**Stanga Ionut Catalin**

**Anul 3 Grupa 1 Informatica ID**

**Proiect Baze de Date**

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

In Bucuresti exista o retea de tramvaie complexa.

Tramvaiele au un numar de scaune, un model al vagoanelor si o data a ultimei revizii.

Acestea sunt conduse de catre vatmani, despre care se cunosc numarul vagonului de care se ocupa, numele, experienta in ani si salariul.

Tramvaiele se retrag in afara orelor de circulatie la depourile din oras. Fiecare depou are un nume, o adresa si o capacitate maxima de garare a tramvaielor. In plus, se cunoaste sectorul in care se afla depoul.

Fiecare depou are un sef de depou care este raspunzator de buna desfasurare a muncii. Despre acestia se cunosc numele, numarul de telefon, dar si depoul de care au grija.

Tramvaiele circula pe trasee prestabilite, numite linii. Liniile au un numar dupa care sunt usor de recunoscut pentru calator. Se cunosc despre fiecare linie cele doua capete de linie (statii). In plus, deoarece Bucurestiul detine o flota restransa de tramvaie bidirectionale, este necesar ca acestea sa fie trimise pe liniile ce necesita tramvaie bidirectionale pentru a functiona. Astfel, despre fiecare linie se cunoaste daca aceasta necesita tramvaie bidirectionale.

In final, de-a lungul traseelor tramvaiele opresc in mai multe statii pentru a lasa si a prelua calatori. Statiile sunt cunoscute dupa numele sugestiv. Alte informatii relevante cunoscute despre statii sunt lungimea peronului (care, in caz ca nu exista cum este cazul unor statii, este 0), numarul mediu de calatori care utilizeaza statia cat si numarul de linii care trec prin statia respectiva. De asemenea, deoarece se doreste o integrare intre diversele tipuri de transport in comun, dar si pentru motive statistice, se cunoaste si ce statii au legatura cu metroul bucurestean.

Pe o linie pot circula mai multe tramvaie. Se cunoaste pentru fiecare statie ce linii opresc in aceasta.

2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

Fiecare sef de depou se ocupa de un singur depou, iar fiecare depou are un singur sef de depou.

Fiecare tramvai se retrage la un depou, iar un depou poate gara 0 sau mai multe tramvaie.

Fiecare vatman conduce un tramvai, iar fiecare tramvai este condus de un singur vatman.

O linie poate fi operata de 1 sau mai multe tramvaie, iar un tramvai trebuie sa fie asignat cel putin unui traseu.

O linie trece prin mai multe statii, iar o statie poate sa deserveasca 0 sau mai multe linii.

3. Descrierea entităților, incluzând precizarea cheii primare.

| **ENTITATE** | **CHEIE PRIMARA** | **DESCRIERE** |
| --- | --- | --- |
| Tramvai | id\_tramvai | Informatii despre tramvaiele din retea. |
| Depou | id\_depou | Informatiile despre depourile in care se retrag tramvaiele. |
| SefDepou | id\_sef\_depou | Informatiile despre sefii de depou care se ocupa de depouri. |
| Vatman | id\_vatman | Informatii despre vatmanii ce conduc tramvaiele. |
| Linii | numar\_linie | Liniile reprezinta traseele urmate de tramvaie dupa care se ghideaza calatorii (identificat prin numarele lor). |
| Statie | nume\_statie | Statiile in care opresc tramvaiele si in care asteapta calatorii sa vina transportul. |

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

| **RELATIE** | **CARDINALITATE** | **OBSERVATII** |
| --- | --- | --- |
| administreaza | SefDepou-Depou,  One-to-One | Fiecare sef de depou administreaza un singur depou.  Un depou are un singur sef. |
| gareaza | Tramvai-Depou,  One-to-Many | Fiecare tramvai este garat intr-un depou.  Un depou adaposteste mai multe tramvaie. |
| conduce | Vatman-Tramvai,  One-to-One | Un tramvai este condus de un singur vatman.  Fiecare vatman are tramvaiul lui. |
| alimenteaza | Tramvai-Linie,  Many-to-Many | Un tramvai poate circula pe una sau mai multe linii.  O linie este alimentata de mai multe tramvaie. |
| deserveste | Linie-Statie,  Many-to-Many | Fiecare traseu trece prin mai multe statii.  O statie deserveste cel putin unei linii. |

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

ENTITATE: SEFDEPOU

| **Atribut** | **Tip** | **Dimensiune/**  **precizie** | Valori posibile si valori default | Observatii, obligatoriu/  optional |
| --- | --- | --- | --- | --- |
| nume | string | 30 |  |  |
| telefon | string | 11 |  |  |

ENTITATE: DEPOU

| **Atribut** | **Tip** | **Dimensiune/**  **precizie** | Valori posibile si valori default | Observatii, obligatoriu/  optional |
| --- | --- | --- | --- | --- |
| nume | string | 20 |  |  |
| adresa | string | 40 |  |  |
| sector | int |  | 1,2,3,4,5 sau 6 |  |
| capacitate | int |  |  | Se refera la numarul de tramvaie maxim care pot gara in depou |

ENTITATE: TRAMVAI

| **Atribut** | **Tip** | **Dimensiune/**  **precizie** | Valori posibile si valori default | Observatii, obligatoriu/  optional |
| --- | --- | --- | --- | --- |
| numar\_scaune | int |  |  |  |
| model\_vagon | string | 20 |  | Modelul vagonului de tramvai, care este updatat pe masura ce noi modele apar in circulatie |
| data\_revizie | date |  |  | Data la care s-a facut ultima revizie a tramvaiului |

ENTITATE: VATMAN

| **Atribut** | **Tip** | **Dimensiune/**  **precizie** | Valori posibile si valori default | Observatii, obligatoriu/  optional |
| --- | --- | --- | --- | --- |
| nume | string | 30 |  |  |
| experienta | int |  |  | Experienta in domeniu masurata in ani |
| salariu | int |  |  |  |

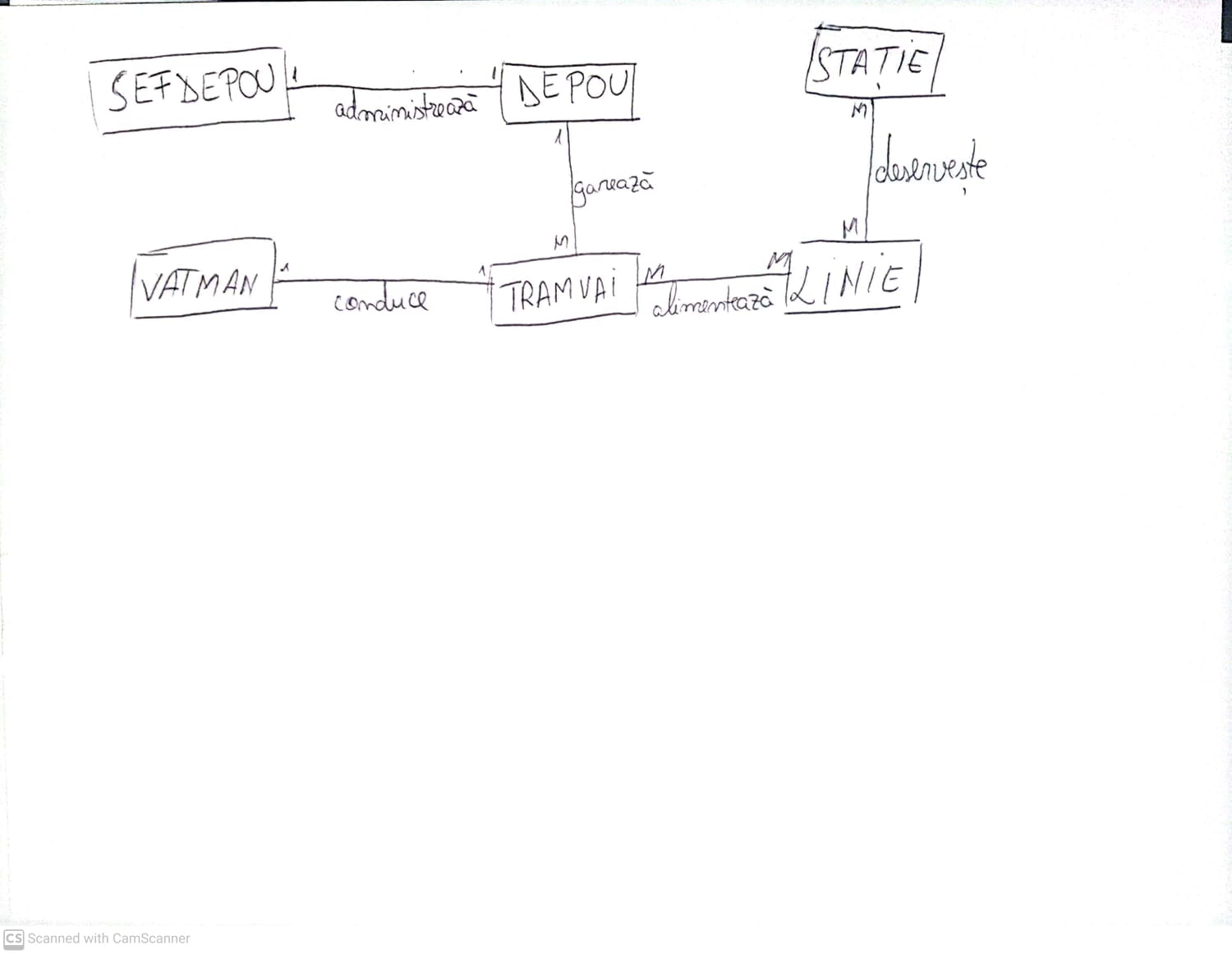
ENTITATE: LINIE

| **Atribut** | **Tip** | **Dimensiune/**  **precizie** | Valori posibile si valori default | Observatii, obligatoriu/  optional |
| --- | --- | --- | --- | --- |
| numar\_tramvaie | int |  |  | Numarul tramvaielor care circula pe linia respectiva |
| statie\_start | string | 20 |  | Statiile de extremitate ale traseului, unde de obicei tramvaiul stationeaza mai mult timp iar vatmanul face o pauza. |
| statie\_stop | string | 20 |  |
| necesita\_bidirectionale | string | 3 | “DA”,”NU” |  |

ENTITATE: STATIE

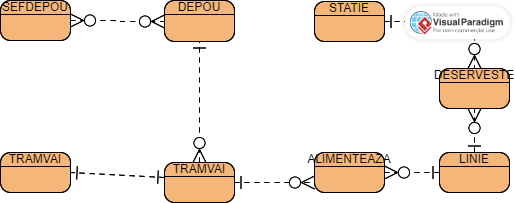
| **Atribut** | **Tip** | **Dimensiune/**  **precizie** | Valori posibile si valori default | Observatii, obligatoriu/  optional |
| --- | --- | --- | --- | --- |
| lungime\_peron | int |  |  | Unele statii nu au un peron efectiv, pietonii trebuind sa mearga de pe trotuar pana la tramvai cand acesta ajunge in statie. Acest lucru este reflectat in peroanele de lungime 0. |
| sector | int |  | 1,2,3,4,5 sau 6 |  |
| nr\_calatori | int |  |  |  |
| nr\_linii | int |  |  |  |
| Legatura\_metrou |  |  | “DA”,”NU” |  |

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.

OBSERVATIE: Se poate folosi pentru diagrama conceptuală și crow-foot.



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

SefDepou(id\_sef\_depou#, id\_depou, nume, telefon)

Depou(id\_depou#, nume, adresa, capacitate, sector)

Tramvai(numar\_vagon#, id\_depou, numar\_scaune, model\_vagon, data\_revizie)

Vatman(id\_vatman#, numar\_vagon, nume, experienta, salariu)

Alimenteaza(numar\_vagon#, numar\_linie#)

Linie(numar\_linie#, numar\_tramvaie, statie\_start, statie\_stop, necesita\_bidirectionale)

Deserveste(nume\_statie#, numar\_linie#)

Statie(nume\_statie#, lungime\_peron, sector, nr\_calatori, nr\_linii, legatura\_metrou)

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

Forma intiala a tabelei tramvai:

tramvai(numar\_vagon PK, id\_depou, nume\_depou, linie, vatman, numar\_scaune, model\_vagon, data\_revizie)

depou(id\_depou, nume, id\_sef\_depou, nume\_sef\_depou, capacitate, tramvai1, tramvai2, …)

Identificam tramvai1, tramvai2,... ca attribute care se repeta. Pentru a trece in FN1, cream o entitate Tramvai, care va retine id\_depou al depoului unde este garat.

depou(id\_depou, nume, id\_sef\_depou, nume\_sef\_depou, capacitate)

tramvai(numar\_vagon, id\_depou)

Identificam o cheie primara multipla in id\_depou si id\_sef\_depou. Pentru a trece in FN2, cream o noua entitate SefDepou care va prelua informatia id\_sef\_depou si nume\_sef\_depou.

depou(id\_depou, nume, capacitate)

sefdepou(id\_sef\_depou, nume\_sef\_depou)

Pentru a trece in FN3, eliminam atributele din depou care depind de alt atribut decat cheia primara. Deoarece nu avem niciun astfel de atribut, schema este in FN3.

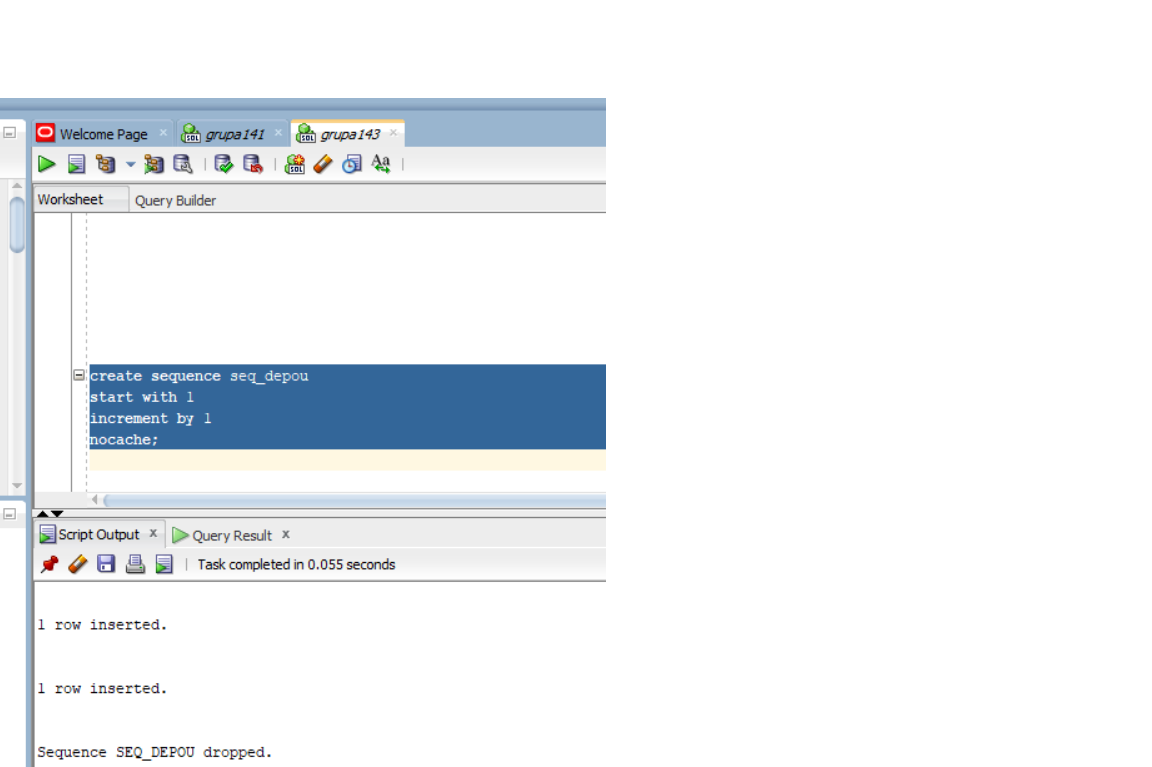
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

create sequence seq\_depou

start with 1

increment by 1

nocache;



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; maxim 30 de înregistrări în fiecare tabel).

create sequence seq\_depou

start with 1

increment by 1

nocache;

CREATE TABLE Depou(

id\_depou INT NOT NULL,

nume VARCHAR2(20),

adresa VARCHAR(40),

capacitate INT,

sector INT,

CONSTRAINT pk\_depou PRIMARY KEY (id\_depou)

);

CREATE TABLE SefDepou (

id\_sef\_depou INT NOT NULL,

id\_depou INT,

nume VARCHAR(30),

telefon VARCHAR(11),

CONSTRAINT pk\_sefdepou PRIMARY KEY (id\_sef\_depou),

CONSTRAINT fk\_sefdepou FOREIGN KEY (id\_depou) REFERENCES Depou(id\_depou)

);

CREATE TABLE Tramvai (

numar\_vagon INT NOT NULL,

id\_depou INT,

numar\_scaune INT,

model\_vagon VARCHAR2(20),

data\_revizie DATE,

CONSTRAINT pk\_tramvai PRIMARY KEY (numar\_vagon),

CONSTRAINT fk\_tramvai FOREIGN KEY (id\_depou) REFERENCES Depou(id\_depou)

);

CREATE TABLE Vatman (

id\_vatman INT NOT NULL,

numar\_vagon INT,

nume VARCHAR2(30),

experienta INT,

salariu INT,

CONSTRAINT pk\_vatman PRIMARY KEY (id\_vatman),

CONSTRAINT fk\_vatman FOREIGN KEY (numar\_vagon) REFERENCES Tramvai(numar\_vagon)

);

CREATE TABLE Linie (

numar\_linie INT NOT NULL,

numar\_tramvaie INT,

statie\_start VARCHAR2(20),

statie\_stop VARCHAR2(20),

necesita\_bidirectionale VARCHAR2(3),

CONSTRAINT pk\_linie PRIMARY KEY (numar\_linie)

);

CREATE TABLE Statie (

nume\_statie VARCHAR2(20) NOT NULL,

lungime\_peron INT,

sector INT,

nr\_calatori INT,

nr\_linii INT,

legatura\_metrou VARCHAR2(3),

CONSTRAINT pk\_statie PRIMARY KEY (nume\_statie)

);

CREATE TABLE Alimenteaza (

numar\_vagon INT NOT NULL,

numar\_linie INT NOT NULL,

CONSTRAINT pk\_alimenteaza PRIMARY KEY (numar\_vagon, numar\_linie),

CONSTRAINT fk1\_alimenteaza FOREIGN KEY (numar\_vagon) REFERENCES Tramvai (numar\_vagon),

CONSTRAINT fk2\_alimenteaza FOREIGN KEY (numar\_linie) REFERENCES Linie(numar\_linie)

);

CREATE TABLE Deserveste (

nume\_statie VARCHAR2(20) NOT NULL,

numar\_linie INT NOT NULL,

CONSTRAINT pk\_deserveste PRIMARY KEY (nume\_statie, numar\_linie),

CONSTRAINT fk1\_deserveste FOREIGN KEY (nume\_statie) REFERENCES Statie (nume\_statie),

CONSTRAINT fk2\_deserveste FOREIGN KEY (numar\_linie) REFERENCES Linie (numar\_linie)

);

INSERT INTO Depou VALUES (seq\_depou.nextval-1,'Militari' , 'Str. Preciziei 19-21' , 84 , 6);

INSERT INTO Depou VALUES (seq\_depou.nextval-1,'Dudesti' , 'Calea Dudesti 184' , 65 , 3);

INSERT INTO Depou VALUES (seq\_depou.nextval-1,'Colentina' , 'Sos. Colentina 278' , 35 , 2);

INSERT INTO Depou VALUES (seq\_depou.nextval-1,'Alexandria' , 'Sos. Alexandria 5' , 70 , 5);

INSERT INTO Depou VALUES (seq\_depou.nextval-1,'Victoria' , 'Str. Mexic 19' , 84 , 1);

INSERT INTO Depou VALUES (seq\_depou.nextval-1,'Bucurestii Noi' , 'Bd. Bucurestii Noi 42' , 25 , 1);

INSERT INTO SefDepou VALUES (1, 3 , 'Caramfil Stefan' , '0700000001');

INSERT INTO SefDepou VALUES (2, 4 , 'Nitu Ioan' , '0700000002');

INSERT INTO SefDepou VALUES (3, 5 , 'Stoica Laurentiu' , '0700000003');

INSERT INTO SefDepou VALUES (4, 1 , 'Szabo George' , '0700000004');

INSERT INTO SefDepou VALUES (5, 6 , 'Pancu Dumitru' , '0700000005');

INSERT INTO SefDepou VALUES (6, 2 , 'Dobre Mihai' , '0700000006');

INSERT INTO Tramvai VALUES (39, 5 , 28 , 'V3A-93-2S', TO\_DATE('2022-11-26','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (104, 5 , 25 , 'V3A-93-2S', TO\_DATE('2022-08-13','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (77, 1 , 16 , 'TATRA T4R', TO\_DATE('2020-08-03','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (78, 2 , 28 , 'V3A-93', TO\_DATE('2022-11-26','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (1001, 1 , 44 , 'Astra Imperio', TO\_DATE('2023-04-15','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (1002, 1 , 44 , 'Astra Imperio', TO\_DATE('2023-04-17','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (13, 2 , 39 , 'Bucur LF', TO\_DATE('2022-12-04','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (202, 4 , 26 , 'V3A-93-PPC', TO\_DATE('2023-02-16','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (216, 6 , 26 , 'V3A-93-PPC', TO\_DATE('2023-03-06','yyyy-mm-dd'));

INSERT INTO Tramvai VALUES (345, 3 , 28 , 'V3A-93', TO\_DATE('2022-11-26','yyyy-mm-dd'));

INSERT INTO Vatman VALUES (1, 345 , 'Vasile Andrei' , 12 , 9800);

INSERT INTO Vatman VALUES (2, 77 , 'Popescu Stelian' , 7 , 8400);

INSERT INTO Vatman VALUES (3, 1001 , 'Stancu Alexandra' , 10 , 9000);

INSERT INTO Vatman VALUES (4, 1002 , 'Ignat Lenuta' , 1 , 5500);

INSERT INTO Vatman VALUES (5, 216 , 'Delea Daniel' , 5 , 7600);

INSERT INTO Vatman VALUES (6, 13 , 'Ionescu Paul' , 11 , 9300);

INSERT INTO Vatman VALUES (7, 39 , 'Sandu Camelia' , 2 , 5600);

INSERT INTO Vatman VALUES (8, 104 , 'Diaconescu Denis' , 5 , 7600);

INSERT INTO Vatman VALUES (9, 202 , 'Georgescu Victor' , 8 , 8800);

INSERT INTO Vatman VALUES (10, 78 , 'Vanea Gabriel' , 4 , 6600);

INSERT INTO Vatman VALUES (12, NULL , 'Eftimie Ioan' , 0 , 5000);

INSERT INTO Vatman VALUES (13, NULL , 'Militaru Ionela' , 3 , 6300);

INSERT INTO Linie VALUES (1, 2, 'Romprim', 'Banu Manta', 'NU');

INSERT INTO Linie VALUES (41, 2, 'Ghencea', 'Piata Presei', 'NU');

INSERT INTO Linie VALUES (44, 2, 'Cartier 16 februarie', 'Vasile Parvan', 'NU');

INSERT INTO Linie VALUES (5, 2, 'Piata Sf. Gheorghe', 'Piata Baneasa', 'DA');

INSERT INTO Linie VALUES (21, 1 , 'Romprim', 'Banu Manta', 'NU');

INSERT INTO Linie VALUES (32, 1, 'Piata Unirii', 'Dep. Alexandria', 'NU');

INSERT INTO Statie VALUES ( 'Bd. Lacul Tei', 30, 2, 1500, 1, 'NU');

INSERT INTO Statie VALUES ( 'Sos. Progresu', 70, 5, 13000, 2, 'NU');

INSERT INTO Statie VALUES ( 'Piata Crangasi', 30, 6, 35000, 1, 'DA');

INSERT INTO Statie VALUES ( 'Piata Obor', 30, 2, 22000, 2, 'DA');

INSERT INTO Statie VALUES ( 'Gradina Cismigiu', 0, 1, 250, 1, 'NU');

INSERT INTO Statie VALUES ( 'Gara Basarab', 45, 6, 10000, 2, 'DA');

INSERT INTO Statie VALUES ( 'Piata Gemeni', 0, 3, 1000, 1, 'NU');

INSERT INTO Statie VALUES ( 'Godeni', 0, 6, 750, 1, 'NU');

INSERT INTO Statie VALUES ( 'Piata Muncii', 35, 4, 18000, 1, 'DA');

INSERT INTO Statie VALUES ( 'Piata Unirii', 30, 5, 29000, 1, 'DA');

INSERT INTO Alimenteaza VALUES (13,1);

INSERT INTO Alimenteaza VALUES (78,1);

INSERT INTO Alimenteaza VALUES (1001,41);

INSERT INTO Alimenteaza VALUES (1002,41);

INSERT INTO Alimenteaza VALUES (77,44);

INSERT INTO Alimenteaza VALUES (216,44);

INSERT INTO Alimenteaza VALUES (39,5);

INSERT INTO Alimenteaza VALUES (104,5);

INSERT INTO Alimenteaza VALUES (345,21);

INSERT INTO Alimenteaza VALUES (202,32);

INSERT INTO Deserveste VALUES ('Bd. Lacul Tei',5);

INSERT INTO Deserveste VALUES ('Sos. Progresu',1);

INSERT INTO Deserveste VALUES ('Sos. Progresu',32);

INSERT INTO Deserveste VALUES ('Piata Crangasi',41);

INSERT INTO Deserveste VALUES ('Piata Obor',21);

INSERT INTO Deserveste VALUES ('Piata Obor',1);

INSERT INTO Deserveste VALUES ('Gradina Cismigiu',44);

INSERT INTO Deserveste VALUES ('Gara Basarab',1);

INSERT INTO Deserveste VALUES ('Gara Basarab',44);

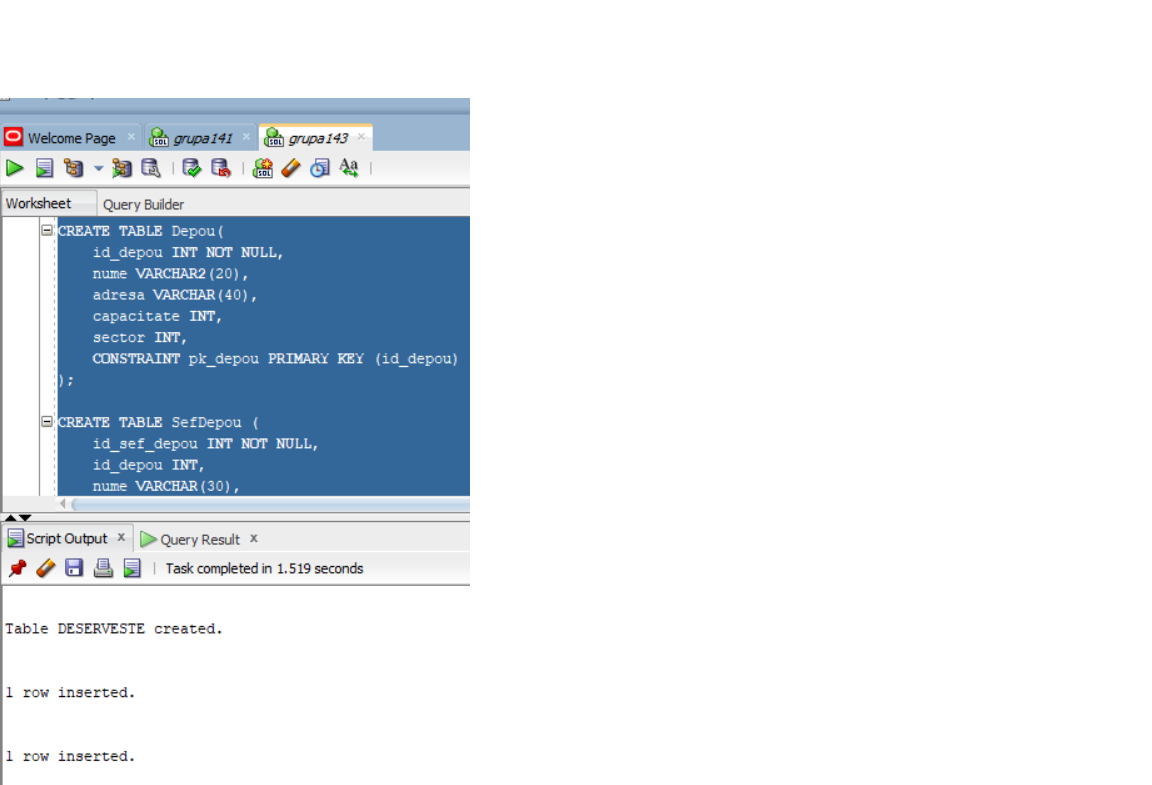
INSERT INTO Deserveste VALUES ('Piata Gemeni',5);

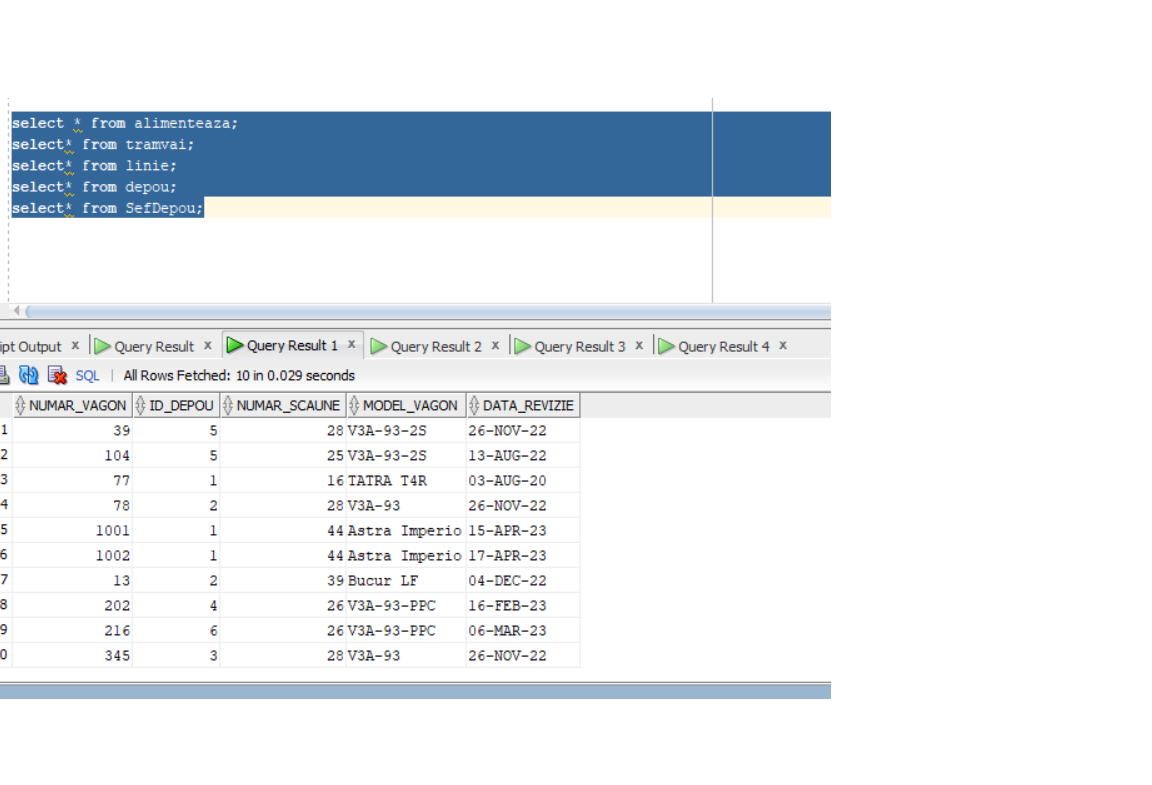
INSERT INTO Deserveste VALUES ('Godeni',44);

INSERT INTO Deserveste VALUES ('Piata Muncii',1);

INSERT INTO Deserveste VALUES ('Piata Unirii',32);

drop sequence seq\_depou;





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.

//Avand in vedere dorinta de a oferi recompensa echitabila, bazata pe munca depusa, se doreste aflarea numelui si salariilor vatmanilor ce circula pe cel mai dificil traseu (linia 1) care castiga sub medie

//cerinta a)

select v.nume, v.salariu

from vatman v join tramvai t on v.numar\_vagon = t.numar\_vagon

join alimenteaza a on t.numar\_vagon = a.numar\_vagon

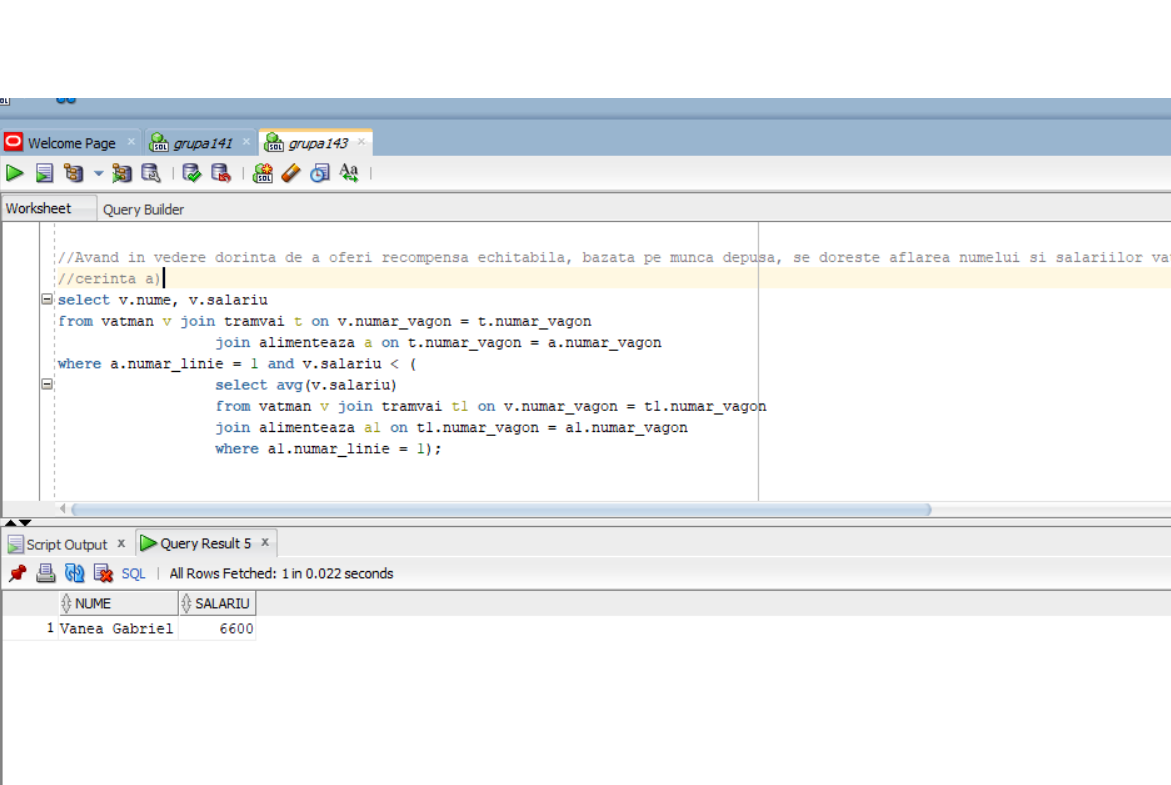
where a.numar\_linie = 1 and v.salariu < (

select avg(v.salariu)

from vatman v join tramvai t1 on v.numar\_vagon = t1.numar\_vagon

join alimenteaza a1 on t1.numar\_vagon = a1.numar\_vagon

where a1.numar\_linie = 1);



//Este necesara revizia periodica a unor tramvaie (cu ultima revizie cu mai mult de un an jumatate in urma). Este necesar sa se cunoasca in ce depouri urmeaza sa participe la revizie. Sa se afle fiecare depou care gareaza tramvaie cu revizie >18 luni cat si numarul de astfel de tramvaie.

//cerinta e),c)

select d.nume as depou, count(t.numar\_vagon) as numar\_tramvaie

from Depou d

join (

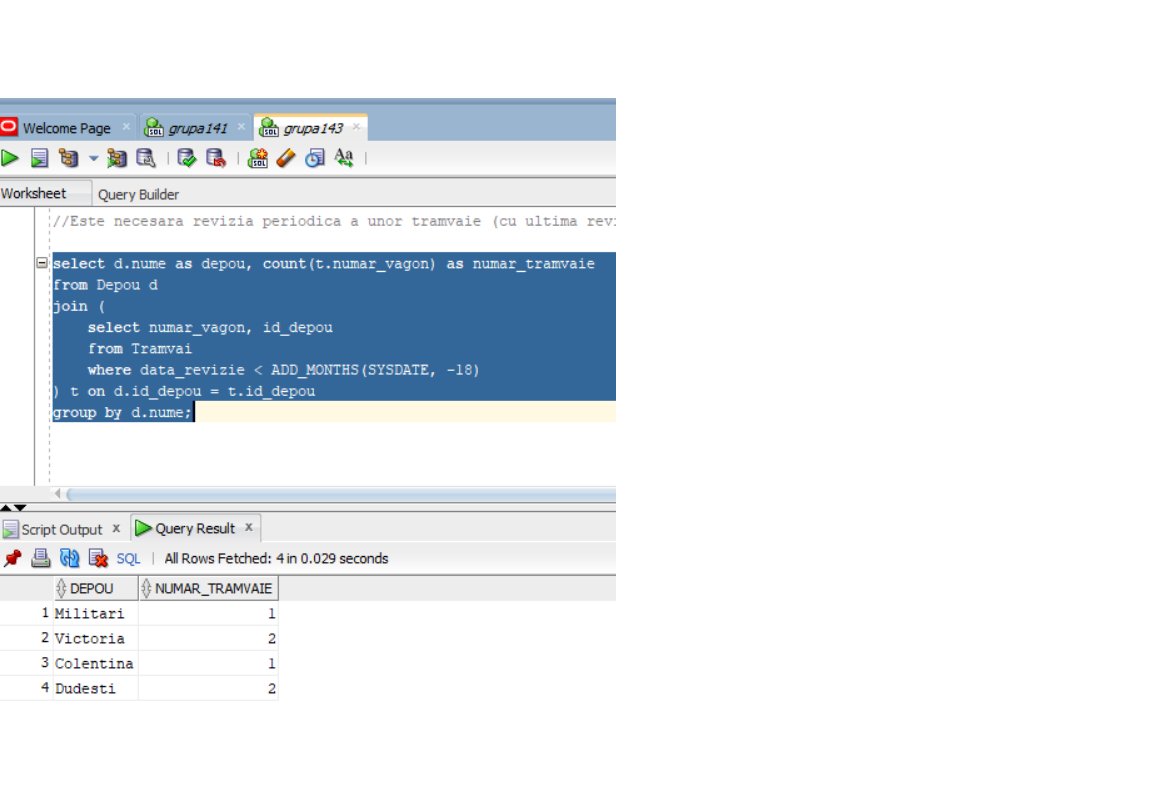
select numar\_vagon, id\_depou

from Tramvai

where data\_revizie < ADD\_MONTHS(SYSDATE, -18)

) t on d.id\_depou = t.id\_depou

group by d.nume;



//Sa se afle, pentru fiecare vatman, numele si salariul acestuia, cat si daca se afla sub, deasupra sau fix la media salariilor.

//cerinta f), d),e)

with MedieSalariu as (

select avg(salariu) as media\_salariilor

from vatman

)

select nume, salariu, NVL(to\_char(numar\_vagon), 'Vagon in mentenanta/neasignat') as Vagon,

decode(sign(v.salariu-m.media\_salariilor), 1, 'Salariu peste medie',

-1, 'Salariu sub medie',

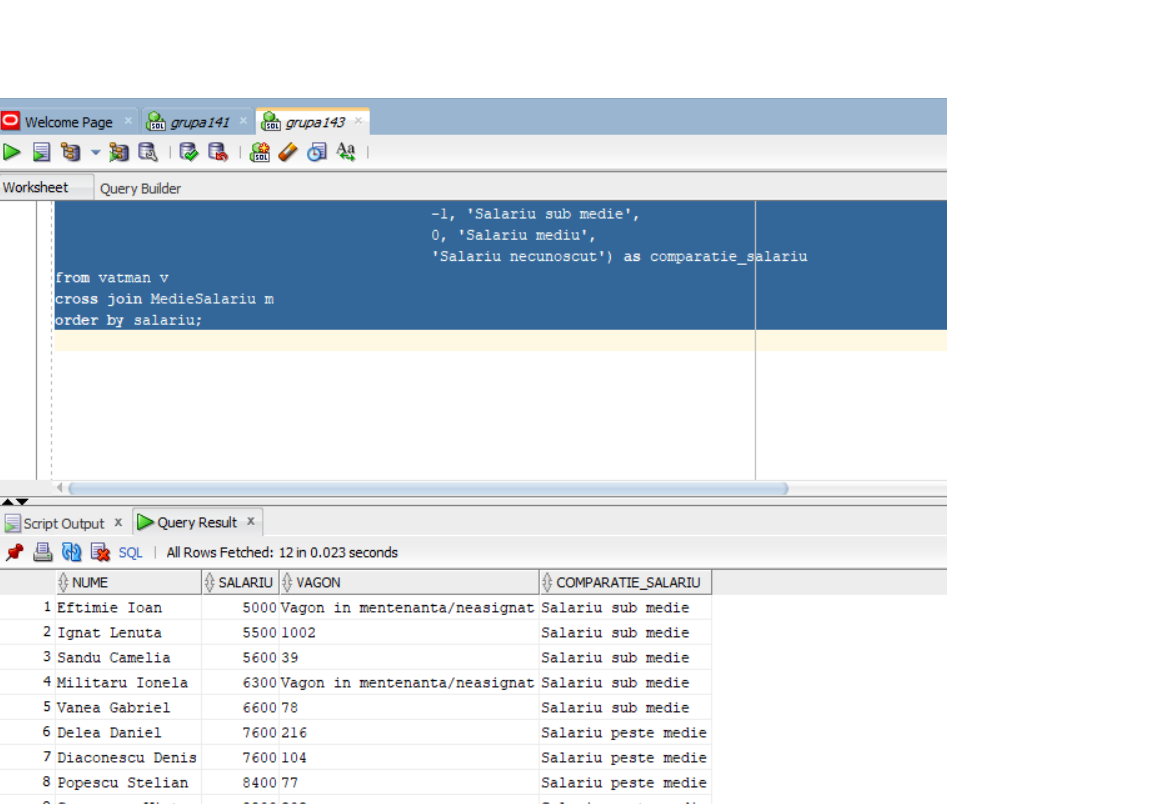
0, 'Salariu mediu',

'Salariu necunoscut') as comparatie\_salariu

from vatman v

cross join MedieSalariu m

order by salariu;



//Deoarece flota de tramvaie bidirectionale este limitata in Bucuresti, trebuie cunoscuta situatia acestora in amanunt. Sa se afiseze numele si capacitatea de transport (suma locurilor tramvaielor bidirectionale) a depourilor care gareaza tramvaie care circula pe linii ce necesita tramvaie bidirectionale.

//cerinta e), b), c), f)

select upper(d.nume) as nume\_depou, sum(t.numar\_scaune) as capacitate

from depou d

join tramvai t on d.id\_depou = t.id\_depou

group by d.nume

having count(

case

when t.numar\_vagon in (

select t2.numar\_vagon

from tramvai t2

join alimenteaza a on t2.numar\_vagon = a.numar\_vagon

join linie l on a.numar\_linie = l.numar\_linie

where l.necesita\_bidirectionale = 'DA'

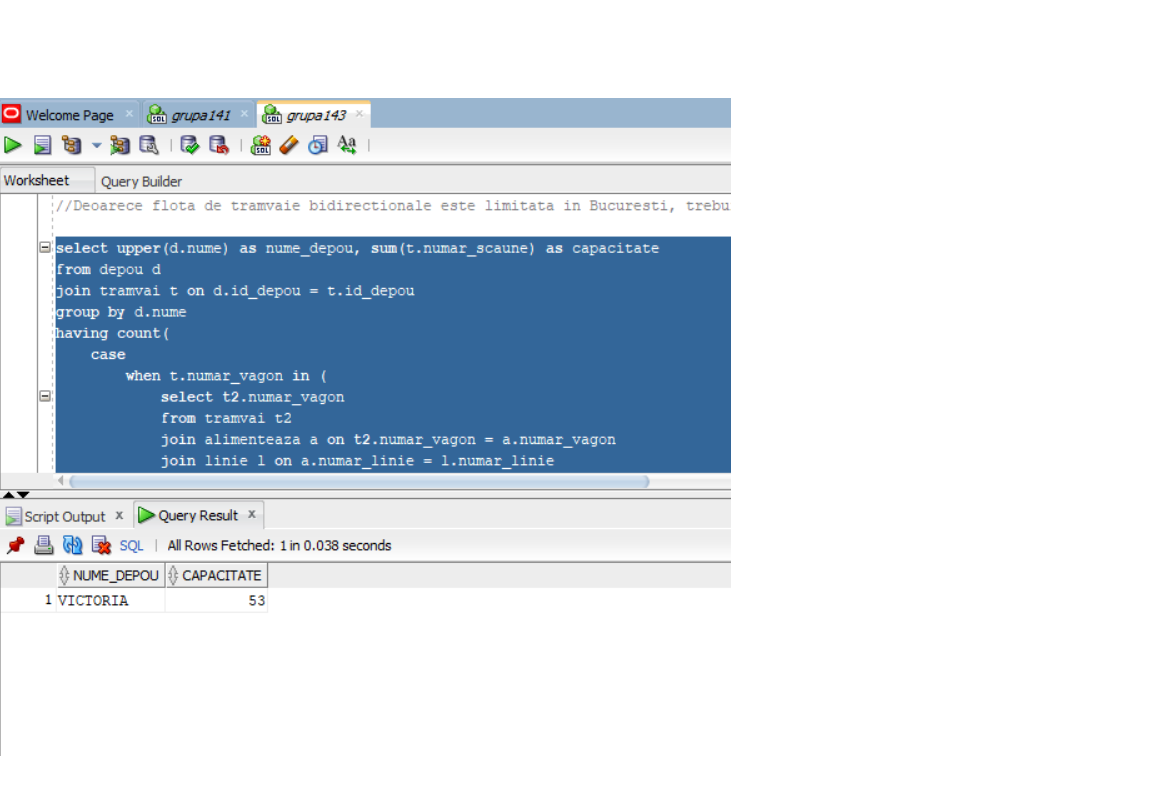
)

then 1

else null

end

) > 0;



//Sa se afiseze toate statiile acoperite de tramvaiele ce gareaza in depoul Militari

//cerinta a), d)

select distinct(s.nume\_statie), s.sector

from statie s join deserveste d on s.nume\_statie = d.nume\_statie

join linie l on l.numar\_linie = d.numar\_linie

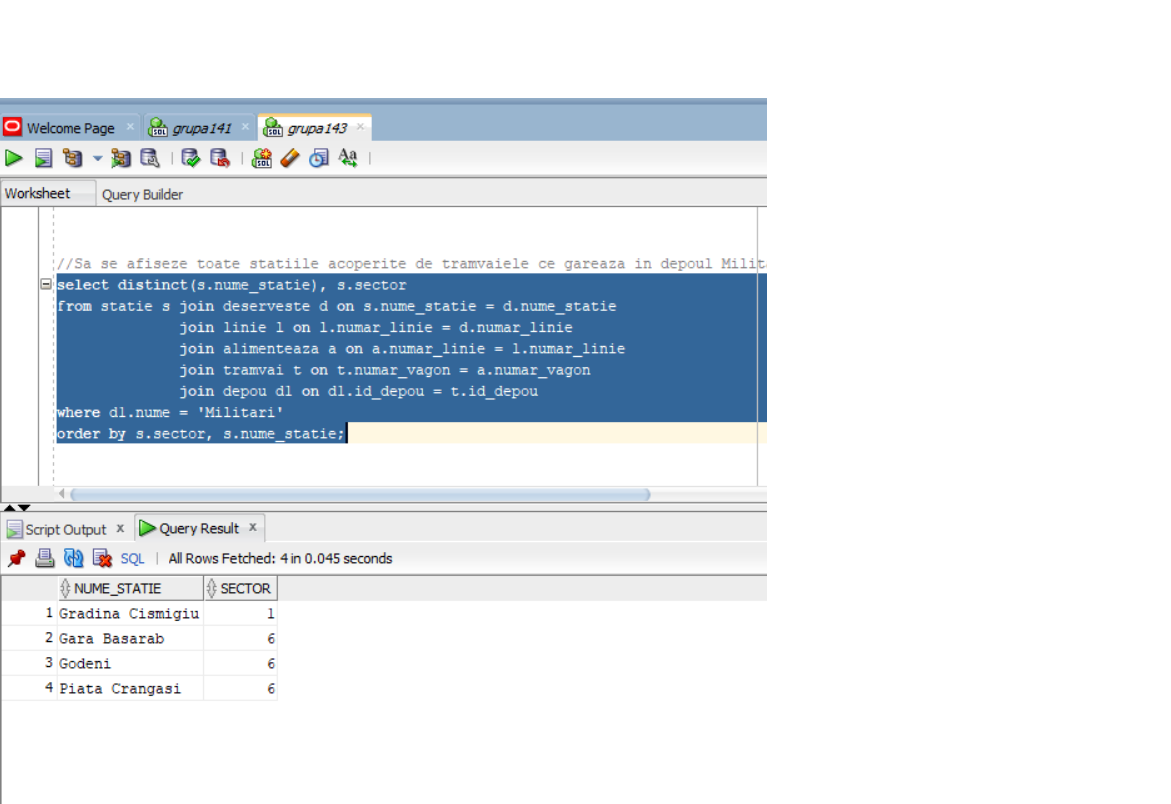
join alimenteaza a on a.numar\_linie = l.numar\_linie

join tramvai t on t.numar\_vagon = a.numar\_vagon

join depou d1 on d1.id\_depou = t.id\_depou

where d1.nume = 'Militari'

order by s.sector, s.nume\_statie;



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

i)//Sa se mute tramvaiele din Colentina si Bucurestii Noi in Militari

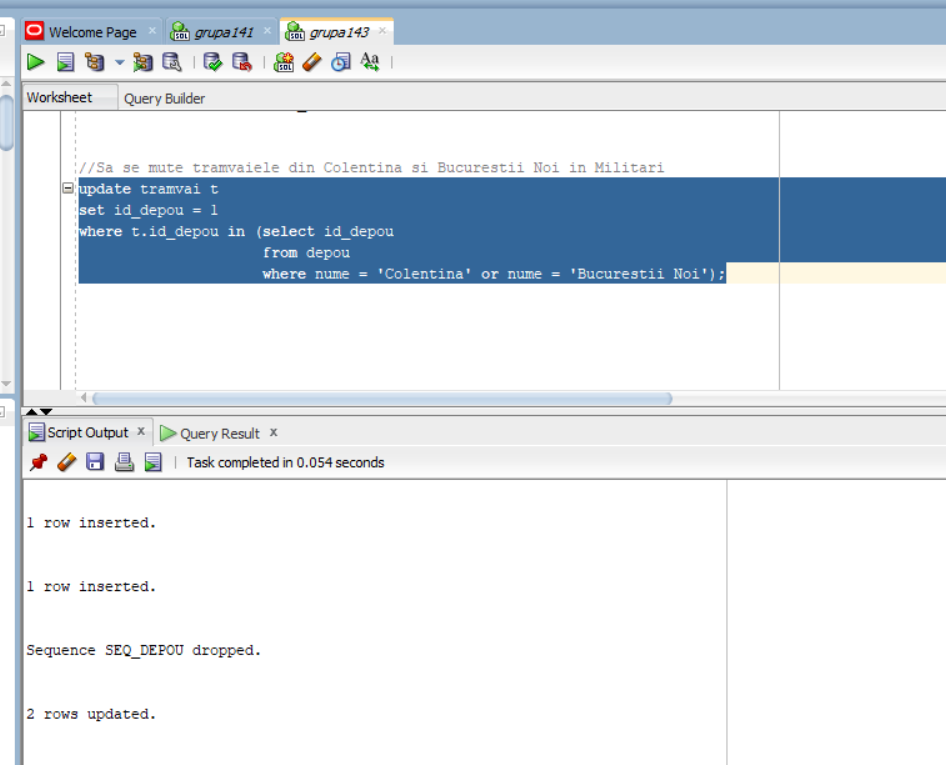
update tramvai t

set id\_depou = 1

where t.id\_depou in (select id\_depou

from depou

where nume = 'Colentina' or nume = 'Bucurestii Noi');



ii)

//Sa se extinda capacitatea depourilor ce alimenteaza statii din sectorul 6 (masura luata in urma cresterii populatiei)

update depou d

set capacitate = capacitate + 10

where exists (

select \*

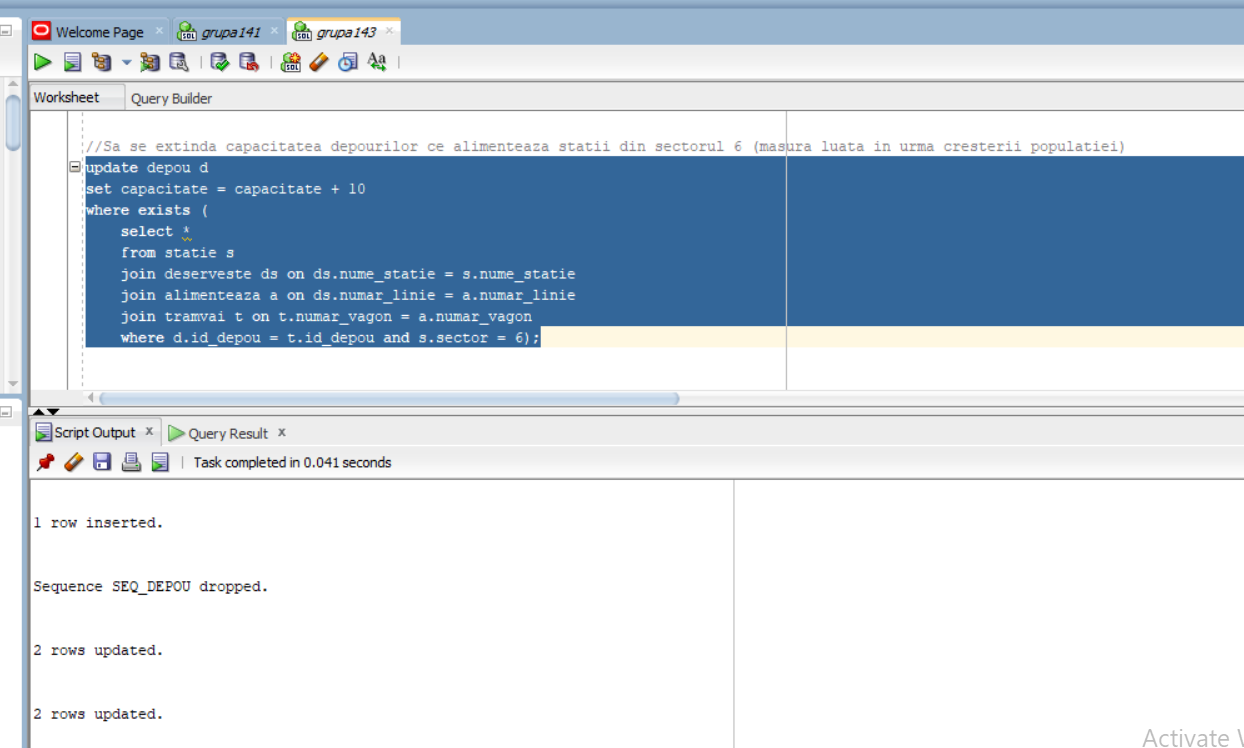
from statie s

join deserveste ds on ds.nume\_statie = s.nume\_statie

join alimenteaza a on ds.numar\_linie = a.numar\_linie

join tramvai t on t.numar\_vagon = a.numar\_vagon

where d.id\_depou = t.id\_depou and s.sector = 6);



iii)//Linia 32 este in lucru si are nevoie de tramvaie bidirectionale. Sa se mute toate tramvaiele bidirectionale pe linia 32

select t.numar\_vagon, t.model\_vagon, l.numar\_linie

from tramvai t

join alimenteaza a on t.numar\_vagon = a.numar\_vagon

join linie l on a.numar\_linie = l.numar\_linie;

update alimenteaza a

set numar\_linie = 32

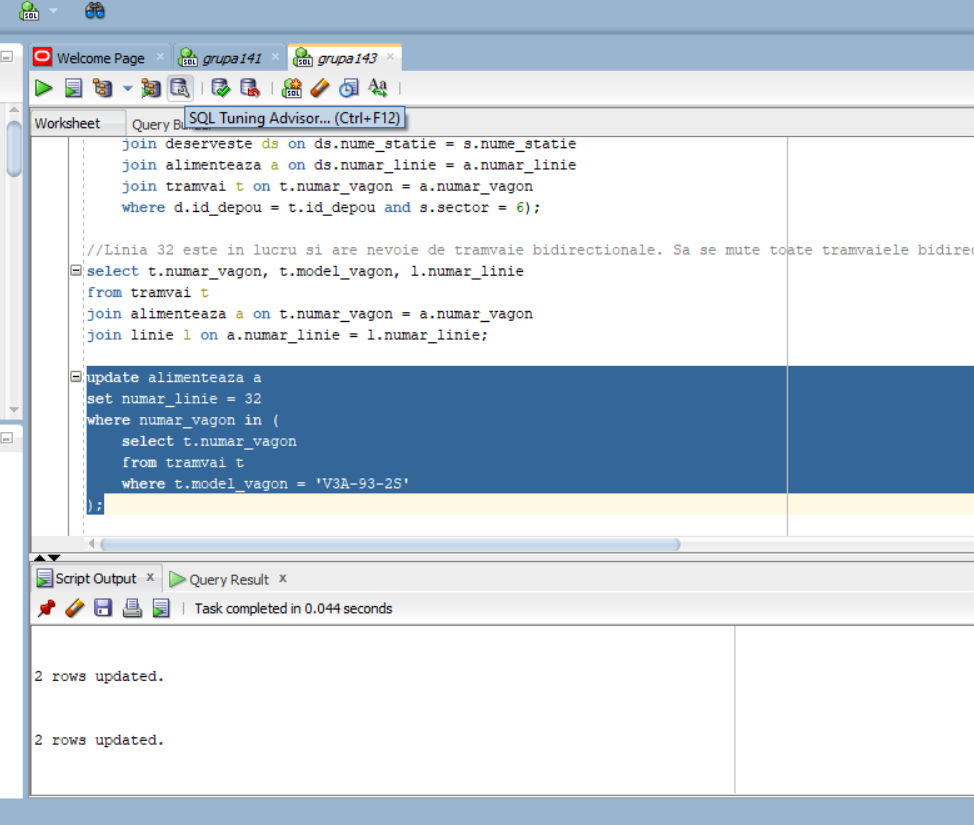
where numar\_vagon in (

select t.numar\_vagon

from tramvai t

where t.model\_vagon = 'V3A-93-2S'

);



iv)

//Avand in vedere vechimea acestora, sa se scoata din circulatie toate tramvaiele cu modelul Tatra T4R

delete from Alimenteaza

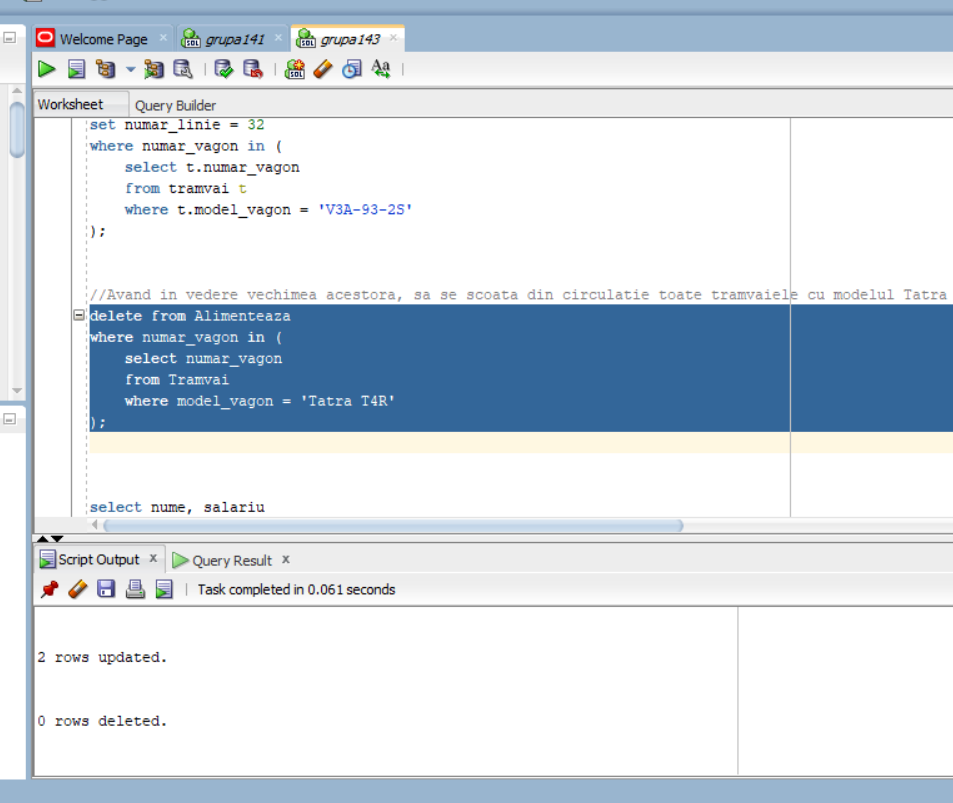
where numar\_vagon in (

select numar\_vagon

from Tramvai

where model\_vagon = 'Tatra T4R'

);



v)//Deoarece linia 44 nu mai este necesara, sa se dea afara toti vatmanii ce operau tramvaie pe aceasta linie

delete from vatman

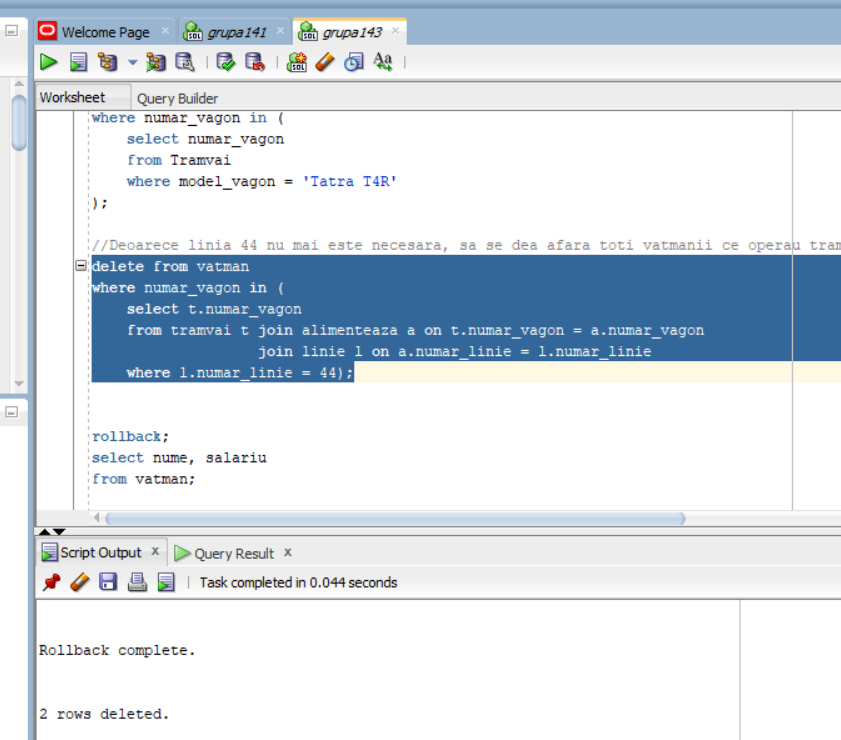
where numar\_vagon in (

select t.numar\_vagon

from tramvai t join alimenteaza a on t.numar\_vagon = a.numar\_vagon

join linie l on a.numar\_linie = l.numar\_linie

where l.numar\_linie = 44);



vi)//Vagonul nr. 77 a avut un incendiu major din cauza unei defectiuni care trebuia prevenita de seful de depou. Sa fie dat afara seful depoului ce gareaza acest vagon.

DELETE FROM SefDepou

WHERE id\_depou IN (

SELECT id\_depou

FROM Tramvai

WHERE numar\_vagon = 77

);

