raul.dumitrache17@gmail.com

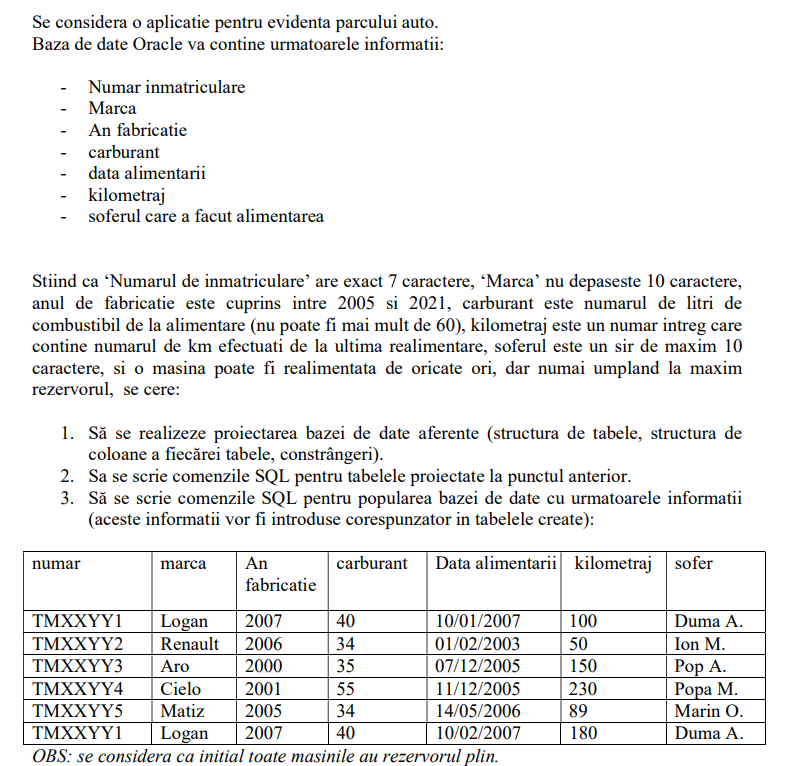
Universitatea politehnică timișoara

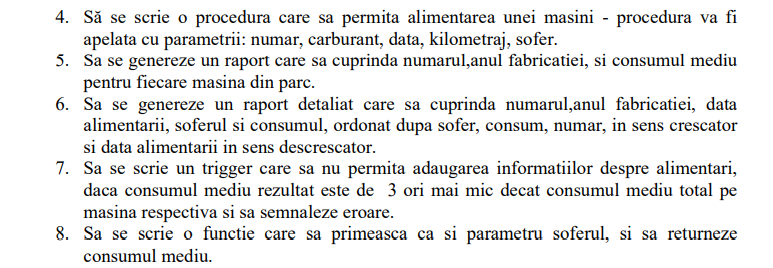
Disciplină: Baze de date

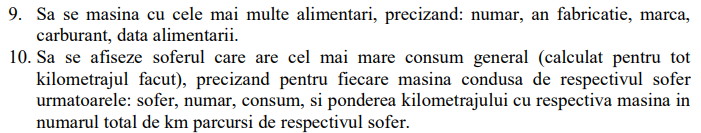
Proiect 5

Raul-Petrișor DUMITRACHE

Subgrupa 2.2







**REZOLVARE:**

**1. Să se realizeze proiectarea bazei de date aferente (structura de tabele, structura de coloane a fiecărei tabele, constrângeri).**

Datele sunt structurate pe 2 tabele, **masini** şi **alimentari**. Tabela **masini** conţine câmpurile: numar(PRIMARY KEY), marca şi anul fabricaţiei. Tabela **alimentari** conţine câmpurile: numar, data\_alimentării, carburant, kilometraj şi sofer. Cele 2 tabele sunt relaţionate prin câmpul număr.

alimentari

masini

- numar(CHAR(7) PRIMARY KEY)

- marca(VACHAR(10) NOT NULL)

-an\_fabricatie(INTEGER NOT NULL)

- numar(CHAR(7) NOT NULL)

- data\_alimentarii (DATE NOT NULL)

-carburant(INTEGER DEFAULT 60 )

-kilometraj(INTEGER NOT NULL)

- sofer(VARCHAR(10) NOT NULL)

REF

**2. Sa se scrie comenzile SQL pentru tabelele proiectate la punctul anterior.**

**CREATE** **TABLE** masini (

numar **CHAR**(7) **PRIMARY** **KEY** **CHECK**(**length**(numar)=7),

marca **VARCHAR**(10) **NOT** **NULL**,

an\_fabricatie **INTEGER** **NOT** **NULL** **CHECK**(an\_fabricatie **BETWEEN** 2005 **AND** 2021)

);

**CREATE** **TABLE** alimentari (

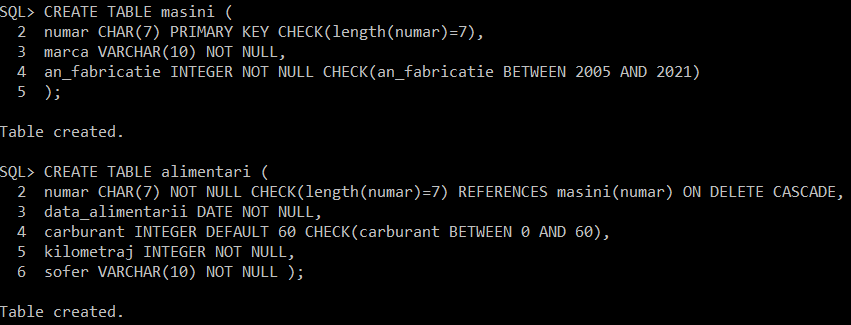
numar **CHAR**(7) **NOT** **NULL** **CHECK**(**length**(numar)=7) **REFERENCES** masini(numar) **ON** **DELETE** **CASCADE**,

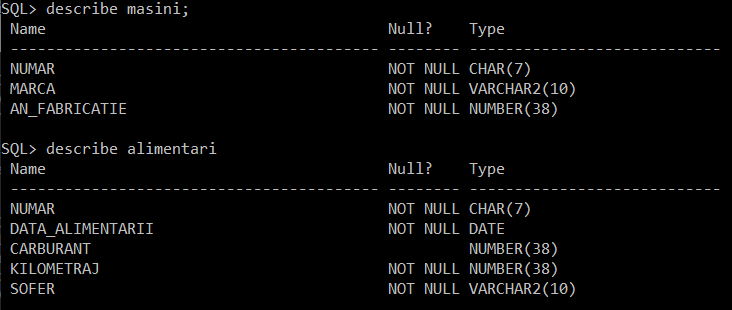
data\_alimentarii **DATE** **NOT** **NULL**,

carburant **INTEGER** **DEFAULT** 60 **CHECK**(carburant **BETWEEN** 0 **AND** 60),

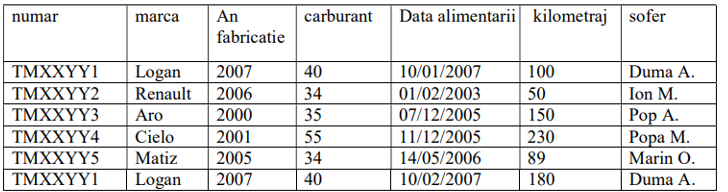
kilometraj **INTEGER** **NOT** **NULL**,

sofer **VARCHAR**(10) **NOT** **NULL** );





**3. Să se scrie comenzile SQL pentru popularea bazei de date cu urmatoarele informatii (aceste informatii vor fi introduse corespunzator in tabelele create):**

****

Se setează pe automat comanda de COMMIT pentru a se salva automat datele după fiecare inserare, apoi se inserează pe rând datele cu comanda INSERT.

**SET** **AUTOCOMMIT** **ON**;

**INSERT** **INTO** masini **VALUES**('TMXXYY1', 'Logan', 2007);

**INSERT** **INTO** masini **VALUES**('TMXXYY2', 'Renault', 2006);

**INSERT** **INTO** masini **VALUES**('TMXXYY3', 'Aro', 2000);

**INSERT** **INTO** masini **VALUES**('TMXXYY4', 'Cielo', 2001);

**INSERT** **INTO** masini **VALUES**('TMXXYY5', 'Matiz', 2005);

**INSERT** **INTO** alimentari **VALUES**('TMXXYY1', '10.01.2007',40, 100, 'Duma A.');

**INSERT** **INTO** alimentari **VALUES**('TMXXYY2', '01.02.2003',34, 50, 'Ion M.' );

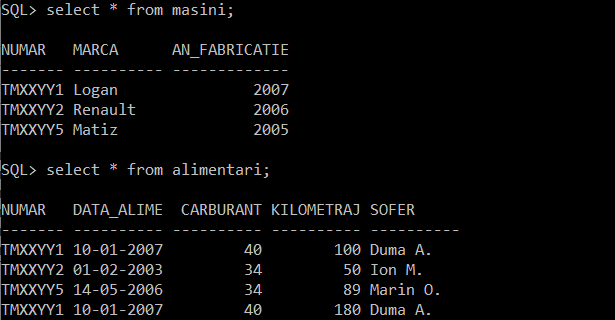
**INSERT** **INTO** alimentari **VALUES**('TMXXYY3', '07.12.2005',5, 150, 'Pop A.' );

**INSERT** **INTO** alimentari **VALUES**('TMXXYY4', '11.12.2005',55, 230,'Popa M.' );

**INSERT** **INTO** alimentari **VALUES**('TMXXYY5', '14.05.2006',34, 89,'Marin O.' );

**INSERT** **INTO** alimentari **VALUES**('TMXXYY1', '10.01.2007',40, 180,'Duma A.' );

După cum se poate observa în captura de ecrand de mai jos, maşinile cu marca ARO şi Cielo, precum şi alimentările pentru acestea nu au fost introduse, deoarece anul de fabicaţie nu se încadrează în intervalul 2005-2021, încalcănd astfel constrangerea CHECK din tabela maşini, fiindcă tabelele sunt relaţionate prin câmpul numar, daca acesta nu există în tabela maşini, nu se pot introduce nici alimentări pentru maşina cu numărul respectiv.



**4. Să se scrie o procedura care sa permita alimentarea unei masini - procedura va fi apelata cu parametrii: numar, carburant, data, kilometraj, sofer.**

Prodedura alimentare insereaza datele primite ca şi parametrii în tabela alimentări, însa mai întai se verifica dacă numărul maşinii se află în tabela părinte, dacă carburantul introdus depaşeste capacitatea rezervorului(maxim 60 litri) şi dacă numărul de kilometri nu este negativ, altfel se afişează un mesaj corespunzator de atenţionare.

Pentru a verifica dacă numărul e în tabela părinte s-a numărat de câte ori acesta apare în tabelă, dacă count(numar) returnează 0, înseamnă ca numărul nu se află în tabelă, deci nu se poate alimenta maşina care are acel număr.

**CREATE** **OR** **REPLACE** **PROCEDURE** alimentare(nr **CHAR**, carb **INTEGER**, dat **DATE**, km **INTEGER**, sof **VARCHAR**)

**AS**

e\_carb **EXCEPTION**;

e\_km **EXCEPTION**;

e\_nr **EXCEPTION**;

flg **INTEGER**;

**BEGIN**

flg:=0;

**SELECT** **count**(numar) **INTO** flg **FROM** masini **WHERE** numar=nr;

**IF** flg=0 **THEN**

**RAISE** e\_nr;

**END IF**;

**IF** carb>60 **THEN**

**RAISE** e\_carb;

**END IF**;

**IF** km < 0 **THEN**

**RAISE** e\_km;

**END IF**;

**INSERT** **INTO** alimentari **VALUES**(nr, dat, carb, km, sof);

**EXCEPTION**

**WHEN** e\_nr **THEN**

**DBMS\_OUTPUT**.**PUT\_LINE**('Masina nu exista in tabela masini!');

**WHEN** e\_carb **THEN**

**DBMS\_OUTPUT**.**PUT\_LINE**('S-a depasit capacitatea rezervorului! (60 litri)');

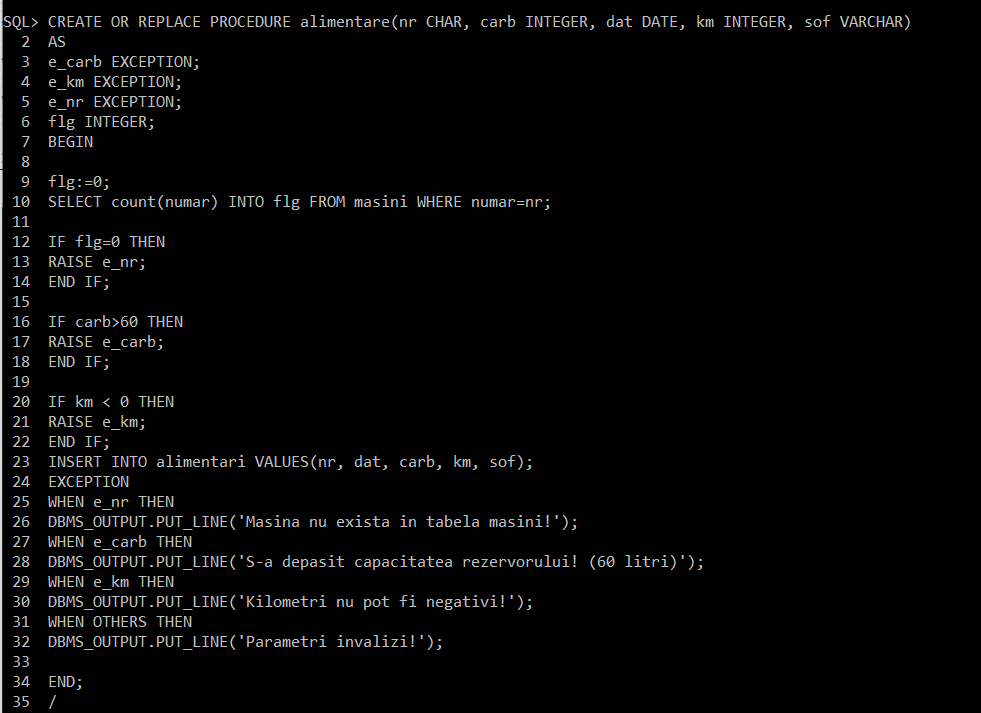
**WHEN** e\_km **THEN**

**DBMS\_OUTPUT**.**PUT\_LINE**('Kilometri nu pot fi negativi!');

**WHEN OTHERS THEN**

**DBMS\_OUTPUT**.**PUT\_LINE**('Parametri invalizi!');

**END**;

****









**5. Sa se genereze un raport care sa cuprinda numarul,anul fabricatiei, si consumul mediu pentru fiecare masina din parc.**

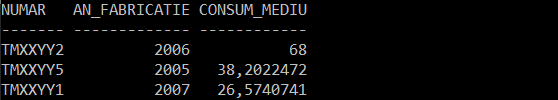
Din tabela maşini se afişează numarul şi anul de fabricaţie, iar pentru consumul mediu se foloşeşte funcţia de grup AVG cu parametrul carburant\*100/kilometraj având aliasul consum\_mediu. Se grupează dupa număr, apoi după anul de fabricaţie.

**SELECT** M.numar, M.an\_fabricatie, **AVG**(**A**.carburant\*100/**A**.kilometraj) **AS** consum\_mediu

**FROM** masini M, alimentari **A**

**WHERE** M.numar=**A**.numar

**GROUP** **BY** M.numar, M.an\_fabricatie;



**6. Sa se genereze un raport detaliat care sa cuprinda numarul,anul fabricatiei, data alimentarii, soferul si consumul, ordonat dupa sofer, consum, numar, în sens crescator si data alimentarii în sens descrescator.**

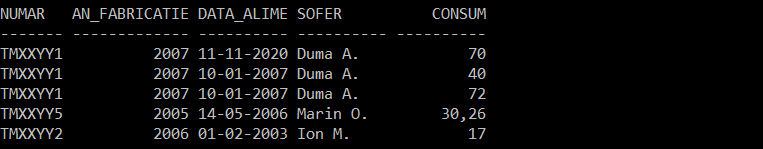
Se selecteaza câmpurile necesare din cele 2 tabele, consumul se calculează cu formula carburant\*100/kilometraj, folosită şi la punctul anterior. Nu am folosit clauza GROUP BY deoarece se doreşte consumul pentru fiecare alimentare în parte.

**SELECT** **A**.numar , M.an\_fabricatie, **A**.data\_alimentarii, **A**.sofer, **A**.carburant/100\***A**.kilometraj **AS** consum

**FROM** masini M, alimentari **A**

**WHERE** M.numar=**A**.numar

**ORDER** **BY** **A**.data\_alimentarii **DESC**, **A**.sofer, consum, **A**.numar;



**7. Sa se scrie un trigger care sa nu permita adaugarea informatiilor despre alimentari, daca consumul mediu rezultat este de 3 ori mai mic decat consumul mediu total pe masina respectiva si sa semnaleze eroare.**

Trigger-ul este de tip BEFORE, definit la nivel de linie, variabilele interne cons\_med si cons\_med\_rez se calculează folosind formula carburant\*100/kilometraj, în cadrul unor comenzi SELECT INTO.

**CREATE** **OR** **REPLACE** **TRIGGER** restrictie\_alimentare

**BEFORE** **INSERT** **ON** alimentari

**FOR** **EACH** **ROW**

**DECLARE**

cons\_med **REAL**;

cons\_med\_rez **REAL**;

e **EXCEPTION**;

**BEGIN**

--consumul mediu inainte de inserare

**SELECT** **SUM**(**A**.carburant)\*100/**SUM**(**A**.kilometraj)

**INTO** cons\_med

**FROM** alimentari **A**

**WHERE** :NEW.numar=**A**.numar;

--consumul mediu rezultat dupa inserare

**SELECT** (**SUM**(**A**.carburant)+:NEW.carburant)\*100/(**SUM**(**A**.kilometraj)+:NEW.kilometraj)

**INTO** cons\_med\_rez

**FROM** alimentari **A**

**WHERE** :NEW.numar=**A**.numar;

--daca consumul mediu rezultat este de 3 ori mai mic decat consumul mediu

--se semnaleaza eroare si se anulează comanda de insert

**IF** 3\*cons\_med\_rez<cons\_med **THEN**

**RAISE** e;

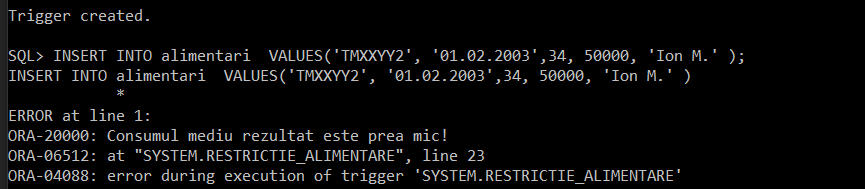
**END IF**;

**EXCEPTION**

**WHEN** e **THEN**

**RAISE\_APPLICATION\_ERROR**(-20000, 'Consumul mediu rezultat este prea mic! ');

**END**;



**8. Sa se scrie o functie care sa primeasca ca si parametru soferul, si sa returneze consumul mediu.**

În variabila interna a, se pune numarul de apariţii ale soferului dat ca parametru funcţiei, prin comanda INSERT ... INTO, daca a este 0 după efectuarea comenzii, şoferul nu se află in tabela şi se ridică o exceptie, funcţia va returna -1. Daca şoferul există în tabelă, se calculează consumul mediu cu formula carburant\*100/kilometraj.

**CREATE** **OR** **REPLACE** **FUNCTION** consum(sof **VARCHAR**)

**RETURN** **REAL** **AS**

a **INTEGER**;

e **EXCEPTION**;

cons\_med **REAL**;

**BEGIN**

--se verifica daca exista soferul dat ca parametru

**a**:=0;

**SELECT** **COUNT**(sofer) **INTO** **a** **FROM** alimentari **WHERE** sofer=sof;

**IF** **a**=0 **THEN**

**RAISE** e;

**END IF**;

--calcul consum mediu pentru sofer

**SELECT** **SUM**(**A**.carburant)\*100/**SUM**(**A**.kilometraj)

**INTO** cons\_med

**FROM** alimentari **A**

**WHERE** **A**.sofer=sof;

**RETURN** cons\_med;

**EXCEPTION**

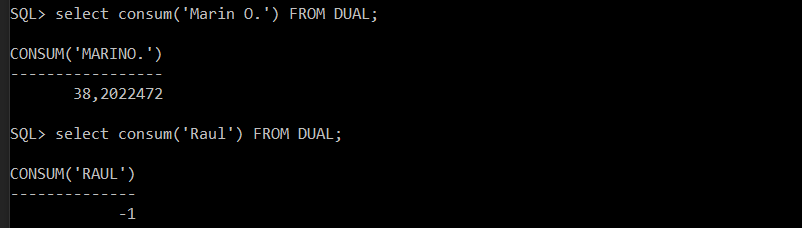
**WHEN** e **THEN**

**RETURN** -1;

**WHEN** OTHERS **THEN**

**RETURN** -1;

**END**;



**9. Sa se afişeze masina cu cele mai multe alimentari, precizand: numar, an fabricatie, marca, carburant, data alimentarii.**

Am creat 2 vederi, vedere1, prin care se pot accesa datele referitoare la numar, an\_fabricatie, marcă, carburant, data\_alimenrării și vedere2, care conține numarul și de câte ori apare acesta în tabela alimentări, adica de cate ori a fost alimentată mașina.

După crearea vederilor, are loc o interogare care extrage toate linile din vedere1, care au numarul egal cu numarul din vedere2 care are cele mai multe apariții in tabela alimentări. În cazul în care 2 mașini au același numar de alimentări vor fi afișate ambele.

**CREATE OR REPLACE VIEW** vedere1 **AS** (

**SELECT** M.numar, M.an\_fabricatie, M.marca, **A**.carburant, **A**.data\_alimentarii

**FROM** masini M, alimentari **A**

**WHERE** **A**.numar=M.numar );

**CREATE** **OR** **REPLACE** **VIEW** vedere2 **AS** (

**SELECT** **A**.numar, **count**(**A**.numar) nr

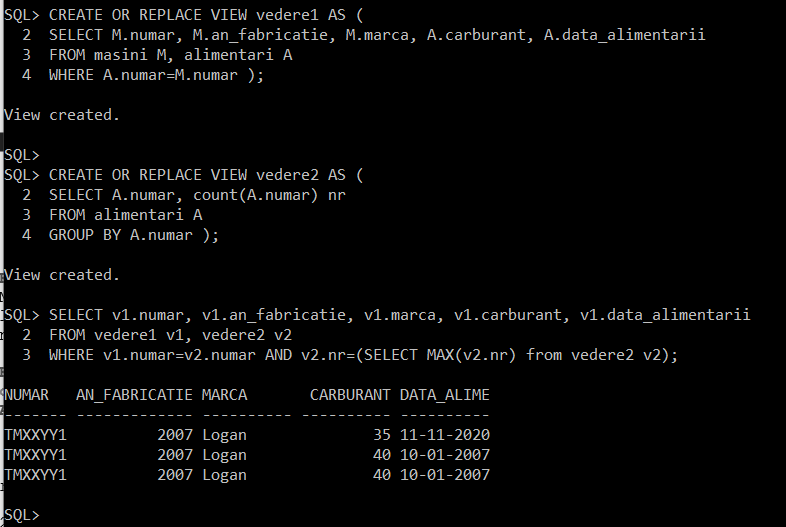
**FROM** alimentari **A**

**GROUP** **BY** **A**.numar );

**SELECT** v1.numar, v1.an\_fabricatie, v1.marca, v1.carburant, v1.data\_alimentarii

**FROM** vedere1 v1, vedere2 v2

**WHERE** v1.numar=v2.numar **AND** v2.nr=(**SELECT** **MAX**(v2.nr) **from** vedere2 v2);



**10. Sa se afiseze soferul care are cel mai mare consum general (calculat pentru tot kilometrajul facut), precizand pentru fiecare masina condusa de respectivul sofer urmatoarele: sofer, numar, consum, si ponderea kilometrajului cu respectiva masina in numarul total de km parcursi de respectivul sofer.**

Am creeat 3 vederi:

Vedere3 - conţine soferul, numarul, consumul şi kilometrajul şoferilor pe fiecare maşina

Vedere4 – conţine şoferul şi consumul mediu general, ordonate descrecător (vederea e creeata cu date dintr-o subinterogare în care am folosit clauza ORDER BY ), astfel încât prima linie din vedere sa conţina datele despre şoferul cu cel mai mare consum general.

Vedere5 – conţine soferul şi kilometrajul total (kilometrajul şoferului pe toate maşinile)

Rezultatul se obţine dintr-o interogare pe cele 3 vederi, se afişează toate randurile din vedere3 care îndeplinesc condiţia ca şoferul să fie egal cu şoferul din prima linie din vederea4, adica şoferul cu cel mai mare consum general. Ponderea (în procente) se calculează cu formula km\_masina\*100/km\_total, iar km\_total pentru fiecare şofer se găsec în vedere5.

**CREATE** **OR** **REPLACE** **VIEW** vedere3 **AS** (

**SELECT** sofer , numar , **SUM**(carburant)\*100/**SUM**(kilometraj) consum\_masina, **SUM**(kilometraj) km\_masina

**FROM** alimentari **GROUP** **BY** sofer , numar);

**CREATE** **OR** **REPLACE** **VIEW** vedere4 **AS** **SELECT** \* **FROM** (

**SELECT** sofer, **SUM**(carburant)\*100/**SUM**(kilometraj) consum\_mediu

**FROM** alimentari

**GROUP** **BY** sofer

**ORDER** **BY** consum\_mediu **DESC** );

**CREATE** **OR** **REPLACE** **VIEW** vedere5 **AS**(

**SELECT** sofer,**SUM**(kilometraj) km\_total

**FROM** alimentari

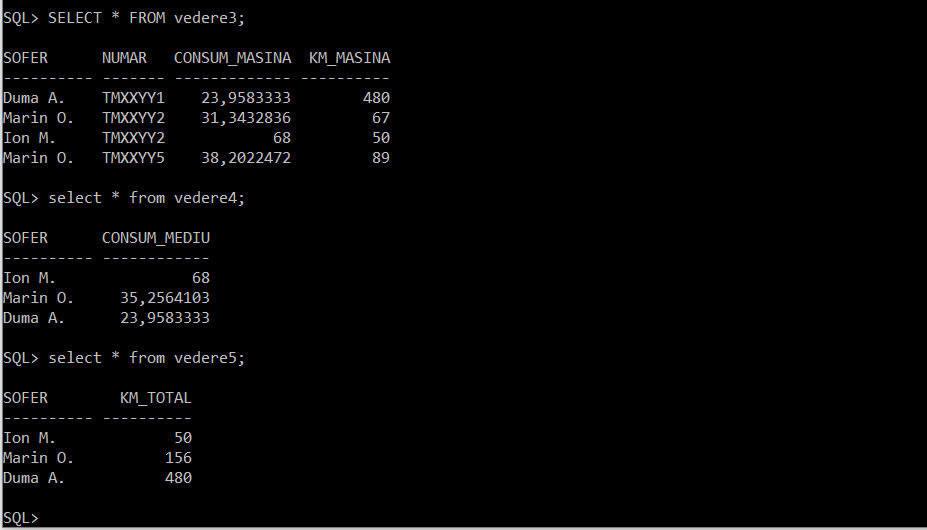
**GROUP** **BY** sofer);

**SELECT** v3.sofer, v3.numar, v3.consum\_masina, v3.km\_masina\*100/v5.km\_total Pondere

**FROM** vedere3 v3, vedere5 v5

**WHERE** v3.sofer=v5.sofer **AND** v3.sofer=(**SELECT** v4.sofer **FROM** vedere4 v4 **WHERE** **ROWNUM**=1);

Conţinutul celor 3 vederi:

****

Rezultatul interogării pe cele 3 vederi:

