

Academia de Studii Economice Bucuresti
Facultatea de Cibernetica, Statistica si Informatica Economica

Sisteme de Gestiune a Bazelor de Date

Proiect

Student: Arosoae Georgiana
Grupa: 1048C
An: II
Profesor: Andreea Alexandra Cirnaru

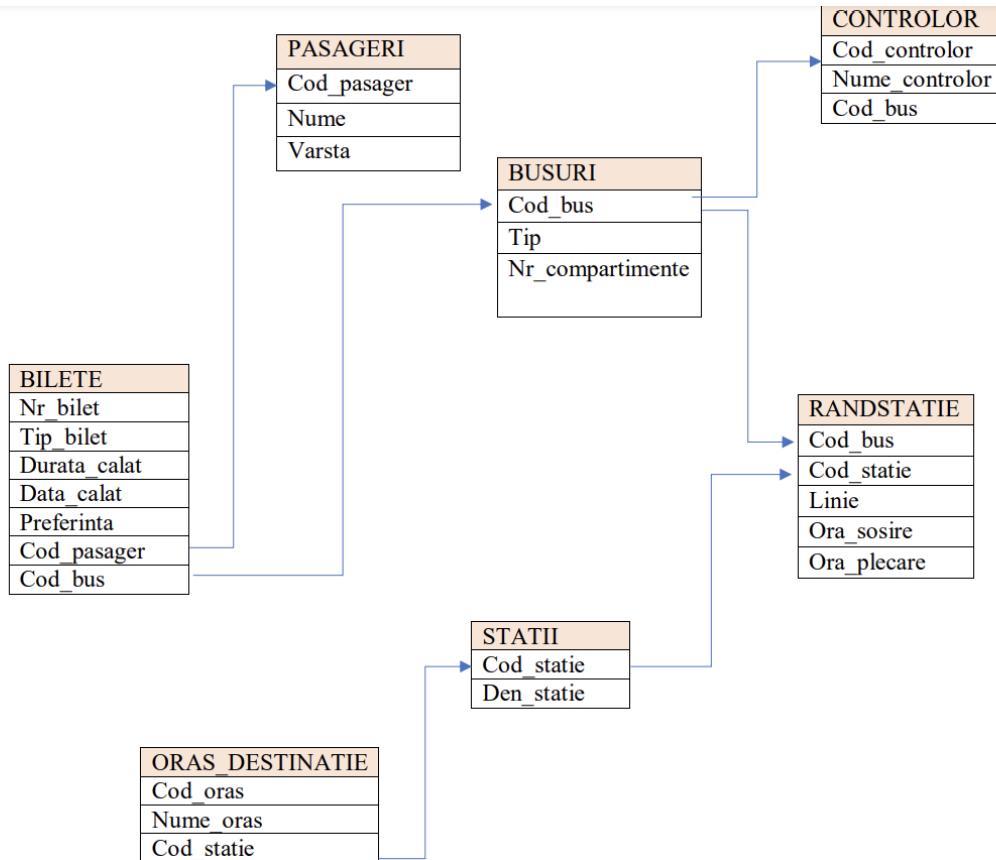
Cerinte:

- Crearea unei baze de date dintr-un domeniu ales de voi care sa contine cel putin 6 tabele, fiecare tabela continand cel putin 10 inregistrari.
- Descrierea problemei și prezentarea schemei conceptuale a tabelelor
- Structuri alternative și repetitive (**minim 4**)
- Tratarea excepțiilor (**minim 2 implicate, 2 explicite**)
- Gestionarea cursorilor: implicați și expliciti (cu și fără parametri) (**minim 2 impliciti, minim 2 expliciti cu parametru, minim 3 expliciti fara parametru**)
- Funcții, proceduri, includerea acestora în pachete (**minim 3 funcții, 3 proceduri și un pachet**)
- Declanșatori (**minim 2**)
- Crearea unei aplicații simple in **Oracle Application Express (APEX)**

A.Descrierea problemei și prezentarea schemei conceptuale a tabelelor.

Proiectul presupune implementarea unei baze de date pentru o companie de transport național. Ea este responsabilă cu gestionarea circulației diverselor autobuze, fiecare oferind mai multe grade de confort.

Structura bazei de date va include următoarele tabele: "PASAGERI" (cu informații despre cod, nume și vârstă a pasagerilor), "BILETE" (detalii despre numărul biletului, tipul acestuia, durata călătoriei, preț, preferință, codul pasagerului și codul autobuzului asociat), "BUSURI" (cu codul autobuzului, tipul și numărul de compartimente), "CONTROLORI" (informații despre codul controlorului, nume și codul autobuzului alocat), "STATII" (descriere a codului stației, denumirea stației și linii) și "ORAS_DESTINATIE" (cu codul orașului, nume și codul stației asociate). În plus, se va adăuga tabela "RANDSTATIE" pentru a facilita corelarea între tabelele "BUSURI" și "STATII", având în vedere că mai multe autobuze tranzitează multiple stații.



B. Crearea unei baze de date dintr-un domeniu ales care sa contine cel putin 6 tabele, fiecare tabela continand cel putin 10 inregistrari.

```

CREATE TABLE PASAGERI(
cod_PASAGERI number(4) CONSTRAINT pk_PASAGERI PRIMARY KEY,
nume varchar2(30),
varsta number(3));
    
```

```

CREATE TABLE BUSURI(
cod_bus number(3) CONSTRAINT pk_busuri PRIMARY KEY,
tip varchar2(10),
nr_compartimente number(2) CONSTRAINT c_compartimente CHECK (nr_compartimente
BETWEEN 1 AND 10));
    
```

```

CREATE TABLE CONTROLOR(
cod_controlor number CONSTRAINT pk_CONTROLOR PRIMARY KEY,
nume_controlor varchar2(50),
cod_bus number,
CONSTRAINT fk_BUS FOREIGN KEY (cod_bus) REFERENCES busuri(cod_bus));
    
```

```

CREATE TABLE BILETE(
nr_bilet number(4) CONSTRAINT pk_BILETE PRIMARY KEY,
tip_bilet varchar2(10) NOT NULL,
durata_calat number(2),
data_calat date, pret_bilet number(10),
preferinta number(1),
cod_PASAGERI number(4),
cod_bus number(3),
CONSTRAINT fk_PASAGERI FOREIGN KEY (cod_PASAGERI) REFERENCES
PASAGERI(cod_PASAGERI),
CONSTRAINT fk_busuri FOREIGN KEY (cod_bus) REFERENCES BUSURI(cod_bus));

```

```

CREATE TABLE STATII(
cod_statie number(3) CONSTRAINT pk_STATII PRIMARY KEY,
den_statie varchar2(15) NOT NULL,
linii number(2));

```

```

CREATE TABLE ORAS_DESTINATIE(
cod_oras number CONSTRAINT pk_ORASE PRIMARY KEY,
nume_oras varchar2(50),
cod_statie number,
CONSTRAINT fk_statii foreign key (cod_statie) references statii(cod_statie));

```

```

CREATE TABLE RANDSTATIE(
cod_bus number(3),
cod_statie number(3),
linie number(2),
ora_sosire float CONSTRAINT c_os CHECK ( ora_sosire BETWEEN 0 AND 23),
ora_plecare float CONSTRAINT c_op CHECK ( ora_plecare BETWEEN 0 AND 23),
CONSTRAINT fk_codbus FOREIGN KEY (cod_bus) REFERENCES BUSURI(cod_bus),
CONSTRAINT fk_codstatie FOREIGN KEY (cod_statie) REFERENCES STATII(cod_statie));

```

Populare table:

```

BEGIN
INSERT INTO PASAGERI VALUES(1111, 'Andronic Eliza',20);
INSERT INTO PASAGERI VALUES (2222,'Badulescu Cristina',5);
INSERT INTO PASAGERI VALUES (3333,'Albu Alex',35);
INSERT INTO PASAGERI VALUES (4444,'Barbu Diana',33);
INSERT INTO PASAGERI VALUES (5555,'Ene Andrei',40);
INSERT INTO PASAGERI VALUES (6666,'Godea Andreea',55);
INSERT INTO PASAGERI VALUES (7777,'Avramescu Andreea',28);
INSERT INTO PASAGERI VALUES (8888,'Bauer Robert',14);
INSERT INTO PASAGERI VALUES (9999,'Badea Cristian',20);
INSERT INTO PASAGERI VALUES (1000,'Andrei Cristina',23);

```

```
INSERT INTO BUSURI VALUES (182, 'midi',2);
INSERT INTO BUSURI VALUES (127, 'maxi',1);
INSERT INTO BUSURI VALUES (131, 'delux',5);
INSERT INTO BUSURI VALUES (111, 'mini',8);
INSERT INTO BUSURI VALUES (212, 'midi',9);
INSERT INTO BUSURI VALUES (313, 'maxi',3);
INSERT INTO BUSURI VALUES(353,'maxi',1);
INSERT INTO BUSURI VALUES(123,'mini',1);
INSERT INTO BUSURI VALUES(763,'delux',1);
INSERT INTO BUSURI VALUES(980,'midi',1);
INSERT INTO BUSURI VALUES(116,'midi',1);
```

```
INSERT INTO CONTROLOR VALUES (100,'Ionescu',182);
INSERT INTO CONTROLOR VALUES (200,'Popescu',127);
INSERT INTO CONTROLOR VALUES (300,'Negru',131);
INSERT INTO CONTROLOR VALUES (400,'Avramesu',111);
INSERT INTO CONTROLOR VALUES (500,'Adam',212);
INSERT INTO CONTROLOR VALUES (600,'Badarau',313);
INSERT INTO CONTROLOR VALUES (700,'Baranceanu',353);
INSERT INTO CONTROLOR VALUES (800,'Palamariu',123);
INSERT INTO CONTROLOR VALUES (900,'Matei',763);
INSERT INTO CONTROLOR VALUES (110,'Maxim',980);
INSERT INTO CONTROLOR VALUES (120,'Maxim',116);
```

```
INSERT INTO STATII VALUES (001,'Bucuresti',10);
INSERT INTO STATII VALUES (002,'Ploesti',7);
INSERT INTO STATII VALUES (003,'Pitesti',6);
INSERT INTO STATII VALUES (004,'Petrosani',4);
INSERT INTO STATII VALUES (005,'Craiova',8);
INSERT INTO STATII VALUES (006,'Oradea',6);
INSERT INTO STATII VALUES (007,'Bacau',4);
INSERT INTO STATII VALUES (008,'Brasov',9);
INSERT INTO STATII VALUES (009,'Arad',5);
INSERT INTO STATII VALUES (010,'Targu Jiu',5);
```

```
INSERT INTO ORAS_DESTINATIE VALUES (100,'Brasov',001);
INSERT INTO ORAS_DESTINATIE VALUES (200,'Sibiu',002);
INSERT INTO ORAS_DESTINATIE VALUES (300,'Botosani',003);
INSERT INTO ORAS_DESTINATIE VALUES (400,'Iasi',004);
INSERT INTO ORAS_DESTINATIE VALUES (500,'Timisoara',005);
INSERT INTO ORAS_DESTINATIE VALUES (600,'Suceava',006);
INSERT INTO ORAS_DESTINATIE VALUES (700,'Cluj',007);
INSERT INTO ORAS_DESTINATIE VALUES (800,'Roman',008);
INSERT INTO ORAS_DESTINATIE VALUES (900,'Arad',009);
INSERT INTO ORAS_DESTINATIE VALUES (110,'Oradea',010);
```

```
INSERT INTO BILETE VALUES (111,'student',6,to_date('20-12-07','dd-mm-yy'),45,2,1111,182);
INSERT INTO BILETE VALUES (112,'elev',4,to_date('20-12-07','dd-mm-yy'),30,2,2222,127);
INSERT INTO BILETE VALUES (113,'intreg',8,to_date('18-11-07','dd-mm-yy'),38,1,3333,127);
INSERT INTO BILETE VALUES (114,'intreg',5,to_date('18-11-07','dd-mm-yy'),35,2,4444,182);
INSERT INTO BILETE VALUES (115,'intreg',3,to_date('20-12-07','dd-mm-yy'),40,1,5555,131);
```

```

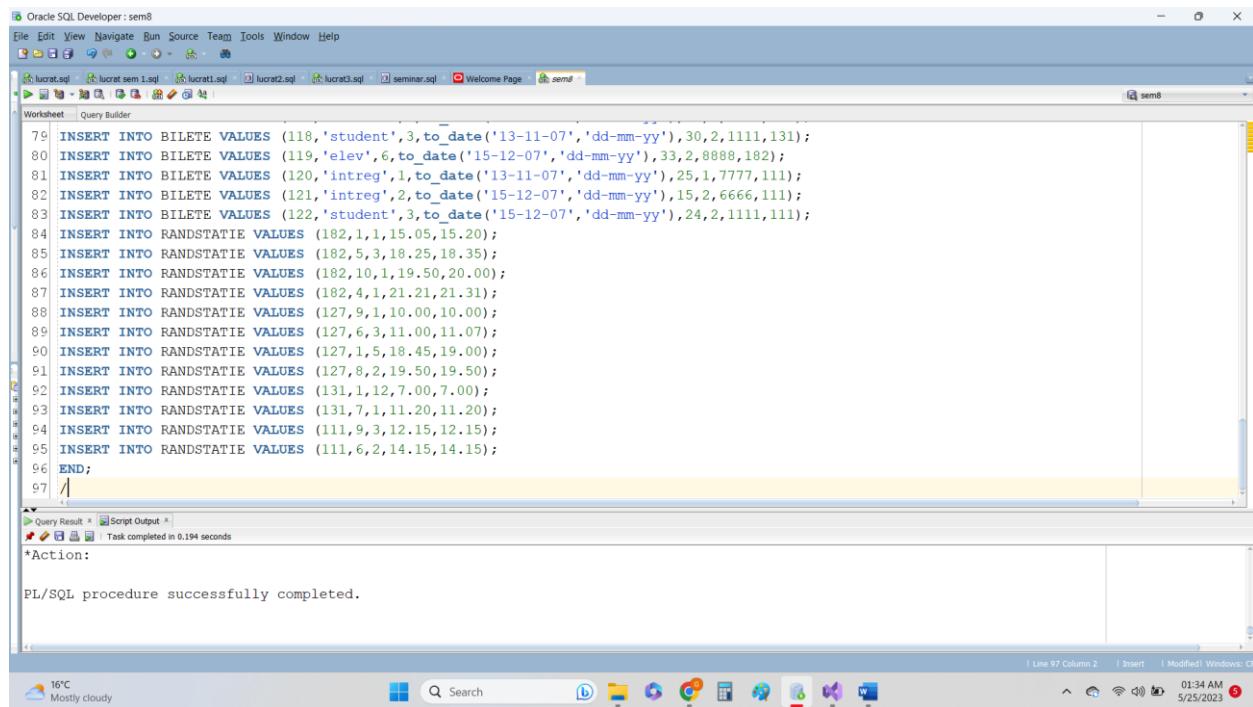
INSERT INTO BILETE VALUES (116,'intreg',10,to_date('13-11-07','dd-mm-yy'),60,1,6666,131);
INSERT INTO BILETE VALUES (117,'student',2,to_date('18-11-07','dd-mm-yy'),20,2,1000,127);
INSERT INTO BILETE VALUES (118,'student',3,to_date('13-11-07','dd-mm-yy'),30,2,1111,131);
INSERT INTO BILETE VALUES (119,'elev',6,to_date('15-12-07','dd-mm-yy'),33,2,8888,182);
INSERT INTO BILETE VALUES (120,'intreg',1,to_date('13-11-07','dd-mm-yy'),25,1,7777,111);
INSERT INTO BILETE VALUES (121,'intreg',2,to_date('15-12-07','dd-mm-yy'),15,2,6666,111);
INSERT INTO BILETE VALUES (122,'student',3,to_date('15-12-07','dd-mm-yy'),24,2,1111,111);

```

```

INSERT INTO RANDSTATIE VALUES (182,1,1,15.05,15.20);
INSERT INTO RANDSTATIE VALUES (182,5,3,18.25,18.35);
INSERT INTO RANDSTATIE VALUES (182,10,1,19.50,20.00);
INSERT INTO RANDSTATIE VALUES (182,4,1,21.21,21.31);
INSERT INTO RANDSTATIE VALUES (127,9,1,10.00,10.00);
INSERT INTO RANDSTATIE VALUES (127,6,3,11.00,11.07);
INSERT INTO RANDSTATIE VALUES (127,1,5,18.45,19.00);
INSERT INTO RANDSTATIE VALUES (127,8,2,19.50,19.50);
INSERT INTO RANDSTATIE VALUES (131,1,12,7.00,7.00);
INSERT INTO RANDSTATIE VALUES (131,7,1,11.20,11.20);
INSERT INTO RANDSTATIE VALUES (111,9,3,12.15,12.15);
INSERT INTO RANDSTATIE VALUES (111,6,2,14.15,14.15);
END;
/

```



The screenshot shows the Oracle SQL Developer interface with a query editor window titled 'sem8'. The code in the editor is identical to the one provided above. Below the editor, the 'Script Output' tab shows the execution results: 'Task completed in 0.194 seconds' and 'PL/SQL procedure successfully completed.' The system tray at the bottom indicates it's 16°C and mostly cloudy. The status bar shows the date and time as 01:34 AM 5/25/2023.

```

79: INSERT INTO BILETE VALUES (118,'student',3,to_date('13-11-07','dd-mm-yy'),30,2,1111,131);
80: INSERT INTO BILETE VALUES (119,'elev',6,to_date('15-12-07','dd-mm-yy'),33,2,8888,182);
81: INSERT INTO BILETE VALUES (120,'intreg',1,to_date('13-11-07','dd-mm-yy'),25,1,7777,111);
82: INSERT INTO BILETE VALUES (121,'intreg',2,to_date('15-12-07','dd-mm-yy'),15,2,6666,111);
83: INSERT INTO BILETE VALUES (122,'student',3,to_date('15-12-07','dd-mm-yy'),24,2,1111,111);
84: INSERT INTO RANDSTATIE VALUES (182,1,1,15.05,15.20);
85: INSERT INTO RANDSTATIE VALUES (182,5,3,18.25,18.35);
86: INSERT INTO RANDSTATIE VALUES (182,10,1,19.50,20.00);
87: INSERT INTO RANDSTATIE VALUES (182,4,1,21.21,21.31);
88: INSERT INTO RANDSTATIE VALUES (127,9,1,10.00,10.00);
89: INSERT INTO RANDSTATIE VALUES (127,6,3,11.00,11.07);
90: INSERT INTO RANDSTATIE VALUES (127,1,5,18.45,19.00);
91: INSERT INTO RANDSTATIE VALUES (127,8,2,19.50,19.50);
92: INSERT INTO RANDSTATIE VALUES (131,1,12,7.00,7.00);
93: INSERT INTO RANDSTATIE VALUES (131,7,1,11.20,11.20);
94: INSERT INTO RANDSTATIE VALUES (111,9,3,12.15,12.15);
95: INSERT INTO RANDSTATIE VALUES (111,6,2,14.15,14.15);
96: END;
97: /

```

C. Structuri alternative și repetitive

1. Controlorul cu id-ul 400 intra in concediu. Afisati un controlor inlocuitor:

SET SERVEROUTPUT ON

```

Declare
v_nume controlor.nume_controlor%type;
Begin
select nume_controlor into v_nume from controlor where cod_controlor=400;
dbms_output.put_line('Controlorul in concediu este:'||v_nume);
if v_nume='Avramesu' then v_nume:='Adam';
else v_nume:='Necunoscut';
End if;
dbms_output.put_line('Controlorul inlocuit este:'||v_nume);
End;
/

```

```

SET SERVEROUTPUT ON
Declare
v_nume controlor.nume_controlor%type;
Begin
select nume_controlor into v_nume from controlor where cod_controlor=400;
dbms_output.put_line('Controlorul in concediu este:'||v_nume);
if v_nume='Avramesu' then v_nume:='Adam';
else v_nume:='Necunoscut';
End if;
dbms_output.put_line('Controlorul inlocuit este:'||v_nume);
End;
/

```

Controlorul in concediu este:Avramesu
Controlorul inlocuit este:Adam

PL/SQL procedure successfully completed.

2. Pentru biletul cu numarul 119 cresteti de 1,2 ori pretul daca acesta e mai mare de 30 si de 1.5 ori daca e mai mic. Afisati pretul initial si final.

```

SET SERVEROUTPUT ON
Declare
v_pret bilete.pret_bilet%type;
Begin
select pret_bilet into v_pret from bilete where nr_bilet=119;
dbms_output.put_line('pretul initial al biletului este :'||v_pret);
if v_pret <30 then v_pret:= v_pret * 1.5;
else v_pret:=v_pret * 1.2;
End if;
dbms_output.put_line('pretul final al biletului este :'||v_pret);
End;
/

```

```

101 1. Pentru biletul cu numarul 119 cresteti de 1,2 ori pretul daca acesta e mai mare de 30 si de 1,5 ori
102 daca e mai mic. Afisati pretul initial si final.
103
104 SET SERVEROUTPUT ON
105Declare
106 v_pret bilete.pret_bilet%type;
107 Begin
108 select pret_bilet into v_pret from bilete where nr_bilet=119;
109 dbms_output.put_line('pretul initial al biletului este :'||v_pret);
110 if v_pret <30 then v_pret:= v_pret * 1.5;
111 else v_pret:=v_pret * 1.2;
112 End if;
113 dbms_output.put_line('pretul final al biletului este :'||v_pret);
114 End;
115

```

Query Result Script Output Task completed in 0.061 seconds
PL/SQL procedure successfully completed.

3. Daca pasagerii au sub 26 de ani, impozitul reprezinta 15% din valoarea biletului, daca au intre 26 si 35 de ani, impozitul reprezinta 20%, iar daca au peste 35 acesta este de 10%. Calculati impozitul pentru biletul fiecarui calator introdus de la tastatura.

Declare

```

v_pret bilete.pret_bilet%type;
v_varsta pasageri.varsta%type;
v_codp number(4):=&codp;
v_impozit number(4,2);
Begin
select pret_bilet into v_pret from bilete where cod_pasageri=v_codp;
select varsta into v_varsta from pasageri where cod_pasageri=v_codp;
v_impozit:=case
when v_varsta <26 then 0.15
when v_varsta between 26 and 35 then 0.2
else 0.1
End;
v_impozit:=v_impozit * v_pret;
dbms_output.put_line('pentru calatorul cu varsta:'||v_varsta||' impozit este:'||v_impozit);
End;
/

```

```

118 35 de ani impozitul reprez 20%, iar daca au peste 35 acesta este de 10%. Calculati impozitul pentru
119 biletul fiecarui calator introdus de la tastatura.
120#Declare
121 v_pret bilete.pret_bilet%type;
122 v_varsta pasageri.varsta%type;
123 v_codp number(4):=&codp;
124 v_impozit number(4,2);
125 Begin
126 select pret_bilet into v_pret from bilete where cod_pasageri=v_codp;
127 select varsta into v_varsta from pasageri where cod_pasageri=v_codp;
128#v_impozit:=case
129 when v_varsta <26 then 0.15
130 when v_varsta between 26 and 35 then 0.2
131 else 0.1
132 End;
133 v_impozit:=v_impozit * v_pret;
134 dbms_output.put_line('pentru calatorul cu varsta:'||v_varsta||' impozitul este:'||v_impozit);
135 End;

```

Query Result X Script Output X
Task completed in 2.456 seconds

pentru calatorul cu varsta:5 impozitul este:4.5

PL/SQL procedure successfully completed.

4.Calculati valoarea reala a pretului fiecarui bilet citit de la tastatura , stiind ca in functie de tipul biletului (elev, student, intreg), pretul a fost redus cu 30%, 50% respectiv cu 0%.

```

Declare
v_pret bilete.pret_bilet%type;
v_tip bilete.tip_bilet%type;
v_nrbiilet bilete.nr_bilet%type:=&codb;
Begin
select pret_bilet, tip_bilet into v_pret, v_tip from bilete where nr_bilet=v_nrbiilet;
dbms_output.put_line('pretul biletului redus este :'||v_pret);
case
when upper(v_tip)='ELEV' then v_pret := v_pret * 1.3;
when upper(v_tip)='STUDENT' then v_pret := v_pret * 1.5;
else v_pret := v_pret *1;
End case;
dbms_output.put_line('pretul biletului real este :'||v_pret||' deoarece biletul este de tip:'||v_tip);
End;
/

```

```

144
145 Declare
146   v_pret bilet.pret_bilet%type;
147   v_tip bilet.tip_bilet%type;
148   v_nr_bilet bilet.nr_bilet%type:=&codb;
149 Begin
150   select pret_bilet, tip_bilet into v_pret, v_tip from bilet where nr_bilet=v_nr_bilet;
151   dbms_output.put_line('pretul biletului redus este :'||v_pret);
152 End case;
153 when upper(v_tip)='ELEV' then v_pret := v_pret * 1.3;
154 when upper(v_tip)='STUDENT' then v_pret := v_pret * 1.5;
155 else v_pret := v_pret *1;
156 End case;
157 dbms_output.put_line('pretul biletului real este :'||v_pret||' deoarece biletul este de tip:'||v_tip);
158 End;
159 /

```

Query Result: Task completed in 2.179 seconds

```

pretul biletului redus este :30
pretul biletului real este :39 deoarece biletul este de tip:elev

PL/SQL procedure successfully completed.

```

D. Tratarea exceptiilor (minim 2 implice, 2 explice).

1. Afisati persoana cea mai tanara care calatoreste cu busul. Tratati exceptiile ce ar putea aparea.

Declare

v_varsta pasageri.varsta%type;

v_nume pasageri.nume%type;

Begin

select min(varsta) into v_varsta from pasageri;

select nume into v_nume from pasageri

where varsta = v_varsta;

dbms_output.put_line('persoana cea mai tanara se numeste :'||v_nume);

exception

when too_many_rows then

dbms_output.put_line('exista mai multe persoane care au aceasta varsta');

End;

/

```

137 dbms_output.put_line('pretul directului real este : '||v_pret|| ' deoarece directul este de tip: '||v_tip);
158 End;
159 /
160
161 Declare
162 v_varsta pasageri.varsta%type;
163 v_nume pasageri.nume%type;
164 Begin
165 select min(varsta) into v_varsta from pasageri;
166 select nume into v_nume from pasageri
167 where varsta = v_varsta;
168 dbms_output.put_line('persoana cea mai tanara se numeste :'||v_nume);
169 exception
170 when too_many_rows then
171 dbms_output.put_line('exista mai multe persoane care au aceasta varsta');
172 End;
173

```

Query Result Script Output Task completed in 0.07 seconds
PL/SQL procedure successfully completed.

persoana cea mai tanara se numeste :Badulescu Cristina

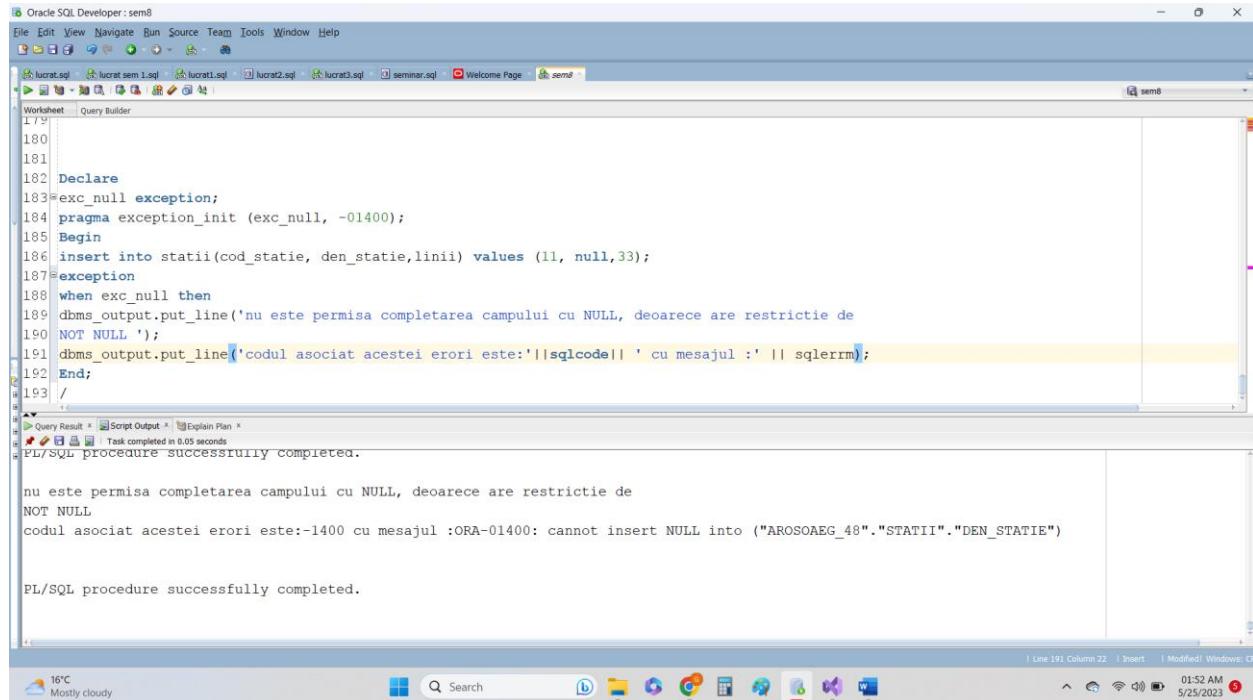
PL/SQL procedure successfully completed.

2. Inserati in tabela STATII inca o statie. Tratati exceptia in cazul in care nu este introdusa denumirea statiei. Folositi-va si de SQLCODE si SQLERRM.

```

Declare
exc_null exception;
pragma exception_init (exc_null, -01400);
Begin
insert into statii(cod_statie, den_statie,linii) values (11, null,33);
exception
when exc_null then
dbms_output.put_line('nu este permisa completarea campului cu NULL, deoarece are restrictie de
NOT NULL ');
dbms_output.put_line('codul asociat acestei erori este:'||sqlcode|| ' cu mesajul :' || sqlerrm);
End;
/

```



```

179
180
181
182 Declare
183 exc_null exception;
184 pragma exception_init (exc_null, -01400);
185 Begin
186 insert into statii(cod_statie, den_statie,linii) values (11, null,33);
187 exception
188 when exc_null then
189 dbms_output.put_line('nu este permisa completarea campului cu NULL, deoarece are restrictie de
190 NOT NULL ');
191 dbms_output.put_line('codul asociat acestei erori este:'||sqlcode|| ' cu mesajul :' || sqlerrm);
192 End;
193 /

```

PL/SQL procedure successfully completed.

nu este permisa completarea campului cu NULL, deoarece are restrictie de
NOT NULL
codul asociat acestei erori este:-1400 cu mesajul :ORA-01400: cannot insert NULL into ("AROSOAE_48"."STATII"."DEN_STATIE")

PL/SQL procedure successfully completed.

3. Afisati tipul biletului cu care calatoreste un pasager introdus de la tastatura. Tratati exceptiile ce ar putea aparea.

```

Declare
v_codp bilete.cod_pasageri%type:=&codp;
v_tip bilete.tip_bilet%type;
Begin
select tip_bilet into v_tip from bilete
where cod_pasageri=v_codp;
dbms_output.put_line('Tipul biletului pasagerului este :'||v_tip);
exception
when no_data_found then
dbms_output.put_line('Nu exista acest pasager sau nu are bilet ');
End;
/

```

```

193 /
194
195 --2. Afisati tipul biletului cu care calatoreste un pasager introdus de la tastatura. Tratati exceptiile
196 --ce ar putea aparea.
197 Declare
198 v_cod_bilet.cod_pasageri%type:=&codp;
199 v_tip_bilet.tip_bilet%type;
200 Begin
201 select tip_bilet into v_tip from bilet
202 where cod_pasageri=v_codp;
203 dbms_output.put_line('Tipul biletului pasagerului este :'||v_tip);
204 exception
205 when no_data_found then
206 dbms_output.put_line('Nu exista acest pasager sau nu are bilet ');
207 End;
208 /

```

exception
when no_data_found then
dbms_output.put_line('Nu exista acest pasager sau nu are bilet ');\nEnd;
Nu exista acest pasager sau nu are bilet

PL/SQL procedure successfully completed.

4.Tratati exceptia in cazul in care exista pasageri ce doresc sa circule in afara orelor in care autogara este inchisa.

```

Declare
exc exception;
Begin
if to_number(to_char(sysdate,'hh24')) between 0 and 5 then
raise exc;
end if;
exception
when exc then
dbms_output.put_line('intre orele 0 si 5 nu circula busuri, autogara este inchisa');
End;
/

```

```

207 End;
208 /
209
210 Tratati exceptia in cazul in care exista pasageri ce doresc sa circule in afara orelor in care
211 autogara este inchisa.
212 Declare
213 exc exception;
214 Begin
215 if to_number(to_char(sysdate,'hh24')) between 0 and 5 then
216 raise exc;
217 end if;
218 exception
219 when exc then
220 dbms_output.put_line('intre orele 0 si 5 nu circula busuri, autogara este inchisa');
221 End;
222 /

```

Query Result | Script Output | Explain Plan | Task completed in 0.066 seconds

PL/SQL procedure successfully completed.

intre orele 0 si 5 nu circula busuri, autogara este inchisa

PL/SQL procedure successfully completed.

5. Updatati tipul biletului unui pasager pe care sa il introduceti de la tastatura.

```

Declare
cod_invalid exception;
pragma exception_init(cod_invalid, -20888);
Begin
update bilete
set tip_bilet='intreg'
where cod_pasageri=1222;
if sql%notfound then
raise_application_error (-20888, 'cod invalid' );
end if;
exception
when cod_invalid then
dbms_output.put_line('Nu exista pasager cu acest cod care sa aiba bilet');
End;
/

```

```

224 5. Updatati tipul biletului unui pasager pe care sa il introduceti de la tastatura.
225 Declare
226 cod_invalid exception;
227 pragma exception_init(cod_invalid, -20888);
228 Begin
229 update bilet
230 set tip_bilet='intreg'
231 where cod_pasageri=1222;
232 if sql%notfound then
233 raise_application_error (-20888, 'cod invalid' );
234 end if;
235 exception
236 when cod_invalid then
237 dbms_output.put_line('Nu exista pasager cu acest cod care sa aiba bilet');
238 End;
239 /

```

PL/SQL procedure successfully completed.

Nu exista pasager cu acest cod care sa aiba bilet

PL/SQL procedure successfully completed.

E. Gestionarea cursorilor: implicați și expliciti (cu și fără parametri) (minim 2 impliciti, minim 2 expliciti cu parametru, minim 3 expliciti fara parametru)

1. Afiseaza o lista cu numele si codurile controlorilor care sunt responsabili cu autobuzele din flota lor. Realizați o procedură stocată care utilizează un cursor pentru a parurge tabela "CONTROLOR" și să returneze rezultatele dorite.

```

DECLARE
CURSOR c_controlori IS
SELECT cod_controlor, nume_controlor
FROM CONTROLOR;
v_cod_controlor CONTROLOR.cod_controlor%TYPE;
v_nume_controlor CONTROLOR.nume_controlor%TYPE;
BEGIN
OPEN c_controlori;
LOOP
FETCH c_controlori INTO v_cod_controlor, v_nume_controlor;
EXIT WHEN c_controlori%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('Cod controlor: ' || v_cod_controlor || ', Nume controlor: ' ||
v_nume_controlor);
END LOOP;
CLOSE c_controlori;
END;
/

```

```

10  FETCH c_controlori INTO v_cod_controlor, v_nume_controlor;
11  EXIT WHEN c_controlori%NOTFOUND;
12  DBMS_OUTPUT.PUT_LINE('Cod controlor: ' || v_cod_controlor || ', Nume controlor: ' || v_nume_controlor);
13  END LOOP;
14  CLOSE c_controlori;
15  END;
16 /
17
18

```

Script Output | Query Result | Task completed in 0.053 seconds

```

Cod controlor: 200, Nume controlor: Popescu
Cod controlor: 300, Nume controlor: Negru
Cod controlor: 400, Nume controlor: Avramesu
Cod controlor: 500, Nume controlor: Adam
Cod controlor: 600, Nume controlor: Badarau
Cod controlor: 700, Nume controlor: Baranceanu
Cod controlor: 800, Nume controlor: Palamariu
Cod controlor: 900, Nume controlor: Matei
Cod controlor: 110, Nume controlor: Maxim
Cod controlor: 120, Nume controlor: Maxim

PL/SQL procedure successfully completed.

```

2. Modificati tipul tuturor busurilor care au mai putin de 3 compartimente. Contorizati numarul de actualizari facute.

Declare

contor number(3);

Begin

update busuri set tip='maxi' where nr_compartimente<20;

contor:=sql%rowcount;

dbms_output.put_line('s-au realizat'||contor|| 'actualizari');

End;

/

```

236 when cod_invalid then
237 dbms_output.put_line('Nu exista pasager cu acest cod care sa aiba bilet');
238 End;
239 /
240
241 1. Modificati in rapid tipul tuturor busurilor care au mai putin de 3 de vagoane. Contorizati
242 numarul de actualizari facute.
243
244 Declare
245 contor number(3);
246 Begin
247 update busuri set tip='maxi' where nr_compartimente<20;
248 contor:=sql%rowcount;
249 dbms_output.put_line('s-au realizat'||contor|| 'actualizari');
250 End;
251 /

```

Query Result | Script Output | Explain Plan | Task completed in 0.083 seconds

```

PL/SQL procedure successfully completed.

s-au realizat 6 actualizari

PL/SQL procedure successfully completed.

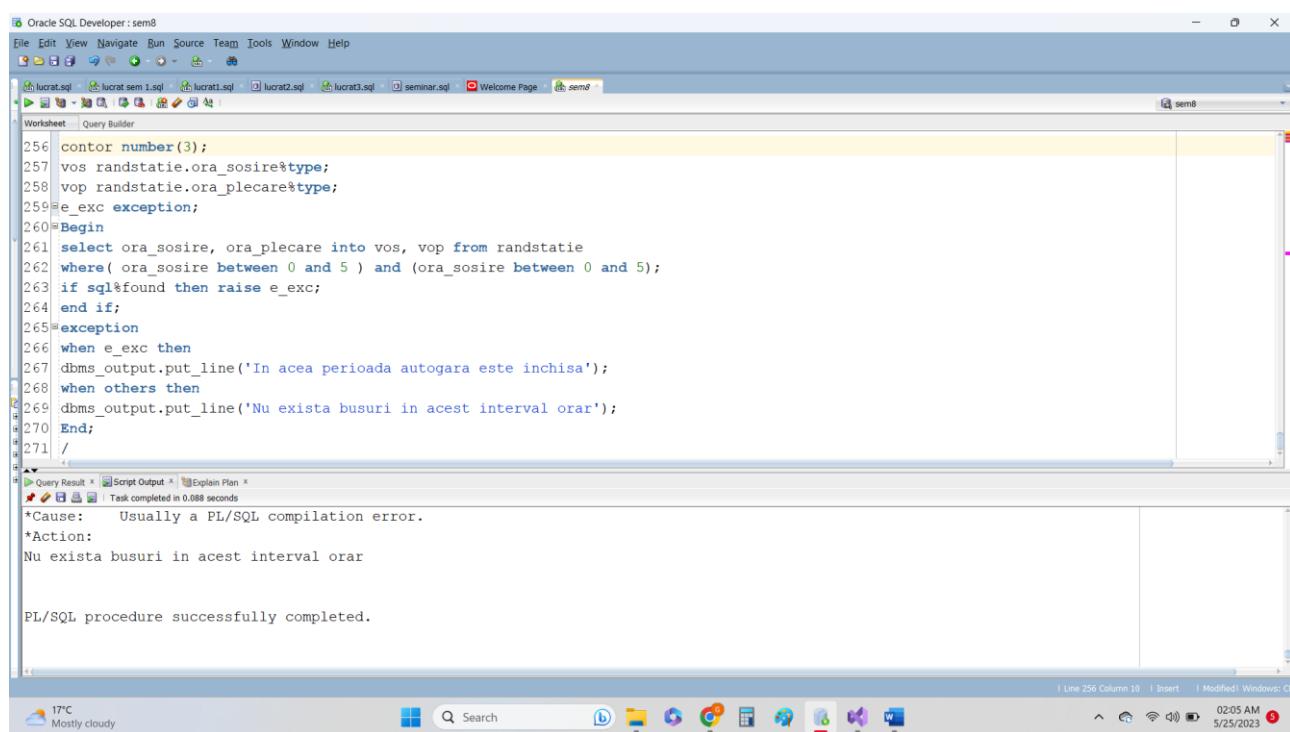
```

3. Verificati daca exista busuri in intervalul orar 0..5 cand autogara este inchisa.

Declare

```
contor number(3);
vos randstatie.ora_sosire%type;
vop randstatie.ora_plecare%type;
e_exc exception;
Begin
select ora_sosire, ora_plecare into vos, vop from randstatie
where( ora_sosire between 0 and 5 ) and (ora_sosire between 0 and 5);
if sql%found then raise e_exc;
end if;
exception
when e_exc then
dbms_output.put_line('In acea perioada autogara este inchisa');
when others then
dbms_output.put_line('Nu exista busuri in acest interval orar');
End;
/

```



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the PL/SQL code. The 'Query Result' tab shows the output of the executed code, which includes a compilation error message and the resulting message 'Nu exista busuri in acest interval orar'. The status bar at the bottom right indicates the date and time as 5/25/2023 02:05 AM.

```
256 contor number(3);
257 vos randstatie.ora_sosire%type;
258 vop randstatie.ora_plecare%type;
259 e_exc exception;
260 Begin
261 select ora_sosire, ora_plecare into vos, vop from randstatie
262 where( ora_sosire between 0 and 5 ) and (ora_sosire between 0 and 5);
263 if sql%found then raise e_exc;
264 end if;
265 exception
266 when e_exc then
267 dbms_output.put_line('In acea perioada autogara este inchisa');
268 when others then
269 dbms_output.put_line('Nu exista busuri in acest interval orar');
270 End;
271 /
*Cause: Usually a PL/SQL compilation error.
*Action:
Nu exista busuri in acest interval orar

PL/SQL procedure successfully completed.
```

4. Stergeti un pasager introdus de la tastatura din tabela PASAGERI. Daca acesta nu exista sau stergerea nu se poate efectua, trateati exceptiile aparute.

```
set serveroutput on
```

Declare

```
v_cod pasageri.cod_pasageri%type:=&codp;
e_exc exception;
Begin
delete from pasageri where cod_pasageri=v_cod;
if sql%notfound then raise e_exc;
```

```

end if;
exception
when e_exc then
dbms_output.put_line('Nu exista un astfel de pasager ');
when others then
dbms_output.put_line('Violeaza restrictia de foreign key');
End;
/

```

```

Oracle SQL Developer : C:\Users\Georgiana\sem8.sql
File Edit View Navigate Run Source Team Tools Window Help
lucrat.sql * lucrat sem 1.sql * lucrat1.sql * lucrat2.sql * lucrat3.sql * seminar.sql * sem8.sql * Welcome Page
+ SQL Worksheet History
Worksheet Query Builder
276 Stergeti un pasager introdus de la tastatura din tabela PASAGERI. Daca acesta nu exista sau
277 stergerea nu se poate efectua, trataati exceptiile aparute.
278
279 set serveroutput on
280 Declare
281 v_cod pasageri.cod_pasageri%type:=&codp;
282 e_exc exception;
283 Begin
284 delete from pasageri where cod_pasageri=v_cod;
285 if sql%notfound then raise e_exc;
286 end if;
287 exception
288 when e_exc then
289 dbms_output.put_line('Nu exista un astfel de pasager ');
290 when others then
291 dbms_output.put_line('Violeaza restrictia de foreign key');
292 End;
293 /

```

Script Output

Violeaza restrictia de foreign key

PL/SQL procedure successfully completed.

5. Sa se afiseze numarul de bilete cumparate de fiecare persoana.

```

Declare
v_cod pasageri.cod_pasageri%type;
v_contor number(3);
cursor c1 is select cod_pasageri, count(nr_bilet) from bilete
group by cod_pasageri
order by count(nr_bilet) desc;
Begin
open c1;
loop
fetch c1 into v_cod, v_contor;
exit when c1%notfound;
dbms_output.put_line('pasagerul cu codul '||v_cod|| ' a achizitionat '|| v_contor|| ' bilet');
end loop;
close c1;
End;
/

```

```

294|
295 4. Sa se afiseze numarul de bilete cumpărate de fiecare persoana.
296 Declare
297 v_cod_pasageri.cod_pasageri%type;
298 v_contor number(3);
299 cursor c1 is select cod_pasageri, count(nr_bilet) from bilete
300 group by cod_pasageri
301 order by count(nr_bilet) desc;
302 Begin
303 open c1;
304 loop
305 fetch c1 into v_cod, v_contor;
306 exit when c1%notfound;
307 dbms_output.put_line('pasagerul cu codul ' ||v_cod|| ' a achizitionat '|| v_contor|| ' bilete');
308 end loop;
309 close c1;
310 End;
311 /

```

Script Output : Task completed in 0.147 seconds

pasagerul cu codul 2222 a achizitionat 1 bilet

PL/SQL procedure successfully completed.

6. Sa se afiseze toate biletele care au pretul mai mare decat valoarea medie a unui bilet

Declare

```

v_cod bilete.nr_bilet%type;
v_tip bilete.tip_bilet%type;
v_pret bilete.pret_bilet%type;
v_med bilete.pret_bilet%type;
cursor c1 is SELECT nr_bilet, tip_bilet, pret_bilet FROM BILETE
WHERE pret_bilet >(SELECT AVG(pret_bilet) FROM BILETE) ;
Begin
select avg(pret_bilet)into v_med from bilete;
dbms_output.put_line('Valoarea medie a pretului este: '||v_med);
open c1;
loop
fetch c1 into v_cod, v_tip, v_pret;
exit when c1%notfound;
dbms_output.put_line('biletele '||v_cod|| ' de tip '|| v_tip|| ' are pretul '|| v_pret);
end loop;
close c1;
End;
/

```

```

315' v_cod bilet.nr_bilet%type;
316 v_tip bilet.tip_bilet%type;
317 v_pret bilet.pret_bilet%type;
318 v_med bilet.pret_bilet%type;
319 cursor c1 is SELECT nr_bilet, tip_bilet, pret_bilet FROM BILETE
320 WHERE pret_bilet >(SELECT AVG(pret_bilet) FROM BILETE) ;
321 Begin
322 select avg(pret_bilet)into v_med from bilet;
323 dbms_output.put_line('Valoarea medie a pretului este: '||v_med);
324 open c1;
325 loop
326 fetch c1 into v_cod, v_tip, v_pret;
327 exit when c1%notfound;
328 dbms_output.put_line('biletele '||v_cod || ' de tip '|| v_tip|| ' are pretul '|| v_pret);
329 end loop;
330 close c1;
331 End;
332 /

```

Script Output : Task completed in 0.095 seconds

biletele 119 de tip elev are pretul 33

PL/SQL procedure successfully completed.

7. Sa se afiseze numele pasagerilor din fiecare bus

Declare

cursor c1 is

```
select bilete.cod_bus, busuri.tip from busuri, bilete
where busuri.cod_bus = bilete.cod_bus
order by cod_bus;
```

cursor c2 (p_cod_bus number) is

```
select bilete.cod_pasageri, pasageri.nume from pasageri, bilete
where bilete.cod_pasageri=pasageri.cod_pasageri and p_cod_bus=bilete.cod_bus
order by cod_pasageri;
```

rec_bus c1%rowtype;

rec_pasag c2%rowtype;

Begin

open c1;

loop

fetch c1 into rec_bus;

exit when c1%notfound;

dbms_output.put_line(' Busul '|| rec_bus.cod_bus||' '|| rec_bus.tip);

open c2(rec_bus.cod_bus);

loop

fetch c2 into rec_pasag;

exit when c2%notfound;

dbms_output.put_line(' Pasagerul '|| rec_pasag.cod_pasageri||' cu numele '||rec_pasag.nume);

end loop;

close c2;

dbms_output.put_line('=====');

end loop;

close c1;

End;

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a file tab labeled 'C:\Users\Georgiana\sem8.sql'. The main workspace contains a 'Worksheet' tab with the following PL/SQL code:

```
346  Begin
347    open cl;
348  loop
349    fetch cl into rec_bus;
350    exit when cl%notfound;
351    dbms_output.put_line(' Busul ' || rec_bus.cod_bus|| ' ' || rec_bus.tip);
352    open c2(rec_bus.cod_bus);
353  end loop;
354  close c1;
355  close c2;
356 End;
```

Below the code, the 'Script Output' pane displays the results of the execution:

```
PL/SQL procedure successfully completed.

Busul 111 maxi
Pasagerul 1111 cu numele Andronic Eliza
Pasagerul 6666 cu numele Godea Andreea
Pasagerul 7777 cu numele Avramescu Andreea
=====
Busul 111 maxi
Pasagerul 1111 cu numele Andronic Eliza
Pasagerul 6666 cu numele Godea Andreea
Pasagerul 7777 cu numele Avramescu Andreea
=====
Busul 111 maxi
Pasagerul 1111 cu numele Andronic Eliza
Pasagerul 6666 cu numele Godea Andreea
Pasagerul 7777 cu numele Avramescu Andreea
```

The system tray at the bottom shows the weather as 26°C Rain showers. The taskbar includes icons for File Explorer, Task View, Start, Taskbar settings, and several pinned applications.

8.Compania de transport național dorește să obțină o listă cu toate stațiile dintr-un anumit oraș, specificat ca parametru. În plus, se dorește și numărul total de linii disponibile în orașul respectiv. Realizați o procedură stocată care utilizează un cursor cu parametru pentru a parcurge tabela "ORAS_DESTINATIE" și să returneze rezultatele dorite.

```
SET SERVEROUTPUT ON
CREATE OR REPLACE PROCEDURE lista_stati(
    p_cod_oras IN ORAS_DESTINATIE.cod_oras%TYPE
) AS
    CURSOR c_stati IS
        SELECT s.cod_statie, s.den_statie, s.linii
        FROM STATII s
        INNER JOIN ORAS_DESTINATIE o ON s.cod_statie = o.cod_statie
        WHERE o.cod_oras = p_cod_oras;
    v_cod_statie STATII.cod_statie%TYPE;
    v_den_statie STATII.den_statie%TYPE;
    v_linii STATII.linii%TYPE;
    v_total_linii NUMBER := 0;
BEGIN
    OPEN c_stati;
    LOOP
        FETCH c_stati INTO v_cod_statie, v_den_statie, v_linii;
        EXIT WHEN c_stati%NOTFOUND;
        v_total_linii := v_total_linii + v_linii;
        DBMS_OUTPUT.PUT_LINE('Cod statie: ' || v_cod_statie || ', Denumire statie: ' || v_den_statie);
    END LOOP;
```

```

CLOSE c_statii;
DBMS_OUTPUT.PUT_LINE('Numar total de linii disponibile in orasul specificat: ' ||
v_total_linii);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Nu exista informatii pentru codul de oras specificat.');
END;
/

```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a PL/SQL script is being run. The code defines a procedure 'lista_statii' that takes a parameter 'p_cod_oras' of type 'ORAS_DESTINATIE.cod_oras%TYPE'. The procedure uses a cursor 'c_statii' to select data from 'STATII' and join it with 'ORAS_DESTINATIE'. It then loops through the cursor to fetch data into variables 'v_cod_statie', 'v_den_statie', and 'v_linii', and adds them to a total 'v_total_linii'. The procedure is compiled successfully.

```

SET SERVEROUTPUT ON
CREATE OR REPLACE PROCEDURE lista_statii(
    p_cod_oras IN ORAS_DESTINATIE.cod_oras%TYPE
) AS
    CURSOR c_statii IS
        SELECT s.cod_statie, s.den_statie, s.linii
        FROM STATII s
        INNER JOIN ORAS_DESTINATIE o ON s.cod_statie = o.cod_statie
        WHERE o.cod_oras = p_cod_oras;
    v_cod_statie STATII.cod_statie%TYPE;
    v_den_statie STATII.den_statie%TYPE;
    v_linii STATII.linii%TYPE;
    v_total_linii NUMBER := 0;
BEGIN
    OPEN c_statii;
    LOOP
        FETCH c_statii INTO v_cod_statie, v_den_statie, v_linii;
        EXIT WHEN c_statii%NOTFOUND;
        v_total_linii := v_total_linii + v_linii;
    END LOOP;
    CLOSE c_statii;
END;
/

```

F. Funcții, proceduri, includerea acestora în pachete (minim 3 funcții, 3 proceduri și un pachet).

- Actualizati pretul tuturor biletelor mai mari de 30 cu cel al persoanei cu codul 5555.

Create or replace procedure actualizare
(codp in bilete.cod_pasageri%type)
is
Begin
update bilete
set pret_bilet=(select pret_bilet from bilete where cod_pasageri=codp)
where pret_bilet >30;
End;
/

apel :

```

call actualizare(5555);
select * from bilete;

```

```

1  (codp in biletete.cod_pasageri%type)
2
3  is
4    Begin
5      update biletete
6      set pret_bilet=(select pret_bilet from biletete where cod_pasageri=codp)
7      where pret_bilet >30;
8    End;
9
10   --apel :
11   call actualizare(5555);
12   select * from biletete;
13

```

Script Output:

NR_BILET	TIP_BILET	DURATA_CALAT	DATA_CALA	PRET_BILET	PREFERINTA	COD_PASAGERI	COD_BUS
111	student	6	20-DEC-07	40	2	1111	182
112	elev	4	20-DEC-07	30	2	2222	127
113	intreg	8	18-NOV-07	40	1	3333	127
114	intreg	5	18-NOV-07	40	2	4444	182
115	intreg	3	20-DEC-07	40	1	5555	131
116	intreg	10	13-NOV-07	40	1	6666	131
117	student	2	18-NOV-07	20	2	1000	127
118	student	3	13-NOV-07	30	2	1111	131
119	elev	6	15-DEC-07	40	2	8888	182
120	intreg	1	13-NOV-07	25	1	7777	111

2.Să se creeze o procedură stocată în limbajul PL/SQL care primește ca parametru codul unui controlor și returnează informații despre acel controlor din tabela "CONTROLOR".

```

CREATE OR REPLACE PROCEDURE informatii_controlor
  p_cod_controlor IN CONTROLOR.cod_controlor%TYPE
) AS
  v_nume_controlor CONTROLOR.nume_controlor%TYPE;
  v_cod_bus CONTROLOR.cod_bus%TYPE;
BEGIN
  SELECT nume_controlor, cod_bus
  INTO v_nume_controlor, v_cod_bus
  FROM CONTROLOR
  WHERE cod_controlor = p_cod_controlor;

  DBMS_OUTPUT.PUT_LINE('Informatii despre controlor:');
  DBMS_OUTPUT.PUT_LINE('Cod Controlor: ' || p_cod_controlor);
  DBMS_OUTPUT.PUT_LINE('Nume Controlor: ' || v_nume_controlor);
  DBMS_OUTPUT.PUT_LINE('Cod Bus: ' || v_cod_bus);

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nu există informații pentru codul de controlor specificat.');
END;
/

```

```

CREATE OR REPLACE PROCEDURE informatii_controlor(
    p_cod_controlor IN CONTROLOR.cod_controlor%TYPE
)
AS
    v_nume_controlor CONTROLOR.nume_controlor%TYPE;
    v_cod_bus CONTROLOR.cod_bus%TYPE;
BEGIN
    SELECT nume_controlor, cod_bus
    INTO v_nume_controlor, v_cod_bus
    FROM CONTROLOR
    WHERE cod_controlor = p_cod_controlor;
    DBMS_OUTPUT.PUT_LINE('Informatii despre controlor:');
    DBMS_OUTPUT.PUT_LINE('Cod Controlor: ' || p_cod_controlor);
    DBMS_OUTPUT.PUT_LINE('Nume Controlor: ' || v_nume_controlor);
    DBMS_OUTPUT.PUT_LINE('Cod Bus: ' || v_cod_bus);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu există informații pentru codul de controlor specificat.');
END;

```

Procedure INFORMATII_CONTROLOR compiled

3. Realizati o procedură verif_ani (p_nr OUT number) prin care să calculați numărul de pasageri care au varsta mai mare decât varsta medie.

a. declarați o excepție definită de utilizator prin care să tratați cazul în care există persoane cu varsta null;

b. în secțiunea executabilă realizați următoarele:

- afișați varsta medie a persoanelor din tabela Pasageri;
- deschideți cursorul și parcurgeți persoanele
- calculați numărul de persoane care au varsta mai mare decât varsta medie
- declanșați și tratați excepția definită de utilizator (varsta null) precum și alte erori apărute.

Apelați procedura dintr-un bloc PL/SQL.

Create or replace procedure verif_ani (p_nr OUT number,v_med out number) is

v_cod pasageri.cod_pasageri%type;

v_varsta pasageri.varsta%type;

e_varsta exception;

pragma exception_init(e_varsta, -01400);

Cursor c1 is

Select cod_pasageri, varsta from pasageri;

Begin

Select avg(varsta) into v_med from pasageri;

Open c1;

Loop

fetch c1 into v_cod, v_varsta;

if v_varsta>v_med then p_nr:=p_nr+1;

end if;

exit when c1%notfound;

end loop;

exception

```

when e_varsta then
dbms_output.put_line('Campul este null');
when others then
dbms_output.put_line('Exista erori in datele introduse');
End;
/

```

apel :

```

Declare
p_nr number(3);
v_med number(3);
Begin
verif_ani(p_nr,v_med);
Dbms_output.put_line('Varsta medie este:'||v_med);
dbms_output.put_line('Numarul de persoane ce au varsta mai mare decat varsta medie:'||p_nr);
End;
/

```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a PL/SQL procedure is being run:

```

19 when e_varsta then
20 dbms_output.put_line('Campul este null');
21 when others then
22 dbms_output.put_line('Exista erori in datele introduse');
23 End;
24 /
25 --apel :
26 Declare
27 p_nr number(3);
28 v_med number(3);
29 Begin
30 verif_ani(p_nr,v_med);
31 Dbms_output.put_line('Varsta medie este:'||v_med);
32 dbms_output.put_line('Numarul de persoane ce au varsta mai mare decat varsta medie:'||p_nr);
33 End;
34 /
35

```

The script output window shows the results of the execution:

```

Varsta medie este:27
Numarul de persoane ce au varsta mai mare decat varsta medie:

PL/SQL procedure successfully completed.

```

4.Să se creeze o funcție în limbajul PL/SQL care primește ca parametru un cod de stație și returnează o listă cu toate orașele destinație disponibile pentru acea stație, împreună cu numărul total de linii disponibile.

```

CREATE OR REPLACE FUNCTION get_orase_destinatie(
    p_cod_statie IN STATII.cod_statie%TYPE
) RETURN SYS_REFCURSOR AS
    v_cursor SYS_REFCURSOR;
BEGIN
    OPEN v_cursor FOR
        SELECT OD.nume_oras, COUNT(DISTINCT S.linii) AS numar_linii

```

```

FROM ORAS_DESTINATIE OD
INNER JOIN RANDSTATIE R ON OD.cod_oras = R.cod_statie
INNER JOIN STATII S ON R.cod_statie = S.cod_statie
WHERE S.cod_statie = p_cod_statie
GROUP BY OD.numor_oras;

RETURN v_cursor;
END;
/

```

5. Creati o functie care sa modifice pretul biletului unui pasager, scazand o suma, in cazul in care biletul este pentru preferinta a 2-a(sa nu stea la geam). Se trateaza cazul in care modificarea nu are loc din cauza precizarii unui cod care nu se regaseste in tabelă.

```

Create or replace function verifica_bilet(codp pasageri.cod_pasageri%type) return Boolean is
v_codp bilete.cod_pasageri%type;
v_codb bilete.nr_bilet%type;
v_pret bilete.pret_bilet%type;
v_medie number(4,2);
Begin
select nr_bilet, pret_bilet,cod_pasageri into v_codb, v_codp,v_pret from bilete
where cod_pasageri=codp;
select avg(pret_bilet )into v_medie from bilete;
if v_pret > v_medie then return true;
else return false;
end if;
exception
when no_data_found then return null;
End;
/

```

```

apel :
Begin
if (verifica_bilet(2222)is null) then dbms_output.put_line('nu exista acest pasager');
end if;
if (verifica_bilet(2222)) then
dbms_output.put_line('pasagerul are un bilet cu pret mai mare decat media');
else
dbms_output.put_line('pasagerul are un bilet cu un pret mai mic decat media');
end if;
End;
/

```

The screenshot shows the Oracle SQL Developer interface. In the top menu bar, 'File Edit View Navigate Run Source Team Tools Window Help' are visible. Below the menu is a toolbar with various icons. The main workspace is titled 'Worksheet - Query Builder' and contains the following PL/SQL code:

```

15 End;
16 /
17
18 --apel :
19 Begin
20 if (verifica_bilet(2222)is null) then dbms_output.put_line('nu exista acest pasager');
21 end if;
22 if (verifica_bilet(2222)) then
23 dbms_output.put_line('pasagerul are un bilet cu pret mai mare decat media');
24 else
25 dbms_output.put_line('pasagerul are un bilet cu un pret mai mic decat media');
26 end if;
27 End;

```

Below the code, the 'Script Output' tab shows the results of the execution:

```

PL/SQL procedure successfully completed.

Function VERIFICA_BILET compiled

pasagerul are un bilet cu pret mai mare decat media

PL/SQL procedure successfully completed.

```

The status bar at the bottom right indicates 'Line 20 Column 64' and 'Modified! Windows: CF'. The system tray shows the date and time as '5/27/2023 05:24 PM'.

6. Realizati o functie care sa calculeze cat de mult sta busul intr-o anumita gara, stiind ca timpul de plecare si de sosire este exprimat in ore.

```

Create or replace function stationare(codb randstatie.cod_bus%type, codg
randstatie.cod_statie%type) return number is
v_oras randstatie.ora_sosire%type;
v_orap randstatie.ora_plecare%type;
v_codb randstatie.cod_bus%type;
v_cods randstatie.cod_statie%type;
v_stat number(4,2);
Begin
select cod_bus, cod_statie, ora_sosire, ora_plecare into v_codb, v_cods, v_oras, v_orap from
randstatie
where cod_bus=codb and cod_statie=codg;
v_stat:=v_orap - v_oras;

```

```

return v_stat;
exception
when no_data_found then return null;
End;
/
--apel :
Declare
codb randstatie.cod_bus%type;
stat number(4,2);
begin
stat :=stationare(182,10);
dbms_output.put_line('In aceata statie busul stationeaza'||stat|| ' ore');
End;
/
select * from randstatie;

```

```

13 return v_stat;
14 exception
15 when no_data_found then return null;
16 end;
17
18 select * from randstatie;

```

COD_BUS	COD_STATIE	LINIE	ORA_SOSIRE	ORA_PLECARE
182	1	1	15.05	15.2
182	5	3	18.25	18.35
182	10	1	19.5	20
182	4	1	21.21	21.31
127	9	1	10	10
127	6	3	11	11.07
127	1	5	18.45	19
127	8	2	19.5	19.5
131	1	12	7	7
131	7	1	11.2	11.2
111	9	3	12.15	12.15

COD_BUS	COD_STATIE	LINIE	ORA_SOSIRE	ORA_PLECARE
111	6	2	14.15	14.15

12 rows selected.

7. Realizati un pachet

```

Create or replace package pachet is
procedure adauga_pasager (v_pas pasageri.cod_pasageri%type, v_num pasageri.nume%type,
v_varsta pasageri.varsta%type);
procedure modifica_pasager (v_pas pasageri.cod_pasageri%type, v_num pasageri.nume%type,
v_varsta pasageri.varsta%type);
function verif_cod (v_pas pasageri.cod_pasageri%type)
return boolean;
e1 exception;
End;
/

```

Create or replace package body pachet is

```
procedure adauga_pasager (v_pas pasageri.cod_pasageri%type, v_nume pasageri.nume%type,
v_varsta pasageri.varsta%type)
is
begin
if verif_cod(v_pas)then raise e1;
else insert into pasageri values (v_pas, v_nume, v_varsta);
end if;
exception
when e1 then dbms_output.put_line('Exista pasagerul');
end;

procedure modifica_pasager(v_pas pasageri.cod_pasageri%type, v_nume pasageri.nume%type,
v_varsta pasageri.varsta%type) is
Begin
if verif_cod(v_pas) then update pasageri
set nume=v_nume, varsta=v_varsta where cod_pasageri=v_pas;
else raise e1;
end if;
exception
when e1 then
dbms_output.put_line('pasag cu acest cod nu exista');
End;

function verif_cod (v_pas pasageri.cod_pasageri%type) return boolean is
v_unu number;
begin
select 1 into v_unu from pasageri where cod_pasageri=v_pas;
return true;
exception
when no_data_found then return false;
end;
end;
/
```

```

30 else raise el;
31 end if;
32 exception
33 when el then
34 dbms_output.put_line('pasag cu acest cod nu exista');
35 End;
36
37 function verif_cod (v_pas pasageri.cod_pasageri%type) return boolean is
38 v_unu number;
39 begin
40 select 1 into v_unu from pasageri where cod_pasageri=v_pas;
41 return true;
42 exception
43 when no_data_found then return false;
44 end;
45 end;
46 /

```

Script Output x Task completed in 0.073 seconds

Package Body PACHET compiled

Package Body PACHET compiled

G. Declanșatori (minim 2).

1. Creati un trigger care sa actualizeze cod_pasager si in tabela BILETE, cand acesta este modificat in tabela PASAGERI.

```

Create or replace trigger inloc_cod
before update of cod_pasageri on pasageri
for each row
Begin
update bilet
set cod_pasageri=:new.cod_pasageri
where cod_pasageri=:old.cod_pasageri;
End;
/

```

--Apelare:

```

update pasageri
set cod_pasageri=2000
where cod_pasageri=1111;
select * from bilete;
select * from pasageri;

```

The screenshot shows the Oracle SQL Developer interface. In the top menu bar, the path "File Edit View Navigate Run Source Team Tools Window Help" is visible. The title bar indicates "Oracle SQL Developer : Arosoae Georgiana 1048C". The main workspace contains a "Worksheet" tab with the following PL/SQL code:

```

3 -- in tabela PASAGERI.
4 Create or replace trigger inloc_cod
5 before update of cod_pasageri on pasageri
6 for each row
7 Begin
8   update bilet
9   set cod_pasageri=:new.cod_pasageri
10 where cod_pasageri=:old.cod_pasageri;
11 End;
12 /
13
14 --Apelare:
15 update pasageri
16 set cod_pasageri=2000
17 where cod_pasageri=1111;
18 select * from bilet;
19 select * from pasageri;

```

Below the code, a "Script Output" window shows the results of the trigger compilation and execution:

```

Trigger INLOC_COD compiled

1 row updated.

```

The status bar at the bottom right shows "Line 10 Column 38 Insert Modified Windows: C:\ 05:41 PM 5/27/2023". The system tray icon shows the date and time as "23°C Mostly cloudy".

2. Realizați un trigger care să nu permită introducerea în tabela STATII valori negative pentru liniile existente în acea statie.

```

Create or replace trigger nu_neg
before insert or update on statii
for each row
Begin
if :new.linii< 0 then
raise_application_error (-20203,'Linie negativa, imposibil);
end if;
End;
/

```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a trigger named 'NU_NEG' is being created:

```

1 Create or replace trigger nu_neg
2 before insert or update on statii
3 for each row
4 Begin
5 if :new.linii< 0 then
6 raise_application_error (-20203,'Linie negativa, imposibil');
7 end if;
8 End;
9 /

```

The code is highlighted in blue, indicating syntax highlighting. Below the code, the output window shows the results of the compilation:

Trigger NU_NEG compiled

Trigger NU_NEG compiled

The status bar at the bottom right indicates the date and time: 5/29/2023 09:39 PM.

3. Realizati un trigger care sa nu permita achizitionarea unui bilet de tip elev cu mai mult 20 de unitati monetare .

Create or replace trigger nepermis
before update on bilete
for each row
when (old.tip_bilet='elev')
Begin
if :new.pret_bilet>20 then
raise_application_error (-20203,'Biletul este prea scump pentru un elev!');
end if;
End;

Oracle SQL Developer : Arosoae Georgiana 1048C

```

20--2. Realizati un trigger care sa nu permita introducerea in tabela
21--liniile existente in acea statie.
22>Create or replace trigger nu_neg
23  before insert or update on statii
24  for each row
25  Begin
26  if :new.linii< 0 then
27  raise_application_error (-20203,'linie negativa, imposibil');
28  end if;
29  End;
30 /
31--3. Realizati un trigger care sa nu permita achizitionarea unui bil
32--unitati monetare .
33>Create or replace trigger nepermis
34  before update on bilet
35  for each row
36  when (old.tip_bilet='elev')

```

Trigger NU_NEG compiled

Trigger NEPERMIS compiled

Microsoft Word window showing a warning message: "UPLOAD BLOCKED" and "SUBSCRIPTION CANCELLED".

APEX:

App Builder

APEX App Builder

Create Import Dashboard Workspace Utilities

Gestione Busuri 21526 GestioneSGBD 21526 TabeleSGBDProject 215354 Tablebus 26169

About

The App Builder enables you to design, develop and deploy applications which are beautiful and responsive.

Recent

TabeleSGBDProject - 21534

Gestione Busuri - 215126

GestioneSGBD - 21526

Tablebus - 26169

Tasks

Manage Backups >

Browse by Facets >

Learn more about App Builder

Get Started

https://apex.oracle.com/pls/apex/r/apex/app-builder/home?fb_flow_id=215334&f4000_p1_f...

Copyright © 1999, 2023, Oracle and/or its affiliates.

Oracle APEX 23.1.0

App Builder Tabela

https://apex.oracle.com/pls/apex/r/apex/app-builder/home?fb_flow_id=215334&fb_flow_page_id=215334&p0_flowpage=215334&recent_pages=&session=101244800739418

Application 215334 - TabeleSGBDProject

Edit Application Definition

About

Use the Application home page to run, edit, import, export, copy, or delete an application. Select a page to edit the page properties, or click **Create Page** to add a page to the application.

Learn More ...

Tasks

Delete this Application >
Copy this Application >
Add Feature Page >
Manage Backups >
Browse by Facets >

Recently Edited Pages

17. Busuri
1. Home
15. Randstatiere
2. Pasageri
3. Pasageri
4. Busuri
5. Busuri
6. Contorlor

Run Application Supporting Objects Shared Components Utilities Export / Import

Q Go Actions Create Page >

0 - Global Page 1 - Home 2 - Pasageri 3 - Pasageri 4 - Busuri 5 - Busuri
6 - Contorlor 7 - Contorlor 8 - Bilete 9 - Bilete 10 - Stati 11 - Status
12 - Destinatie 13 - Destinatie 14 - Randstatiere 15 - Randstatiere 17 - Busuri 9999 - Login Page

1 - 18

Copyright © 1999, 2023, Oracle and/or its affiliates.

erossorocigiana21@stud.usc.ro projectsgbd2023 en

1 Calea Victoriei Closed road

Search

Oracle APEX 23.1

05:38 PM 5/28/2023

Page Designer Tabela GPT-4

https://apex.oracle.com/pls/apex/r/apex/app-builder/page-designer?fb_flow_id=215334&fb_flow_page_id=17&session=101244800739418

APEX App Builder SQL Workshop Team Development Gallery

Application 215334 \ Page Designer

✓ Form page created.

Layout Page Search Help

Page

Filter Identification

Name: Busuri
Alias: busuri2
Title: Busuri
Page Group: -Select-

Appearance

Page Mode: Normal
Page Template: Theme Default
Template Options: Use Template Defaults
CSS Classes
Media Type

BANNER AFTER LOGO BEFORE NAVIGATION BAR AFTER NAVIGATION BAR

BREADCRUMB BAR

UP REGION BODY REGION CONTENT SUB REGIONS

Regions Items Buttons

Avatar Badge Breadcrumb Calendar Cards
Chart Classic Report Column Toggle Report Comments Content Row
Dynamic Content Faceted Search Form Help Text Interactive Grid

23°C Sunny

Search

05:36 PM 5/28/2023

Object Browser Randstatie +

<https://apex.oracle.com/pls/apex/r/apex/sql-workshop/ob?session=101535042090923>

APEX App Builder SQL Workshop Team Development Gallery

Object Browser Search Schema WKSP_PROJECTSGBD2023 AG projectsghd2023

PACHET

Specification Body Dependencies Errors Grants

Download Save and Compile Drop Refresh

Tables: BILETE, BUSURI, CONTROLOR, ORAS_DESTINATIE, PASAGERI, RANDSTATIE, STATII, TABLEBUS

Views: INDEXES, SEQUENCES, TYPES

Packages: **PACHET**

```

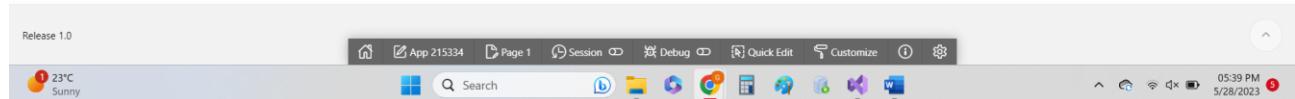
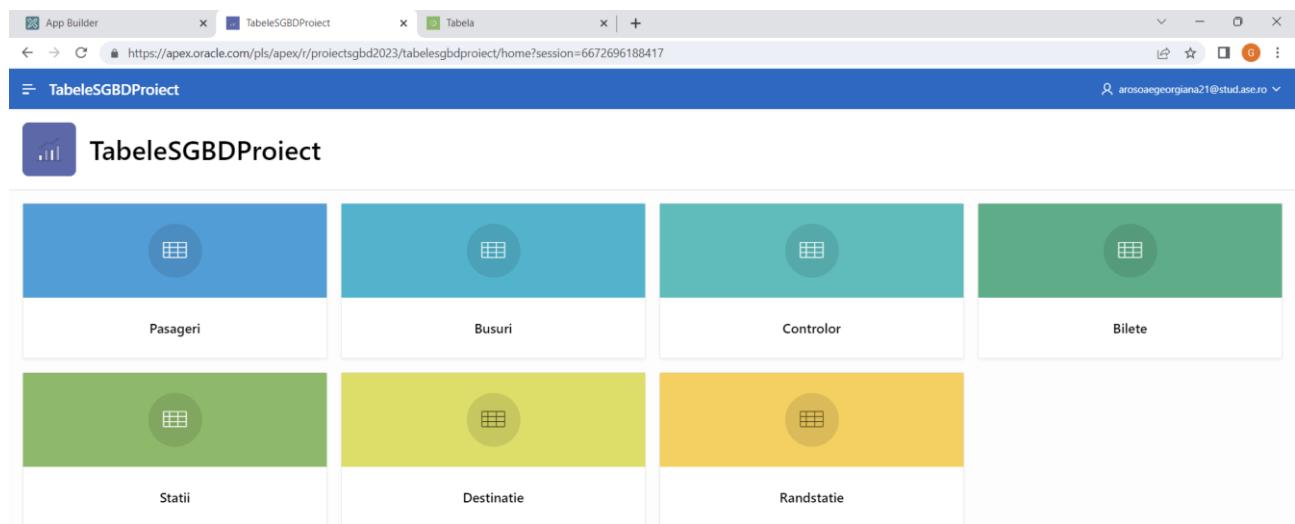
1  create or replace package pachet is
2  procedure adauga_pasager (v_pas_pasageri.cod_pasageri%type, v_nume_pasageri.nume%type,
3  v_varsta_pasageri.varsta%type);
4  procedure modifica_pasager (v_pas_pasageri.cod_pasageri%type, v_nume_pasageri.nume%type,
5  v_varsta_pasageri.varsta%type);
6  function verif_cod (v_pas_pasageri.cod_pasageri%type)
7  return boolean;
8  e1 exception;
9  End;
10 /
11
12 Create or replace package body pachet is
13
14 procedure adauga_pasager (v_pas_pasageri.cod_pasageri%type, v_nume_pasageri.nume%type,
15 v_varsta_pasageri.varsta%type)
16 is
17 begin
18 if verif_cod(v_pas)then raise e1;
19 else insert into pasageri values (v_pas, v_nume, v_varsta);
20 end if;
21 exception
22 when e1 then dbms_output.put_line('Exista pasagerul');
23 end;
24
25 procedure modifica_pasager(v_pas_pasageri.cod_pasageri%type, v_nume_pasageri.nume%type,
26 v_varsta_pasageri.varsta%type)
27 is
28 begin
29 update pasageri set nume=v_nume, varsta=v_varsta where cod=v_pas;
30 end;
31
32 end;

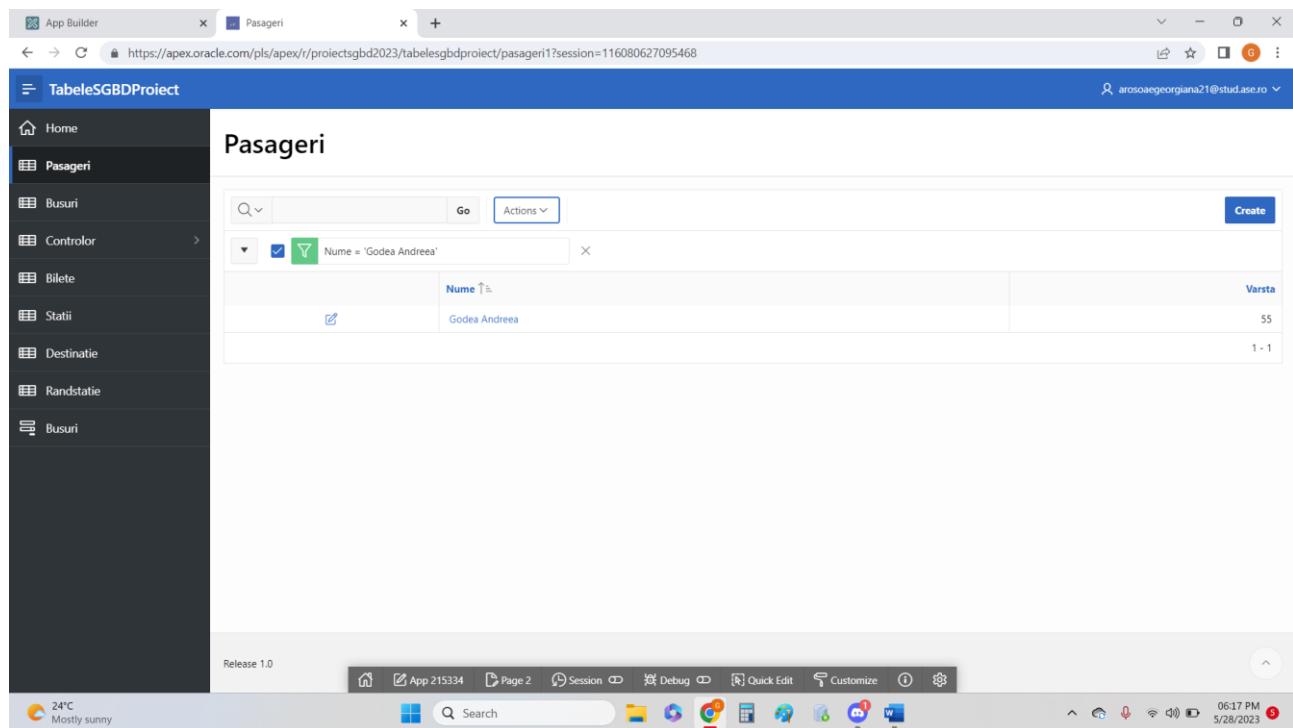
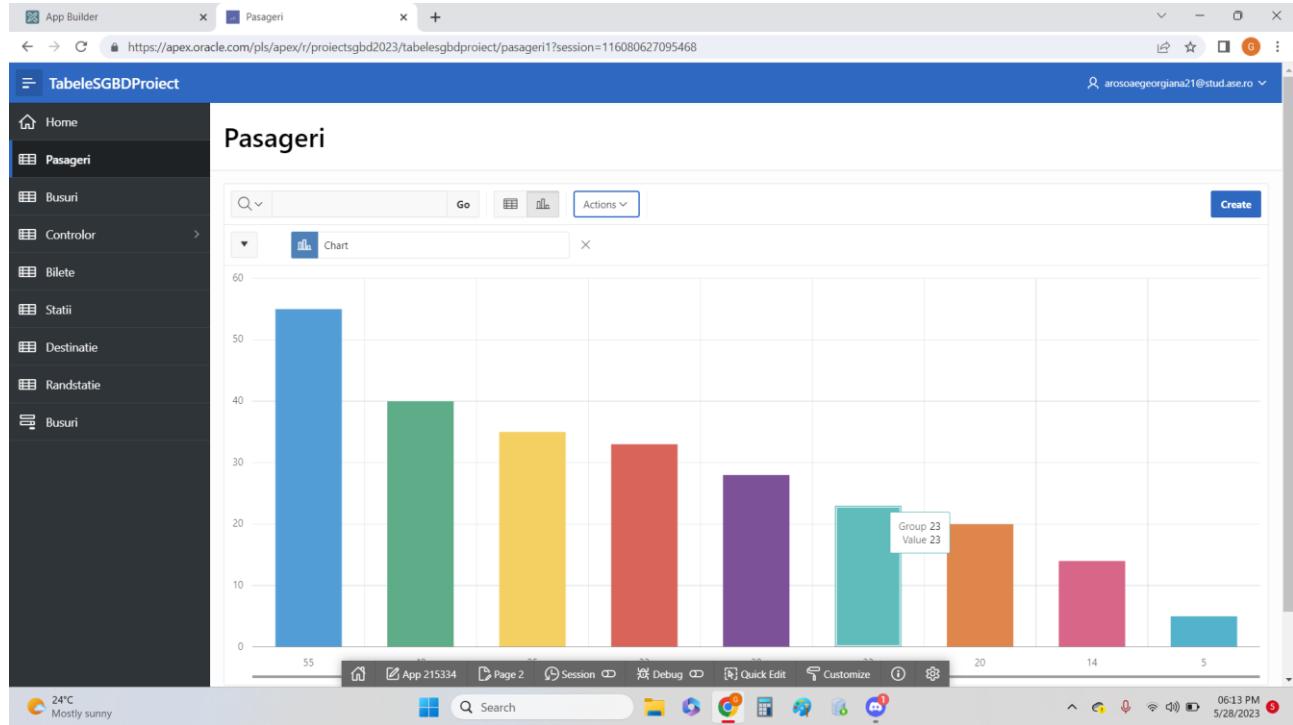
```

Copyright © 1999, 2023, Oracle and/or its affiliates.

arosoaegeorgiana21@stud.ase.ro projectsghd2023 06:33 PM 5/28/2023

23°C Partly sunny





App Builder x Randstatiere x + https://apex.oracle.com/pls/apex/r/proiectsgbd2023/tabelesgbdproject/randstatiere?session=116080627095468 v - o x

TabelesGBDProject x

Home x Randstatiere

- Pasageri
- Busuri
- Controlor
- Bilete
- Statii
- Destinatie
- Randstatiere**
- Busuri

Randstatiere

	Cod Bus ↑	Cod Statie
<input checked="" type="checkbox"/>	delux	Bacau
<input checked="" type="checkbox"/>	delux	Bucuresti
<input checked="" type="checkbox"/>	maxi	Oradea
<input checked="" type="checkbox"/>	maxi	Brasov
<input checked="" type="checkbox"/>	maxi	Bucuresti
<input checked="" type="checkbox"/>	maxi	Arad
<input checked="" type="checkbox"/>	midi	Bucuresti
<input checked="" type="checkbox"/>	midi	Craiova
<input checked="" type="checkbox"/>	midi	Petrosani
<input checked="" type="checkbox"/>	midi	Targu Jiu
<input checked="" type="checkbox"/>	mini	Arad
<input checked="" type="checkbox"/>	mini	Oradea

Release 1.0 Go Actions v

24°C Mostly sunny Search Session Debug Quick Edit Customize Create 06:18 PM 5/28/2023

App Builder x Randstatiere x + https://apex.oracle.com/pls/apex/r/proiectsgbd2023/tabelesgbdproject/randstatiere?session=116080627095468 v - o x

TabelesGBDProject x

Home x Randstatiere

- Pasageri
- Busuri
- Controlor
- Bilete
- Statii
- Destinatie
- Randstatiere**
- Busuri

arosoaegeorgiana21@stud.ase.ro

Randstatiere

	7	10	11	11.2	12.15	14.15	15.05	18.25	18.45	19.5	21.21
Ora Plecare	Maximum Linie										
18.35									3.00		
14.15								2.00			
10		1.00									
11.07			3.00								
7	12.00										
12.15				3.00							
21.31										1.00	
20									1.00		
15.2							1.00				
11.2				1.00							
19.5								2.00			
19								5.00			

1 - 12 Search Session Debug Quick Edit Customize 06:20 PM 5/28/2023

23°C Partly sunny

