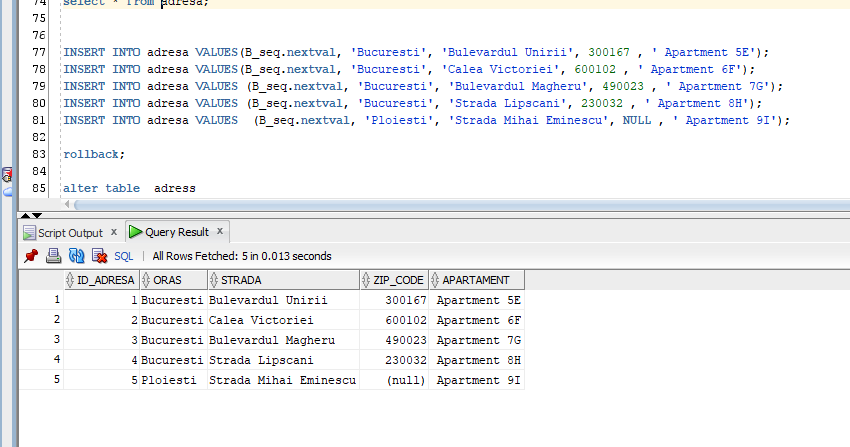
INSERT INTO adresa VALUES(B\_seq.nextval, 'Bucuresti', 'Bulevardul Unirii', 300167 , ' Apartment 5E');

INSERT INTO adresa VALUES(B\_seq.nextval, 'Bucuresti', 'Calea Victoriei', 600102 , ' Apartment 6F');

INSERT INTO adresa VALUES (B\_seq.nextval, 'Bucuresti', 'Bulevardul Magheru', 490023 , ' Apartment 7G');

INSERT INTO adresa VALUES (B\_seq.nextval, 'Bucuresti', 'Strada Lipscani', 230032 , ' Apartment 8H');

INSERT INTO adresa VALUES (B\_seq.nextval, 'Ploiesti', 'Strada Mihai Eminescu' , NULL,' Apartment 9I');

****

**Copil**

INSERT INTO copil (id\_copil, nume, prenume, date\_nastere, id\_grupa, id\_parinte)

VALUES (1, 'Popescu', 'Maria', TO\_DATE('2021-03-10', 'YYYY-MM-DD'), 1, 1);

INSERT INTO copil values (2, 'Ionescu', 'Andrei', TO\_DATE('2020-07-25', 'YYYY-MM-DD'), 2, 2);

INSERT INTO copil values (3, 'Dumitrescu', 'Ana', TO\_DATE('2021-11-15', 'YYYY-MM-DD'), 1, 3);

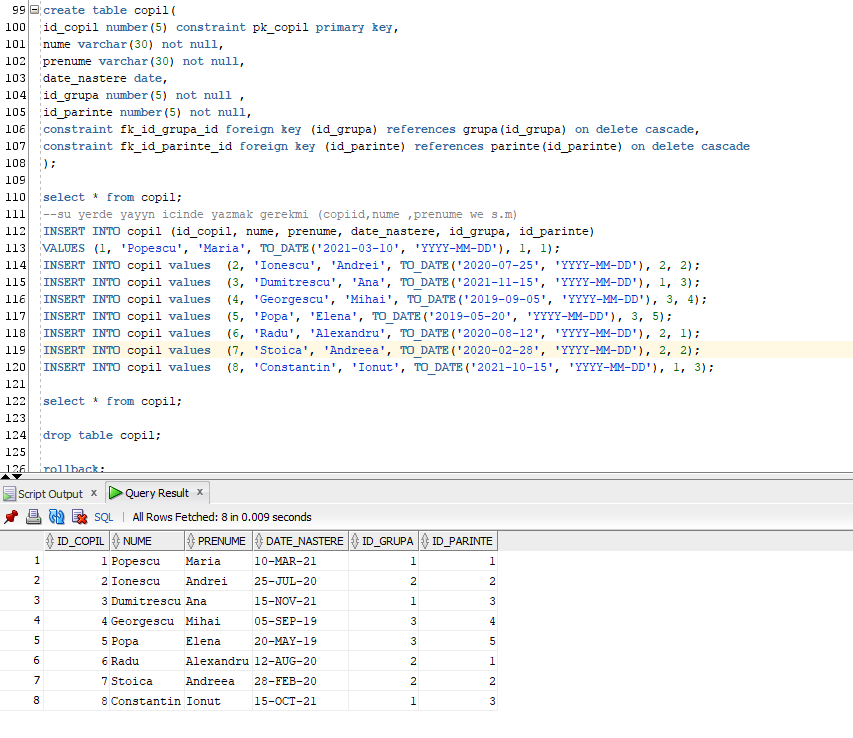
INSERT INTO copil values (4, 'Georgescu', 'Mihai', TO\_DATE('2019-09-05', 'YYYY-MM-DD'), 3, 4);

INSERT INTO copil values (5, 'Popa', 'Elena', TO\_DATE('2019-05-20', 'YYYY-MM-DD'), 3, 5);

INSERT INTO copil values (6, 'Radu', 'Alexandru', TO\_DATE('2020-08-12', 'YYYY-MM-DD'), 2, 1);

INSERT INTO copil values (7, 'Stoica', 'Andreea', TO\_DATE('2020-02-28', 'YYYY-MM-DD'), 2, 2);

INSERT INTO copil values (8, 'Constantin', 'Ionut', TO\_DATE('2021-10-15', 'YYYY-MM-DD'), 3, 3);

****

**Grupa**

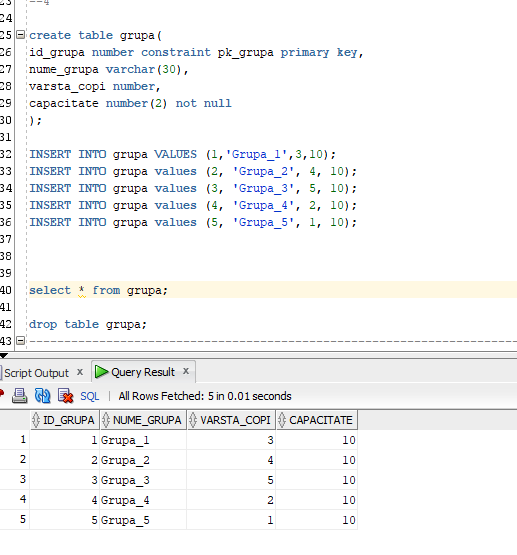
INSERT INTO grupa VALUES (1,'Grupa\_1',3,10);

INSERT INTO grupa values (2, 'Grupa\_2', 4, 10);

INSERT INTO grupa values (3, 'Grupa\_3', 5, 10);

INSERT INTO grupa values (4, 'Grupa\_4', 2, 10);

INSERT INTO grupa values (5, 'Grupa\_5', 1, 10);

****

**Angajat**

insert into angajat values(1,'Alexsandre','Anamaria','Anamariaoffice@gmail.com',0723974361,' Director');

insert into angajat values(2,'Casagandre','Martin','Martinoffice@gmail.com',0723444368,' Manager');

insert into angajat values(3, 'Popescu', 'Ion', 'popescu@example.com', 0722123456, ' prf\_matematica');

insert into angajat values (4, 'Ionescu', 'Maria', 'ionescu@example.com', 0763123456, ' Prf \_Scriss');

insert into angajat values (5, 'Dumitrescu', 'Andrei', 'dumitrescu@example.com', 0734123456, ' Prf\_Arta');

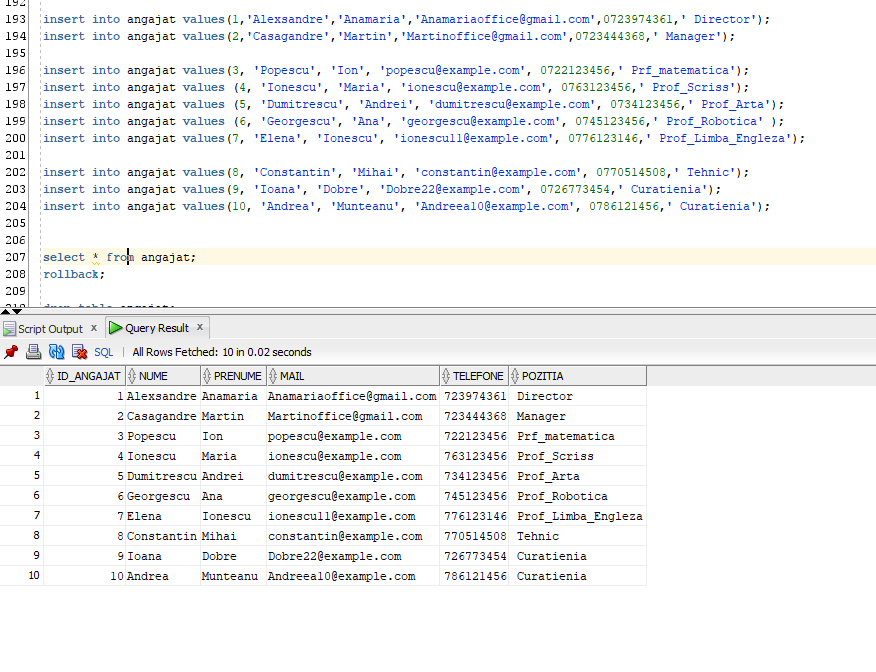
insert into angajat values (6, 'Georgescu', 'Ana', 'georgescu@example.com', 0745123456, ' Prf \_Robotica');

insert into angajat values(7, 'Elena', 'Ionescu', 'ionescu11@example.com', 0776123146, ' Prf\_limba\_Engleza');

insert into angajat values(8, 'Constantin', 'Mihai', 'constantin@example.com', 0770514508, ' Tehnic');

insert into angajat values(9, 'Ioana', 'Dobre', 'Dobre22@example.com', 0726773454, ' Curatenia');

insert into angajat values(10, 'Andrea', 'Munteanu', 'Andreea10@example.com', 0786121456, ' Curatenia');

****

**Subiect**

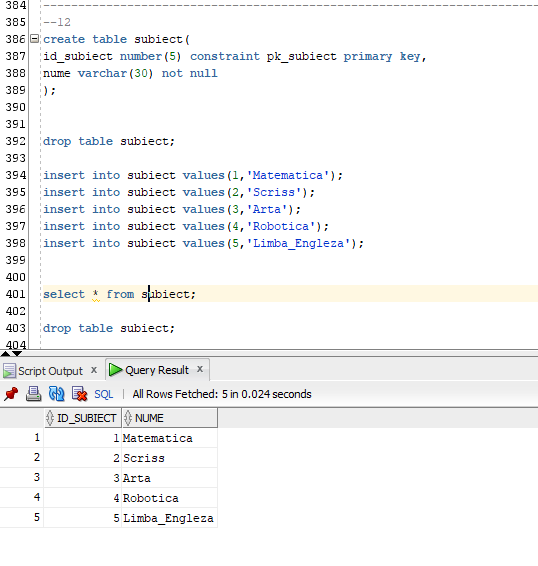
insert into subiect values(1,'Matematica');

insert into subiect values(2,'Scriss');

insert into subiect values(3,'Arta');

insert into subiect values(4,'Robotica');

insert into subiect values(5,'Limba\_Engleza');

****

**7.Lecția**

insert into lectia values(1,3,1,1);

insert into lectia values(2,5,1,3);

insert into lectia values(3,6,1,4);

insert into lectia values(4,7,1,5);

insert into lectia values(5,4,1,2);

insert into lectia values(6,3,2,1);

insert into lectia values(7,4,2,2);

insert into lectia values(8,5,2,3);

insert into lectia values(9,6,2,4);

insert into lectia values(10,7,2,5);

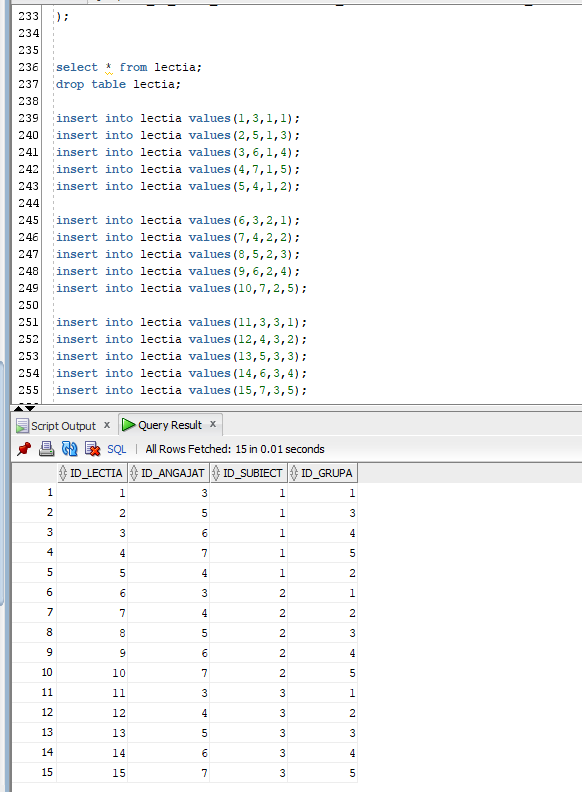
insert into lectia values(11,3,3,1);

insert into lectia values(12,4,3,2);

insert into lectia values(13,5,3,3);

insert into lectia values(14,6,3,4);

insert into lectia values(15,7,3,5);

****

**12.** 12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în

ansamblul lor, următoarele elemente:

a) subcereri sincronizate în care intervin cel puțin 3 tabele

b) subcereri nesincronizate în clauza FROM

c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate

(în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)

d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice,

a cel puțin unei expresii CASE

f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

Observație: Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel

încât cele 5 cereri să le cuprindă pe toate.

**1**.Sa afiseze numele parintilor ca nume parintilor,orasulor,telefone lor ale caror copi sunt in grupa 3.

**A.**

select p.nume as nume\_parinte,p.telefone,a.oras as Locueste

from parinte p

join adresa a on p.id\_adresa=a.id\_adresa

where p.id\_parinte in(

select id\_parinte

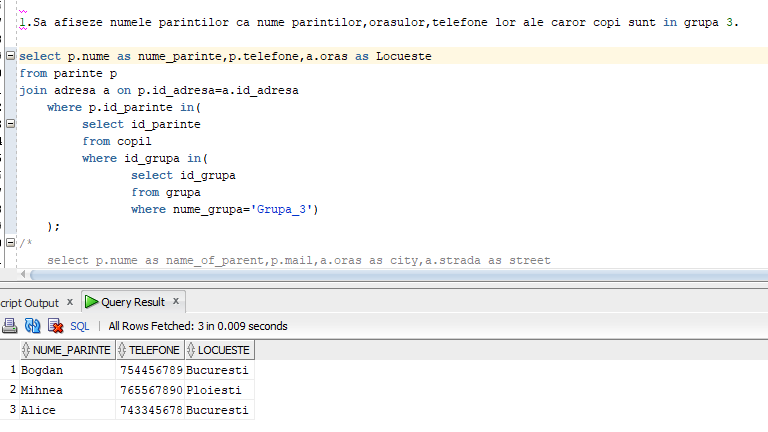
from copil

where id\_grupa in(

select id\_grupa

from grupa

where nume\_grupa='Grupa\_3')

); 

**B.** Sa se afișeze numarul de copii din fiecare grupa și grupa după numele grupului.

select g.nume\_grupa,count (id\_copil) as numar\_de\_copi

from(

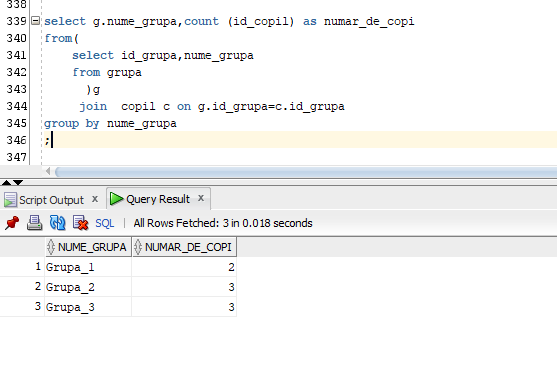
select id\_grupa,nume\_grupa

from grupa

)g

join copil c on g.id\_grupa=c.id\_grupa

group by nume\_grupaș



**C**. --3 afiseaza numele grupurilor și numărul total de copii din fiecare grup.

-- Apoi filtrează rezultatele pentru a include doar grupurile în care

--numărul total de copii depășește media numărului de copii din toate grupurile.

SELECT g.nume\_grupa, COUNT(c.id\_copil) AS total\_children

FROM grupa g

JOIN copil c ON g.id\_grupa = c.id\_grupa

GROUP BY g.nume\_grupa

HAVING COUNT(c.id\_copil) > (

SELECT AVG(child\_count)

FROM (

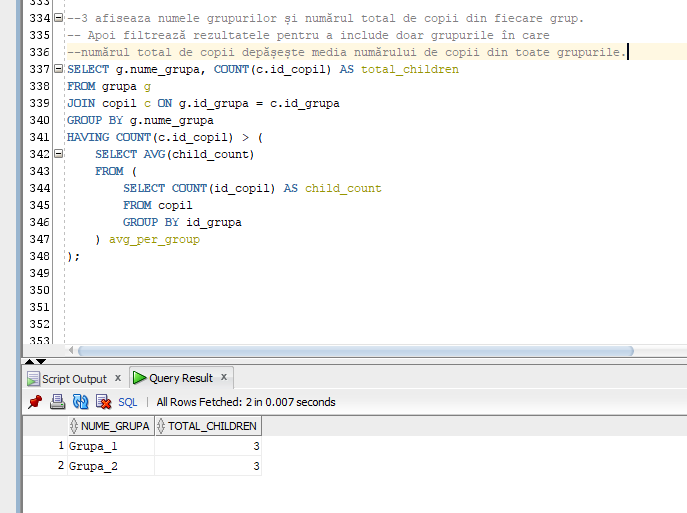
SELECT COUNT(id\_copil) AS child\_count

FROM copil

GROUP BY id\_grupa

) avg\_per\_group

);

****

**D.**  Sa afișeze nume si prenume,orasul parintilor,iar daca cineva nu are mail scrie”Nu e mail” sau una dintre ei locueste in Ploiesti pune coment ‘Locueste in alt oras’

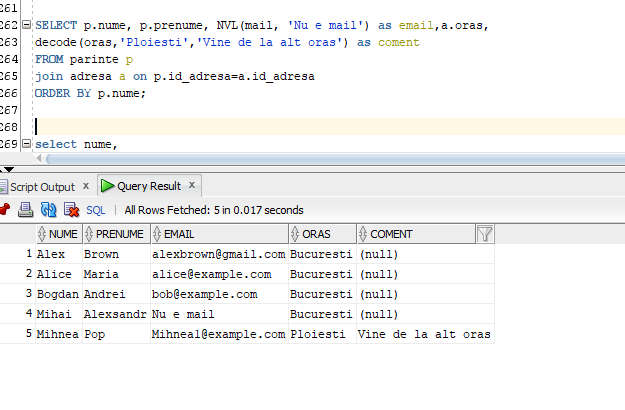
SELECT p.nume, p.prenume, NVL(mail, 'Nu e mail') as email,a.oras,

decode(oras,'Ploiesti','Vine de la alt oras') as coment

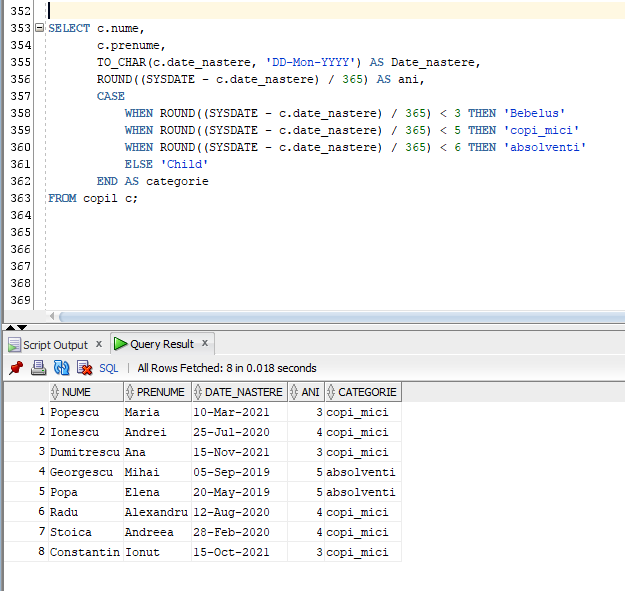
FROM parinte p

join adresa a on p.id\_adresa=a.id\_adresa

ORDER BY p.nume;



**E**.Face categorii de copi in funcție de vărsta lor.Dacă copii sunt mici de 3 ani sunt bebeluș.Dacă ei sunt mai mare de 3 ani și mai mici de 5 ani sunt copi\_mici,asltfel sunt obsolvenți.



F. sa afișeze nume și prenume parinților sa stim fiecare parinte are câți copii.După facem ordine in funcție de numele parinților.

WITH ParinteCopia as (

SELECT id\_parinte, COUNT(\*)as num\_copi

FROM copil

GROUP BY id\_parinte

)

SELECT p.nume,

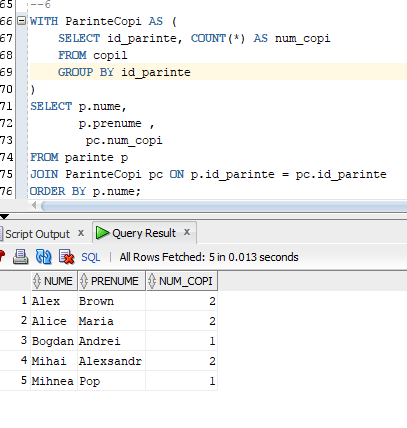
p.prenume ,

pc.num\_copi

FROM parinte p

JOIN ParinteCopi pc ON p.id\_parinte = pc.id\_parinte

ORDER BY p.nume;



**13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri**

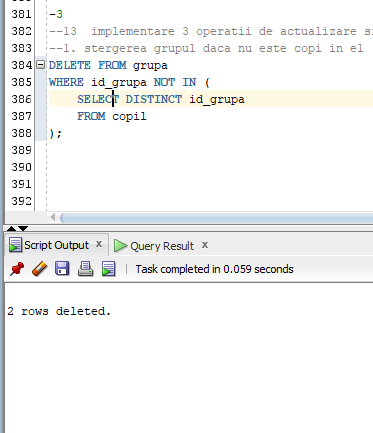
DELETE FROM grupa

WHERE id\_grupa NOT IN (

SELECT DISTINCT id\_grupa

FROM copil

);



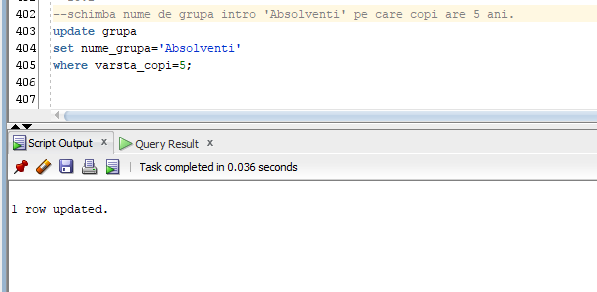
--13.2

--schimba nume de grupa intro 'Absolventi' pe care copi are 5 ani.

update grupa

set nume\_grupa='Absolventi'

where varsta\_copi=5;



--13.3 schima numarul de telefone al parintului lui Popa.

UPDATE parinte

SET telefone = 9936303238

WHERE id\_parinte IN (

SELECT id\_parinte

FROM copil

WHERE nume = 'Popa'

);

