

Proiectarea unei Baze de Date a Aeroporturilor din România

Realizat de: Lăzăroiu Teodora-Bianca, grupa 241

Contents

Descrierea modelului	2
Diagrama Entitate-Relație	3
Diagrama Conceptuală	3
Crearea tabelelor și a constrângerilor	4
Inserarea datelor	7
Problema cu subprogram stocat și două tipuri de colecție	22
Problema cu subprogram stocat și un cursor	24
Problema cu funcție și 3 tabele într-o comandă	26
Problema cu procedură și 5 tabele într-o comandă	30
Trigger de tip LMD la nivel de comandă	32
Trigger de tip LMD la nivel de linie	33
Trigger de tip LDD	34
Pachet cu obiectele din proiect	35
Pachet cu flux de acțiuni integrate	40

Descrierea modelului

În acest paragraf vor fi prezentate descrierea modelului real al bazei de date ce urmează a fi proiectată, utilitățile acestuia dar și reguli sale de funcționare.

Modelul de date va gestiona aeroporturile din România și zborurile lor comerciale și interne, ceea ce înseamnă că, pentru crearea acestui model, se vor ignora zborurile în afara țării. Fiecare aeroport se află într-un oraș, iar orașul la rândul său va putea fi încadrat într-o regiune. De asemenea, la fiecare aeroport sunt angajați piloți și însoțitori de zbor care vor reprezenta echipajul necesar călătoriilor desfășurate. Un pilot sau însoțitor poate fi angajat la un singur aeroport. O companie aeriană deține și operează diferite modele de avioane. Se menționează că între aeroport și avioane nu există nicio legătură deoarece aeroporturile nu dețin propriile aeronave ci doar primesc o chirie plătită de firmele aeriene pentru a opera pe acel aeroport.

Un pasager își poate rezerva un loc pentru o călătorie iar, în cadrul unei rezervări, clientul trebuie să își aleagă clasa la care dorește să călătorească. Bineînțeles, anumite clase vor avea un preț adițional, separat de prețul biletului. Așadar pentru o călătorie se vor cunoaște aeroportul de plecare, cel de sosire, cei doi piloți și avionul pe care aceștia îl vor opera, data și ora decolării și a sosirii. Deoarece se gestionează doar curse interne nu există curse cu escală și sunt necesari exact doi piloți. Unui zbor pot să îi fie atașați un anumit număr de însoțitori de zbor, însă nu este obligatoriu. În cazul în care există cel puțin un însoțitor de zbor, pentru zborul respectiv va fi desemnat un șef de cabină. Acesta va fi unul dintre însoțitorii de zbor din acea călătorie.

În cadrul acestui model de date vor fi folosite date reale pentru gestionarea aeroporturilor, a orașelor și a regiunilor. Pentru o călătorie ce pleacă dintr-un anumit aeroport se vor folosi doar piloți și însoțitori de zbor din respectivul aeroport și se va neglija întoarcerea acestora la aeroportul de plecare printr-un alt zbor. Pretul unui bilet pentru o călătorie va fi separat de prețul adițional pentru o clasă de zbor superioară, așadar prețul unui bilet va fi identic pentru toți pasagerii unei călătorii.

Diagrama Entitate-Relație

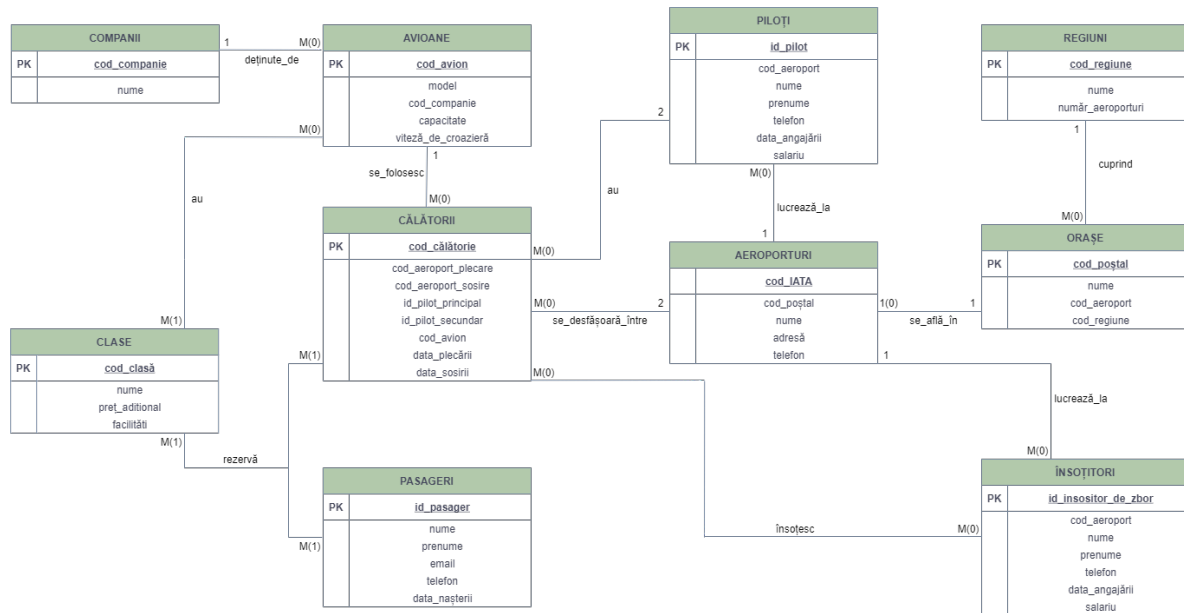


Figure 1: Diagrama Entitate-Relație

Diagrama Conceptuală

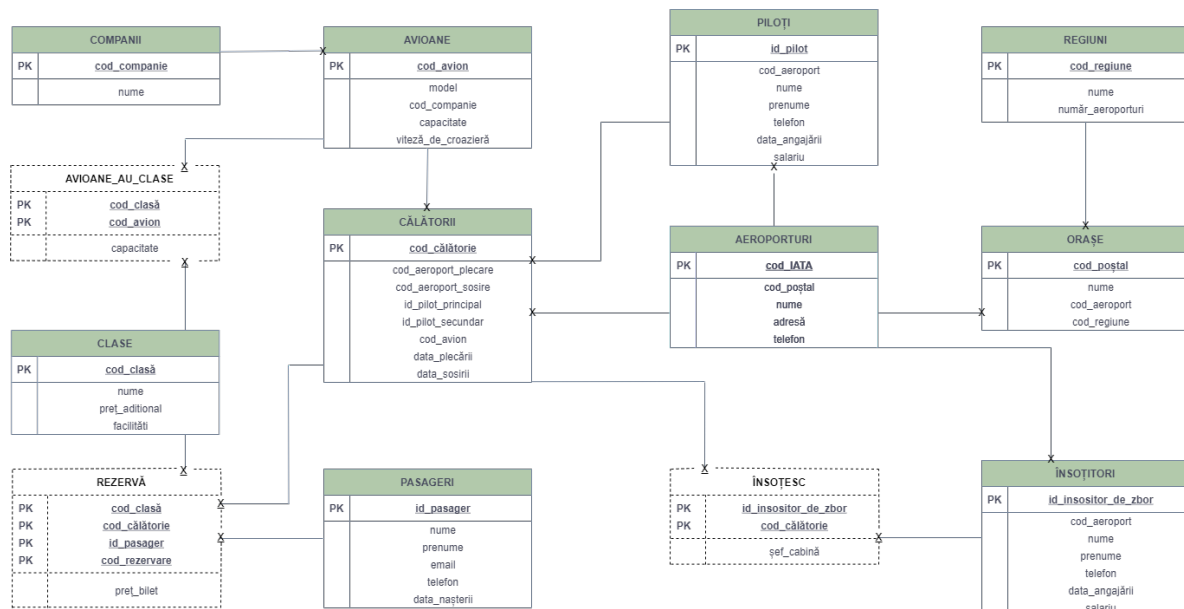


Figure 2: Diagrama Conceptuală

Crearea tabelelor și a constrângerilor

```
1  --- Vom crea cele 13 tabele n SQL si vom adauga constr ngerile pentru
    modelul de date
2
3  CREATE TABLE AEROPORTURI
4  (cod_IATA varchar2(3) CONSTRAINT aeroporturi_pk PRIMARY KEY,
5   cod_postal varchar2(6) NOT NULL,
6   nume varchar2(55) NOT NULL,
7   adresa varchar2(50),
8   telefon varchar2(10));
9
10 CREATE TABLE ORASE
11 (cod_postal varchar2(6) CONSTRAINT orase_pk PRIMARY KEY,
12 nume varchar2(15) NOT NULL,
13 cod_aeroport varchar2(3),
14 cod_regiune varchar2(3)); -- cheie externa din tabelul AEROPORTURI
15
16 -- odata ce a fost creat tabelul ORASE putem marca cheia externa in
    tabelul AEROPORTURI
17 ALTER TABLE AEROPORTURI
18 ADD CONSTRAINT aeroporturi_orase_fk FOREIGN KEY(cod_postal)
19 REFERENCES ORASE(cod_postal);
20
21 CREATE TABLE REGIUNI
22 (cod_regiune varchar2(3) CONSTRAINT regiuni_pk PRIMARY KEY,
23 nume varchar2(15) NOT NULL,
24 numar_aeroporturi number(2));
25
26 CREATE TABLE PILOTI
27 (id_pilot number(4) CONSTRAINT piloti_pk PRIMARY KEY,
28 cod_aeroport varchar2(3),
29 nume varchar2(30) NOT NULL,
30 prenume varchar2(30),
31 telefon varchar2(10) NOT NULL,
32 data_angajarii date DEFAULT sysdate,
33 salariu number(6),
34 CONSTRAINT piloti_aeroport_fk FOREIGN KEY(cod_aeroport)
35 REFERENCES AEROPORTURI(cod_IATA));
36
37 CREATE TABLE INSOTITORI
38 (id_insotitor_de_zbor number(4) CONSTRAINT insotitori_pk PRIMARY KEY,
39 cod_aeroport varchar2(3),
40 nume varchar2(30) NOT NULL,
41 prenume varchar2(30),
42 telefon varchar2(10) NOT NULL,
43 data_angajarii date DEFAULT sysdate,
44 salariu number(6),
45 CONSTRAINT insotitori_aeroport_fk FOREIGN KEY(cod_aeroport)
46 REFERENCES AEROPORTURI(cod_IATA));
47
48 CREATE TABLE CALATORII
49 (cod_calatorie number(6) CONSTRAINT calatorii_pk PRIMARY KEY,
50 cod_aeroport_plecare varchar2(3) NOT NULL,
51 cod_aeroport_sosire varchar2(3) NOT NULL,
52 id_pilot_principal number(4) NOT NULL,
53 id_pilot_secundar number(4) NOT NULL,
54 cod_avion number(4) NOT NULL,
55 data_plecarii date,
56 data_sosirii date,
57 CONSTRAINT calatorii_aeroport_fk1 FOREIGN KEY(cod_aeroport_plecare)
58 REFERENCES AEROPORTURI(cod_IATA),
59 CONSTRAINT calatorii_aeroport_fk2 FOREIGN KEY(cod_aeroport_sosire)
```

```

60 REFERENCES AEROPORTURI(cod_IATA),
61 -- cele doua coduri ale aeroporturilor vor fi chei externe din tabelul
   AEROPORTURI
62 CONSTRAINT calatorii_piloti_fk1 FOREIGN KEY(id_pilot_principal)
63 REFERENCES PILOTI(id_pilot),
64 CONSTRAINT calatorii_piloti_fk2 FOREIGN KEY(id_pilot_secundar)
65 REFERENCES PILOTI(id_pilot),
66 -- cele doua id-uri vor fi chei externe din tabelul PILOTI
67 CONSTRAINT valid_data_sosirii CHECK(data_sosirii > data_plecarii),
68 -- data sosirii trebuie sa fie mai mare decat data plecarii
69 CONSTRAINT distinct_aeroport CHECK (cod_aeroport_plecare !=
   cod_aeroport_sosire),
70 -- aeroportul de plecare si cel de sosire trebuie sa fie diferite
71 CONSTRAINT distinct_pilot CHECK (id_pilot_principal != id_pilot_secundar)
72 -- trebuie sa existe 2 piloti distincti
73 );
74
75 CREATE TABLE AVIOANE
76 (cod_avion number(4) CONSTRAINT avioane_pk PRIMARY KEY,
77 model varchar2(30) NOT NULL,
78 cod_companie number(4) NOT NULL,
79 capacitate number(3),
80 viteza_de_croaziera number(4));
81
82 -- dupa ce am creat tabelul AVIOANE putem sa referim cheia externa "
   cod_avion"
83 -- in tabelul CALATORII
84
85 ALTER TABLE CALATORII
86 ADD CONSTRAINT calatorii_avioane_fk FOREIGN KEY(cod_avion)
87 REFERENCES AVIOANE(cod_avion);
88
89 CREATE TABLE COMPANII
90 (cod_companie number(4) CONSTRAINT companii_pk PRIMARY KEY,
91 nume varchar2(30) NOT NULL);
92
93 -- dupa ce am creat tabelul COMPANII putem sa referim cheia externa
94 -- "cod_companie" in tabelul AVIOANE
95
96 ALTER TABLE AVIOANE
97 ADD CONSTRAINT avioane_companii_fk FOREIGN KEY(cod_companie)
98 REFERENCES COMPANII(cod_companie);
99
100 CREATE TABLE CLASE
101 (cod_clasa number(2) CONSTRAINT clase_pk PRIMARY KEY,
102 nume varchar2(20) NOT NULL,
103 pret_aditional number(4),
104 facilitati varchar2(60));
105
106 CREATE TABLE PASAGERI
107 (id_pasager number(6) CONSTRAINT pasageri_pk PRIMARY KEY,
108 nume varchar2(30) NOT NULL,
109 prenume varchar2(30),
110 email varchar2(30),
111 telefon varchar2(10) NOT NULL,
112 data_nasterii date);
113
114 CREATE TABLE AVIOANE_AU_CLASE
115 (cod_clasa number(2),
116 cod_avion number(4),
117 capacitate number(3),
118 CONSTRAINT avioane_au_clase_clase_fk FOREIGN KEY(cod_clasa)
119 REFERENCES CLASE(cod_clasa),

```

```

120 -- referim cheia externa din tabelul CLASE
121 CONSTRAINT avioane_au_clase_avioane_fk FOREIGN KEY(cod_avion)
122 REFERENCES AVIOANE(cod_avion),
123 -- referim cheia externa din tabelul AVIOANE
124 CONSTRAINT avioane_au_clase_pk PRIMARY KEY(cod_clasa, cod_avion));
125 -- cheie primara compusa
126
127 CREATE TABLE REZERVA
128 (cod_clasa number(2),
129 cod_calatorie number(6),
130 id_pasager number(6),
131 cod_rezervare number(8),
132 pret_bilet number(4),
133 CONSTRAINT rezerva_clase_fk FOREIGN KEY(cod_clasa)
134 REFERENCES CLASE(cod_clasa),
135 -- cheie externa din tabelul CLASE
136 CONSTRAINT rezerva_calatorii_fk FOREIGN KEY(cod_calatorie)
137 REFERENCES CALATORII(cod_calatorie),
138 -- cheie externa din tabelul CALATORII
139 CONSTRAINT rezerva_pasageri_fk FOREIGN KEY(id_pasager)
140 REFERENCES PASAGERI(id_pasager),
141 -- cheie externa din tabelul PASAGERI
142 CONSTRAINT rezerva_pk PRIMARY KEY(cod_clasa, cod_calatorie, id_pasager,
143 cod_rezervare));
144 -- cheie primara compusa din 4 elemente
145
146 CREATE TABLE INSOTESC
147 (id_insotitor_de_zbor number(4),
148 cod_calatorie number(6),
149 sef_cabina number(4),
150 CONSTRAINT insotesc_insotitori_fk1 FOREIGN KEY(id_insotitor_de_zbor)
151 REFERENCES INSOTITORI(id_insotitor_de_zbor),
152 -- cheie externa din tabelul INSOTITORI
153 CONSTRAINT insotesc_calatorii_fk FOREIGN KEY(cod_calatorie)
154 REFERENCES CALATORII(cod_calatorie),
155 -- cheie externa din tabelul CALATORII
156 CONSTRAINT insotesc_insotitori_fk2 FOREIGN KEY(sef_cabina)
157 REFERENCES INSOTITORI(id_insotitor_de_zbor),
158 -- cheie externa din tabelul INSOTITORI
159 CONSTRAINT insotesc_pk PRIMARY KEY(id_insotitor_de_zbor, cod_calatorie));
-- cheie primara compusa

```

Inserarea datelor

```
1 --- Inserari in tabelul REGIUNI
2 INSERT INTO REGIUNI
3 VALUES ('MUN', 'Muntenia', 1);
4
5 INSERT INTO REGIUNI
6 VALUES ('TRA', 'Transilvania', 2);
7
8 INSERT INTO REGIUNI
9 VALUES ('MOL', 'Moldova', 1);
10
11 INSERT INTO REGIUNI
12 VALUES ('DOB', 'Dobrogea', 1);
13
14 INSERT INTO REGIUNI
15 VALUES ('BAN', 'Banat', 1);
16
17 INSERT INTO REGIUNI
18 VALUES ('OLT', 'Oltenia', 1);
19
20 INSERT INTO REGIUNI
21 VALUES ('CRI', 'Crisana', 1);
22
23 INSERT INTO REGIUNI
24 VALUES ('MAR', 'Maramures', 0);
25
26 --- Inserari in tabelul ORASE
27 INSERT INTO ORASE
28 VALUES (075150, 'Bucuresti', 'OTP', 'MUN');
29
30 INSERT INTO ORASE
31 VALUES (400397, 'Cluj', 'CLJ', 'TRA');
32
33 INSERT INTO ORASE
34 VALUES (700750, 'Iasi', 'IAS', 'MOL');
35
36 INSERT INTO ORASE
37 VALUES (907195, 'Constanta', 'CND', 'DOB');
38
39 INSERT INTO ORASE
40 VALUES (307200, 'Timisoara', 'TSR', 'BAN');
41
42 INSERT INTO ORASE
43 VALUES (410223, 'Oradea', 'OMR', 'CRI');
44
45 INSERT INTO ORASE
46 VALUES (200621, 'Craiova', 'CRA', 'OLT');
47
48 INSERT INTO ORASE
49 VALUES (550052, 'Sibiu', 'SBZ', 'TRA');
50
51 INSERT INTO ORASE
52 VALUES (110001, 'Pitesti', NULL, 'MUN');
53
54 --- Inserari in tabelul AEROPORTURI
55 INSERT INTO AEROPORTURI
56 VALUES('OTP', 075150, 'Henri Coanda International Airport', 'Calea
    Bucurestilor 224E', '0212041000');
57
58 INSERT INTO AEROPORTURI
59 VALUES('CLJ', 400397, 'Avram Iancu International Airport', 'Strada Traian
    Vuia 149-151', '0264307500');
```

```

60
61 INSERT INTO AEROPORTURI
62 VALUES('IAS', 700750, 'Iasi International Airport', 'Strada Moara de Vant
    34', '0232271590');
63
64 INSERT INTO AEROPORTURI
65 VALUES('CND', 907195, 'Mihail Kogalniceanu International Airport', '
    Strada Tudor Vladimirescu', '0241255100');
66
67 INSERT INTO AEROPORTURI
68 VALUES('TSR', 307200, 'Traian Vuia International Airport', 'Strada
    Aeroportului 2', '0256386089');
69
70 INSERT INTO AEROPORTURI
71 VALUES('OMR', 410223, 'Oradea International Airport', 'Calea Aradului 80'
    , '0259416082');
72
73 INSERT INTO AEROPORTURI
74 VALUES('CRA', 200621, 'Craiova International Airport', 'Calea Bucuresti
    325', '0754022508');
75
76 INSERT INTO AEROPORTURI
77 VALUES('SBZ', 550052, 'Sibiu International Airport', 'Soseaua Alba Iulia
    73', '0269253135');
78
79 --- Adaugam constrangerea de FOREIGN KEY pentru cod_aeroporturi din
    tabelul ORASE
80 ALTER TABLE ORASE
81 ADD CONSTRAINT orase_aeroporturi_fk FOREIGN KEY(cod_aeroport)
82 REFERENCES AEROPORTURI(cod_IATA);
83
84 --- Adaugam constrangerea de FOREIGN KEY pentru cod_regiune din tabelul
    ORASE
85 ALTER TABLE ORASE
86 ADD CONSTRAINT orase_regiuni_fk FOREIGN KEY(cod_regiune)
87 REFERENCES REGIUNI(cod_regiune);
88
89 --- secventa pentru incrementarea atributului ID_PILOT din tabelul PILOTI
90 CREATE SEQUENCE SEQ_PILOTI
91 START WITH 120
92 INCREMENT by 10
93 MAXVALUE 9990
94 NOCYCLE;
95
96 --- Inserari in tabelul PILOTI
97 INSERT INTO PILOTI
98 VALUES (SEQ_PILOTI.NEXTVAL, 'OTP', 'Popescu', 'Mihai', '0723467897',
    to_date('04-05-2018','DD-MM-YYYY'), 100000);
99
100 INSERT INTO PILOTI
101 VALUES (SEQ_PILOTI.NEXTVAL, 'OTP', 'Dumitru', 'Andrei', '0773567890',
    to_date('07-09-2018','DD-MM-YYYY'), 150000);
102
103 INSERT INTO PILOTI
104 VALUES (SEQ_PILOTI.NEXTVAL, 'TSR', 'Marinescu', 'Adrian', '0793857856',
    to_date('05-02-2016','DD-MM-YYYY'), 550000);
105
106 INSERT INTO PILOTI
107 VALUES (SEQ_PILOTI.NEXTVAL, 'TSR', 'Coman', 'Stefania', '0775852235',
    to_date('05-12-2020','DD-MM-YYYY'), 200000);
108
109 INSERT INTO PILOTI
110 VALUES (SEQ_PILOTI.NEXTVAL, 'CLJ', 'Florescu', 'Mihai', '0788451775',

```



```

    to_date('07-12-2010','DD-MM-YYYY'), 500000);
111
112 INSERT INTO PILOTI
113 VALUES (SEQ_PILOTI.NEXTVAL, 'CLJ', 'Moise', 'Horia', '0788449075',
    to_date('23-06-2015','DD-MM-YYYY'), 350000);
114
115 INSERT INTO PILOTI
116 VALUES (SEQ_PILOTI.NEXTVAL, 'CLJ', 'Stoica', 'Irina', '0702390175',
    to_date('15-01-2021','DD-MM-YYYY'), 180000);
117
118 INSERT INTO PILOTI
119 VALUES (SEQ_PILOTI.NEXTVAL, 'IAS', 'Naum', 'Florin', '0712345175',
    to_date('15-11-2019','DD-MM-YYYY'), 250000);
120
121 INSERT INTO PILOTI
122 VALUES (SEQ_PILOTI.NEXTVAL, 'IAS', 'Olteanu', 'Paul', '0712743275',
    to_date('01-01-2014','DD-MM-YYYY'), 550000);
123
124 INSERT INTO PILOTI
125 VALUES (SEQ_PILOTI.NEXTVAL, 'CND', 'Avram', 'Daniel', '0766320275',
    to_date('01-06-2020','DD-MM-YYYY'), 222000);
126
127 INSERT INTO PILOTI
128 VALUES (SEQ_PILOTI.NEXTVAL, 'CND', 'Stefan', 'George', '0766320275',
    to_date('30-08-2013','DD-MM-YYYY'), 355000);
129
130 INSERT INTO PILOTI
131 VALUES (SEQ_PILOTI.NEXTVAL, 'CND', 'Stancu', 'Elena', '0733332146',
    to_date('01-05-2021','DD-MM-YYYY'), 99000);
132
133 INSERT INTO PILOTI
134 VALUES (SEQ_PILOTI.NEXTVAL, 'CRA', 'Mihailescu', 'Cosmin', '0754999146',
    to_date('01-05-2018','DD-MM-YYYY'), 340000);
135
136 INSERT INTO PILOTI
137 VALUES (SEQ_PILOTI.NEXTVAL, 'CRA', 'Olteanu', 'Valentin', '0765412146',
    to_date('17-04-2010','DD-MM-YYYY'), 450000);
138
139 --- secventa pentru incrementarea atributului ID_insotitor_DE_ZBOR din
    tabelul INSOTITORI
140 CREATE SEQUENCE SEQ_INSOTITORI
141 START WITH 10
142 INCREMENT by 10
143 MINVALUE 0
144 MAXVALUE 9990
145 NOCYCLE;
146
147 ---Inserari in tabelul INSOTITORI
148 INSERT INTO INSOTITORI
149 VALUES (SEQ_INSOTITORI.NEXTVAL, 'OTP', 'Voiculescu', 'Alina', '0745678146',
    to_date('04-10-2018','DD-MM-YYYY'), 33000);
150
151 INSERT INTO INSOTITORI
152 VALUES (SEQ_INSOTITORI.NEXTVAL, 'OTP', 'Moiescu', 'Denisa', '0792218597',
    to_date('13-03-2019','DD-MM-YYYY'), 40000);
153
154 INSERT INTO INSOTITORI
155 VALUES (SEQ_INSOTITORI.NEXTVAL, 'OTP', 'Ionescu', 'Florin', '0726688610',
    to_date('02-04-2012','DD-MM-YYYY'), 90000);
156
157 INSERT INTO INSOTITORI
158 VALUES (SEQ_INSOTITORI.NEXTVAL, 'CLJ', 'Georgescu', 'Calin', '0768216950',
    to_date('01-07-2016','DD-MM-YYYY'), 80000);

```

```

159
160 INSERT INTO INSOTITORI
161 VALUES (SEQ_INSOTITORI.NEXTVAL, 'CLJ', 'Antonie', 'Mihaela', '0783717731'
, to_date('06-12-2020', 'DD-MM-YYYY'), 15000);
162
163 INSERT INTO INSOTITORI
164 VALUES (SEQ_INSOTITORI.NEXTVAL, 'TSR', 'Voica', 'Laura', '0766138430',
to_date('31-01-2021', 'DD-MM-YYYY'), 13000);
165
166 INSERT INTO INSOTITORI
167 VALUES (SEQ_INSOTITORI.NEXTVAL, 'TSR', 'Enache', 'Maria', '0778767173',
to_date('25-10-2015', 'DD-MM-YYYY'), 40000);
168
169 INSERT INTO INSOTITORI
170 VALUES (SEQ_INSOTITORI.NEXTVAL, 'IAS', 'Matache', 'Irina', '0762706994',
to_date('20-03-2015', 'DD-MM-YYYY'), 45000);
171
172 INSERT INTO INSOTITORI
173 VALUES (SEQ_INSOTITORI.NEXTVAL, 'IAS', 'Mirea', 'Mihai', '0785072934',
to_date('07-07-2019', 'DD-MM-YYYY'), 27000);
174
175 INSERT INTO INSOTITORI
176 VALUES (SEQ_INSOTITORI.NEXTVAL, 'CND', 'Lazar', 'Nicoleta', '0718538567',
to_date('09-11-2018', 'DD-MM-YYYY'), 33000);
177
178 INSERT INTO INSOTITORI
179 VALUES (SEQ_INSOTITORI.NEXTVAL, 'CND', 'Florea', 'Stefania', '0784725925'
, to_date('23-09-2009', 'DD-MM-YYYY'), 80000);
180
181 INSERT INTO INSOTITORI
182 VALUES (SEQ_INSOTITORI.NEXTVAL, 'CRA', 'Miulescu', 'Ana', '0758812546',
to_date('06-06-2020', 'DD-MM-YYYY'), 12000);
183
184 INSERT INTO INSOTITORI
185 VALUES (SEQ_INSOTITORI.NEXTVAL, 'SBZ', 'Diaconescu', 'Daniela', '
0734456845', to_date('27-04-2018', 'DD-MM-YYYY'), 16000);
186
187 ---Inserari in tabelul COMPANII
188 INSERT INTO COMPANII
189 VALUES (120, 'Tarom');
190
191 INSERT INTO COMPANII
192 VALUES (121, 'Blue Air');
193
194 INSERT INTO COMPANII
195 VALUES (122, 'Wizz Air');
196
197 INSERT INTO COMPANII
198 VALUES (123, 'KLM');
199
200 INSERT INTO COMPANII
201 VALUES (124, 'Ryanair');
202
203 INSERT INTO COMPANII
204 VALUES (125, 'Lufthansa');
205
206 ---Inserari in tabelul AVIOANE
207 INSERT INTO AVIOANE
208 VALUES (1000, 'BOEING 737-700', 120, 149, 870);
209
210 INSERT INTO AVIOANE
211 VALUES (1001, 'BOEING 737-300', 120, 141, 800);
212

```

```

213 INSERT INTO AVIOANE
214 VALUES (1002, 'AIRBUS A318-111', 120, 132, 850);
215
216 INSERT INTO AVIOANE
217 VALUES (1003, 'ATR 42-500', 120, 52, 550);
218
219 INSERT INTO AVIOANE
220 VALUES (1004, 'BOEING 737-300', 121, 141, 800);
221
222 INSERT INTO AVIOANE
223 VALUES (1005, 'BOEING 737-500', 121, 126, 785);
224
225 INSERT INTO AVIOANE
226 VALUES (1006, 'BOEING 737-800', 121, 189, 823);
227
228 INSERT INTO AVIOANE
229 VALUES (1007, 'AIRBUS A320-200', 122, 186, 900);
230
231 INSERT INTO AVIOANE
232 VALUES (1008, 'AIRBUS A321-200', 122, 230, 840);
233
234 INSERT INTO AVIOANE
235 VALUES (1009, 'AIRBUS A330-200', 123, 185, 913);
236
237 INSERT INTO AVIOANE
238 VALUES (1010, 'BOEING 737-800', 124, 162, 946);
239
240 --- secventa pentru incrementarea atributului COD_CALATORIE din tabelul
    CALATORII
241 CREATE SEQUENCE SEQ_CALATORII
242 START WITH 1001
243 INCREMENT by 1
244 MAXVALUE 999999
245 NOCYCLE;
246
247 ---Inserari in tabelul CALATORII
248 INSERT INTO CALATORII
249 VALUES (SEQ_CALATORII.NEXTVAL, 'OTP', 'CLJ', 120, 130, 1006, to_date('
    01-06-2021 07:30','DD-MM-YYYY HH24:MI'), to_date('01-06-2021 08:25','
    DD-MM-YYYY HH24:MI'));
250
251 INSERT INTO CALATORII
252 VALUES (SEQ_CALATORII.NEXTVAL, 'OTP', 'IAS', 120, 130, 1002, to_date('
    02-06-2021 08:00','DD-MM-YYYY HH24:MI'), to_date('02-06-2021 09:05','
    DD-MM-YYYY HH24:MI'));
253
254 INSERT INTO CALATORII
255 VALUES (SEQ_CALATORII.NEXTVAL, 'OTP', 'TSR', 120, 130, 1000, to_date('
    03-06-2021 21:00','DD-MM-YYYY HH24:MI'), to_date('03-06-2021 22:00','
    DD-MM-YYYY HH24:MI'));
256
257 INSERT INTO CALATORII
258 VALUES (SEQ_CALATORII.NEXTVAL, 'OTP', 'TSR', 120, 130, 1010, to_date('
    04-06-2021 07:30','DD-MM-YYYY HH24:MI'), to_date('04-06-2021 08:30','
    DD-MM-YYYY HH24:MI'));
259
260 INSERT INTO CALATORII
261 VALUES (SEQ_CALATORII.NEXTVAL, 'CLJ', 'IAS', 160, 180, 1003, to_date('
    27-05-2021 16:00','DD-MM-YYYY HH24:MI'), to_date('27-05-2021 16:55','
    DD-MM-YYYY HH24:MI'));
262
263 INSERT INTO CALATORII
264 VALUES (SEQ_CALATORII.NEXTVAL, 'CLJ', 'CND', 160, 170, 1008, to_date('

```

```

30-05-2021 23:55','DD-MM-YYYY HH24:MI'), to_date('31-05-2021 00:30','
DD-MM-YYYY HH24:MI'));
265
266 INSERT INTO CALATORII
267 VALUES (SEQ_CALATORII.NEXTVAL, 'IAS', 'CRA', 190, 200, 1009, to_date('
04-06-2021 13:00','DD-MM-YYYY HH24:MI'), to_date('04-06-2021 14:10','
DD-MM-YYYY HH24:MI'));
268
269 INSERT INTO CALATORII
270 VALUES (SEQ_CALATORII.NEXTVAL, 'IAS', 'OTP', 190, 200, 1002, to_date('
05-06-2021 05:30','DD-MM-YYYY HH24:MI'), to_date('05-06-2021 06:40','
DD-MM-YYYY HH24:MI'));
271
272 INSERT INTO CALATORII
273 VALUES (SEQ_CALATORII.NEXTVAL, 'CRA', 'IAS', 240, 250, 1010, to_date('
04-06-2021 19:30','DD-MM-YYYY HH24:MI'), to_date('04-06-2021 20:40','
DD-MM-YYYY HH24:MI'));
274
275 INSERT INTO CALATORII
276 VALUES (SEQ_CALATORII.NEXTVAL, 'CRA', 'CLJ', 240, 250, 1001, to_date('
08-06-2021 09:45','DD-MM-YYYY HH24:MI'), to_date('08-06-2021 10:45','
DD-MM-YYYY HH24:MI'));
277
278 INSERT INTO CALATORII
279 VALUES (SEQ_CALATORII.NEXTVAL, 'TSR', 'OTP', 140, 150, 1005, to_date('
30-05-2021 07:40','DD-MM-YYYY HH24:MI'), to_date('30-05-2021 08:55','
DD-MM-YYYY HH24:MI'));
280
281 INSERT INTO CALATORII
282 VALUES (SEQ_CALATORII.NEXTVAL, 'TSR', 'CND', 140, 150, 1007, to_date('
28-05-2021 12:00','DD-MM-YYYY HH24:MI'), to_date('28-05-2021 13:45','
DD-MM-YYYY HH24:MI'));
283
284 INSERT INTO CALATORII
285 VALUES (SEQ_CALATORII.NEXTVAL, 'OTP', 'OMR', 120, 130, 1003, to_date('
11-06-2021 05:55','DD-MM-YYYY HH24:MI'), to_date('11-06-2021 07:20','
DD-MM-YYYY HH24:MI'));
286
287 ---Inserari in tabelul CLASE
288 INSERT INTO CLASE
289 VALUES (10, 'Economica', NULL, 'masuta retractabila');
290
291 INSERT INTO CLASE
292 VALUES (20, 'Business', 300, 'scaune ajustabile, o masa calda');
293
294 INSERT INTO CLASE
295 VALUES (30, 'First class', 500, 'lounge, meniu variat, paturi');
296
297 INSERT INTO CLASE
298 VALUES (40, 'Business Premium', 800, 'facilitati business, lounge');
299
300 INSERT INTO CLASE
301 VALUES (50, 'First Class Premium', 1000, 'facilitati first class, meniu
lux');
302
303 --- secventa pentru incrementarea atributului ID_PASAGER din tabelul
PASAGERI
304 CREATE SEQUENCE SEQ_PASAGERI
305 START WITH 1
306 INCREMENT by 1
307 MAXVALUE 999999
308 NOCYCLE;
309

```

```

310 ---Inserari in tabelul PASAGERI
311 INSERT INTO PASAGERI
312 VALUES (SEQ_PASAGERI.NEXTVAL, 'Ionita', 'Alexandra', 'aionita@yahoo.com',
          '0741974529', to_date('03-05-2001','dd-mm-yyyy'));
313
314 INSERT INTO PASAGERI
315 VALUES (SEQ_PASAGERI.NEXTVAL, 'Moldoveanu', 'Victor', 'victor_m@yahoo.com
          ', '0785483431', to_date('08-03-1980','dd-mm-yyyy'));
316
317 INSERT INTO PASAGERI
318 VALUES (SEQ_PASAGERI.NEXTVAL, 'Stoica', 'Valentin', 'valistoica@gmail.com
          ', '0791459679', to_date('14-12-1996','dd-mm-yyyy'));
319
320 INSERT INTO PASAGERI
321 VALUES (SEQ_PASAGERI.NEXTVAL, 'Dabija', 'Adrian', 'dabadi@yahoo.com', '
          0760523629', to_date('06-07-1990','dd-mm-yyyy'));
322
323 INSERT INTO PASAGERI
324 VALUES (SEQ_PASAGERI.NEXTVAL, 'Barbu', 'Dumitru', 'dumitruB@yahoo.com', '
          0746888320', to_date('13-05-2000','dd-mm-yyyy'));
325
326 INSERT INTO PASAGERI
327 VALUES (SEQ_PASAGERI.NEXTVAL, 'Teodorescu', 'Eugen', 'eugen_t@yahoo.com',
          '0770522210', to_date('23-11-1965','dd-mm-yyyy'));
328
329 INSERT INTO PASAGERI
330 VALUES (SEQ_PASAGERI.NEXTVAL, 'Lupu', 'Maria', 'maria_lupu@yahoo.com', '
          0757241888', to_date('03-12-1977','dd-mm-yyyy'));
331
332 INSERT INTO PASAGERI
333 VALUES (SEQ_PASAGERI.NEXTVAL, 'Sava', 'Florentina', 'savaflo@gmail.com',
          '0766560779', to_date('29-06-2002','dd-mm-yyyy'));
334
335 INSERT INTO PASAGERI
336 VALUES (SEQ_PASAGERI.NEXTVAL, 'Nistor', 'Mariana', 'mariananistor@gmail.
          com', '0745582866', to_date('03-05-2001','dd-mm-yyyy'));
337
338 INSERT INTO PASAGERI
339 VALUES (SEQ_PASAGERI.NEXTVAL, 'Barbu', 'Ana Maria', 'anamariab@yahoo.com'
          , '0788821557', to_date('31-01-1992','dd-mm-yyyy'));
340
341 INSERT INTO PASAGERI
342 VALUES (SEQ_PASAGERI.NEXTVAL, 'Dumitrescu', 'Andreea', '
          andreea_dumi@yahoo.com', '0749267946', to_date('07-07-1979','dd-mm-
          yyyy'));
343
344 INSERT INTO PASAGERI
345 VALUES (SEQ_PASAGERI.NEXTVAL, 'Gheorghiu', 'Mihaela', 'mihaela.g@yahoo.
          com', '0727774485', to_date('24-11-1969','dd-mm-yyyy'));
346
347 INSERT INTO PASAGERI
348 VALUES (SEQ_PASAGERI.NEXTVAL, 'Pop', 'Rodica', 'rodica_pop@yahoo.com', '
          0791567138', to_date('16-04-1999','dd-mm-yyyy'));
349
350 INSERT INTO PASAGERI
351 VALUES (SEQ_PASAGERI.NEXTVAL, 'Constantinescu', 'Roxana', 'roxana.c@yahoo
          .com', '0743479564', to_date('12-12-1982','dd-mm-yyyy'));
352
353 INSERT INTO PASAGERI
354 VALUES (SEQ_PASAGERI.NEXTVAL, 'Dobre', 'Gabriela', 'gabi_dobre@gmail.com'
          , '0765870667', to_date('28-02-1997','dd-mm-yyyy'));
355
356 INSERT INTO PASAGERI

```

```

357 VALUES (SEQ_PASAGERI.NEXTVAL, 'Ciobanu', 'Adriana', 'a.ciobanu@yahoo.com'
358 , '0732936930', to_date('03-11-1960','dd-mm-yyyy'));
359
359 INSERT INTO PASAGERI
360 VALUES (SEQ_PASAGERI.NEXTVAL, 'Toma', 'Claudia', 'claudia.toma@yahoo.com'
361 , '0736630215', to_date('16-09-1995','dd-mm-yyyy'));
362
362 INSERT INTO PASAGERI
363 VALUES (SEQ_PASAGERI.NEXTVAL, 'Stanescu', 'Constantin', '
364 ctin_stanescu@gmail.com', '0756729015', to_date('08-08-1955','dd-mm-
365 yyyy'));
366
365 INSERT INTO PASAGERI
366 VALUES (SEQ_PASAGERI.NEXTVAL, 'Dima', 'Gheorghe', 'dima.gh@gmail.com', '
367 0774034758', to_date('16-04-1968','dd-mm-yyyy'));
368
368 INSERT INTO PASAGERI
369 VALUES (SEQ_PASAGERI.NEXTVAL, 'Dumitrescu', 'Cristina', 'cristinaD@gmail.
370 com', '0784550324', to_date('15-05-2000','dd-mm-yyyy'));
371
371 INSERT INTO PASAGERI
372 VALUES (SEQ_PASAGERI.NEXTVAL, 'Popescu', 'Elena', 'elenapop@yahoo.com', '
373 0781236689', to_date('01-06-1985','dd-mm-yyyy'));
374
374 ---Inserari in tabelul AVIOANE_AU_CLASE
375 INSERT INTO AVIOANE_AU_CLASE
376 VALUES (10, 1000, 100);
377
378 INSERT INTO AVIOANE_AU_CLASE
379 VALUES (20, 1000, 49);
380
381 INSERT INTO AVIOANE_AU_CLASE
382 VALUES (10, 1001, 141);
383
384 INSERT INTO AVIOANE_AU_CLASE
385 VALUES (10, 1002, 132);
386
387 INSERT INTO AVIOANE_AU_CLASE
388 VALUES (10, 1003, 52);
389
390 INSERT INTO AVIOANE_AU_CLASE
391 VALUES (10, 1004, 80);
392
393 INSERT INTO AVIOANE_AU_CLASE
394 VALUES (20, 1004, 61);
395
396 INSERT INTO AVIOANE_AU_CLASE
397 VALUES (10, 1005, 126);
398
399 INSERT INTO AVIOANE_AU_CLASE
400 VALUES (10, 1006, 100);
401
402 INSERT INTO AVIOANE_AU_CLASE
403 VALUES (20, 1006, 40);
404
405 INSERT INTO AVIOANE_AU_CLASE
406 VALUES (30, 1006, 49);
407
408 INSERT INTO AVIOANE_AU_CLASE
409 VALUES (10, 1007, 160);
410
411 INSERT INTO AVIOANE_AU_CLASE
412 VALUES (20, 1007, 26);

```

```

413
414 INSERT INTO AVIOANE_AU_CLASE
415 VALUES (10, 1008, 100);
416
417 INSERT INTO AVIOANE_AU_CLASE
418 VALUES (20, 1008, 100);
419
420 INSERT INTO AVIOANE_AU_CLASE
421 VALUES (30, 1008, 20);
422
423 INSERT INTO AVIOANE_AU_CLASE
424 VALUES (40, 1008, 10);
425
426 INSERT INTO AVIOANE_AU_CLASE
427 VALUES (10, 1009, 185);
428
429 INSERT INTO AVIOANE_AU_CLASE
430 VALUES (10, 1010, 162);
431
432 ---Inserari in tabelul REZERVA
433 INSERT INTO REZERVA
434 VALUES (30, 1001, 2, 100001, 300);
435
436 INSERT INTO REZERVA
437 VALUES (30, 1001, 3, 100002, 300);
438
439 INSERT INTO REZERVA
440 VALUES (10, 1001, 15, 100003, 300);
441
442 INSERT INTO REZERVA
443 VALUES (10, 1010, 7, 100004, 200);
444
445 INSERT INTO REZERVA
446 VALUES (10, 1010, 10, 100005, 200);
447
448 INSERT INTO REZERVA
449 VALUES (10, 1010, 11, 100006, 200);
450
451 INSERT INTO REZERVA
452 VALUES (20, 1006, 5, 100007, 250);
453
454 INSERT INTO REZERVA
455 VALUES (40, 1006, 12, 100008, 250);
456
457 INSERT INTO REZERVA
458 VALUES (40, 1006, 13, 100009, 250);
459
460 INSERT INTO REZERVA
461 VALUES (30, 1006, 14, 100010, 250);
462
463 INSERT INTO REZERVA
464 VALUES (20, 1012, 1, 100011, 250);
465
466 INSERT INTO REZERVA
467 VALUES (10, 1012, 17, 100012, 250);
468
469 INSERT INTO REZERVA
470 VALUES (10, 1012, 18, 100013, 250);
471
472 INSERT INTO REZERVA
473 VALUES (10, 1012, 20, 100014, 250);
474
475 INSERT INTO REZERVA

```

```

476 VALUES (10, 1004, 4, 100015, 270);
477
478 INSERT INTO REZERVA
479 VALUES (10, 1004, 6, 100016, 270);
480
481 INSERT INTO REZERVA
482 VALUES (10, 1003, 16, 100017, 150);
483
484 INSERT INTO REZERVA
485 VALUES (20, 1003, 8, 100018, 150);
486
487 INSERT INTO REZERVA
488 VALUES (20, 1003, 9, 100019, 150);
489
490 INSERT INTO REZERVA
491 VALUES (10, 1013, 19, 100020, 150);
492
493 INSERT INTO REZERVA
494 VALUES (10, 1007, 10, 100021, 240);
495
496 INSERT INTO REZERVA
497 VALUES (10, 1007, 11, 100022, 240);
498
499 INSERT INTO REZERVA
500 VALUES (10, 1011, 2, 100023, 350);
501
502 INSERT INTO REZERVA
503 VALUES (10, 1011, 3, 100024, 350);
504
505 INSERT INTO REZERVA
506 VALUES (10, 1008, 19, 100025, 275);
507
508 INSERT INTO REZERVA
509 VALUES (10, 1008, 17, 100026, 275);
510
511 INSERT INTO REZERVA
512 VALUES (10, 1008, 18, 100027, 275);
513
514 ---Inserari in tabelul INSOTESC
515 INSERT INTO INSOTESC
516 VALUES (10, 1001, 20);
517
518 INSERT INTO INSOTESC
519 VALUES (20, 1001, NULL);
520
521 INSERT INTO INSOTESC
522 VALUES (30, 1001, 20);
523
524 INSERT INTO INSOTESC
525 VALUES (10, 1003, NULL);
526
527 INSERT INTO INSOTESC
528 VALUES (30, 1003, 10);
529
530 INSERT INTO INSOTESC
531 VALUES (10, 1004, 20);
532
533 INSERT INTO INSOTESC
534 VALUES (20, 1004, NULL);
535
536 INSERT INTO INSOTESC
537 VALUES (40, 1005, NULL);
538

```



```
539 INSERT INTO INSOTESC
540 VALUES (40, 1006, NULL);
541
542 INSERT INTO INSOTESC
543 VALUES (50, 1006, 40);
544
545 INSERT INTO INSOTESC
546 VALUES (80, 1007, 90);
547
548 INSERT INTO INSOTESC
549 VALUES (90, 1007, NULL);
550
551 INSERT INTO INSOTESC
552 VALUES (90, 1008, NULL);
553
554 INSERT INTO INSOTESC
555 VALUES (120, 1009, NULL);
556
557 INSERT INTO INSOTESC
558 VALUES (120, 1010, NULL);
559
560 INSERT INTO INSOTESC
561 VALUES (60, 1011, NULL);
562
563 INSERT INTO INSOTESC
564 VALUES (70, 1011, 60);
565
566 INSERT INTO INSOTESC
567 VALUES (60, 1012, 70);
568
569 INSERT INTO INSOTESC
570 VALUES (70, 1012, NULL);
571
572 INSERT INTO INSOTESC
573 VALUES (20, 1013, NULL);
574
575 INSERT INTO INSOTESC
576 VALUES (30, 1013, 20);
577
578 COMMIT;
```

AEROPORTURI:

✂	COD_IATA	✂	COD_POSTAL	✂	NUME	✂	ADRESA	✂	TELEFON
1	OTP		75150		Henri Coanda International Airport		Calea Bucurestilor 224E		0212041000
2	CLJ		400397		Avram Iancu International Airport		Strada Traian Vuia 149-151		0264307500
3	IAS		700750		Iasi International Airport		Strada Moara de Vant 34		0232271590
4	CND		907195		Mihail Kogalniceanu International Airport		Strada Tudor Vladimirescu		0241255100
5	TSR		307200		Traian Vuia International Airport		Strada Aeroportului 2		0256386089
6	OMR		410223		Oradea International Airport		Calea Aradului 80		0259416082
7	CRA		200621		Craiova International Airport		Calea Bucuresti 325		0754022508
8	SBZ		550052		Sibiu International Airport		Soseaua Alba Iulia 73		0269253135

ORAȘE:

REGIUNI:

✂	COD_POSTAL	✂	NUME	✂	COD_AEROPORT	✂	COD_REGIUNE	✂	COD_REGIUNE	✂	NUME	✂	NUMAR_AEROPORTURI
1	75150		Bucuresti		OTP		MUN		1		Muntenia		1
2	400397		Cluj		CLJ		TRA		2		Transilvania		2
3	700750		Iasi		IAS		MOL		3		Moldova		1
4	907195		Constanta		CND		DOB		4		Dobrogea		1
5	307200		Timisoara		TSR		BAN		5		Banat		1
6	410223		Oradea		OMR		CRI		6		Oltenia		1
7	200621		Craiova		CRA		OLT		7		Crisana		1
8	550052		Sibiu		SBZ		TRA		8		Maramures		0
9	110001		Pitesti		(null)		MUN						

PILOȚI:

	✂	ID_...	✂	COD_AEROPORT	✂	NUME	✂	PRENUME	✂	TELEFON	✂	DATA_ANGAJARII	✂	SALARIU
1		120		OTP		Popescu		Mihai		0723467897		04-MAY-18		100000
2		130		OTP		Dumitru		Andrei		0773567890		07-SEP-18		150000
3		140		TSR		Marinescu		Adrian		0793857856		05-FEB-16		550000
4		150		TSR		Coman		Stefania		0775852235		05-DEC-20		200000
5		160		CLJ		Florescu		Mihai		0788451775		07-DEC-10		500000
6		170		CLJ		Moise		Horia		0788449075		23-JUN-15		350000
7		180		CLJ		Stoica		Irina		0702390175		15-JAN-21		180000
8		190		IAS		Naum		Florin		0712345175		15-NOV-19		250000
9		200		IAS		Olteanu		Paul		0712743275		01-JAN-14		550000
10		210		CND		Avram		Daniel		0766320275		01-JUN-20		222000
11		220		CND		Stefan		George		0766320275		30-AUG-13		355000
12		230		CND		Stancu		Elena		0733332146		01-MAY-21		99000
13		240		CRA		Mihailescu		Cosmin		0754999146		01-MAY-18		340000
14		250		CRA		Olteanu		Valentin		0765412146		17-APR-10		450000

CĂLĂTORII:

	✂ COD_CALATORIE	✂ COD_AEROPORT_PLECARE	✂ COD_AEROPORT_SOSIRE	✂ ID_PILOT_PRINCIPAL	✂ ID_PILOT_SECUNDAR	✂ COD_AVION	✂ DATA_PLECarii	✂ DATA_SOSIRII
1	1001	OTP	CLJ	120	130	1006	01-JUN-21	01-JUN-21
2	1002	OTP	IAS	120	130	1002	02-JUN-21	02-JUN-21
3	1003	OTP	TSR	120	130	1000	03-JUN-21	03-JUN-21
4	1004	OTP	TSR	120	130	1010	04-JUN-21	04-JUN-21
5	1005	CLJ	IAS	160	180	1003	27-MAY-21	27-MAY-21
6	1006	CLJ	CND	160	170	1008	30-MAY-21	31-MAY-21
7	1007	IAS	CRA	190	200	1009	04-JUN-21	04-JUN-21
8	1008	IAS	OTP	190	200	1002	05-JUN-21	05-JUN-21
9	1009	CRA	IAS	240	250	1010	04-JUN-21	04-JUN-21
10	1010	CRA	CLJ	240	250	1001	08-JUN-21	08-JUN-21
11	1011	TSR	OTP	140	150	1005	30-MAY-21	30-MAY-21
12	1012	TSR	CND	140	150	1007	28-MAY-21	28-MAY-21
13	1013	OTP	OMR	120	130	1003	11-JUN-21	11-JUN-21

ÎNȘOȚITORI:

	✂ ID_INSOTITOR_DE_ZBOR	✂ COD_AEROPORT	✂ NUME	✂ PRENUME	✂ TELEFON	✂ DATA_ANGAJARII	✂ SALARIU
1	10	OTP	Voiculescu	Alina	0745678146	04-OCT-18	33000
2	20	OTP	Moiescu	Denisa	0792218597	13-MAR-19	40000
3	30	OTP	Ionescu	Florin	0726688610	02-APR-12	90000
4	40	CLJ	Georgescu	Calin	0768216950	01-JUL-16	80000
5	50	CLJ	Antonie	Mihaela	0783717731	06-DEC-20	15000
6	60	TSR	Voica	Laura	0766138430	31-JAN-21	13000
7	70	TSR	Enache	Maria	0778767173	25-OCT-15	40000
8	80	IAS	Matache	Irina	0762706994	20-MAR-15	45000
9	90	IAS	Mirea	Mihai	0785072934	07-JUL-19	27000
10	100	CND	Lazar	Nicoleta	0718538567	09-NOV-18	33000
11	110	CND	Florea	Stefania	0784725925	23-SEP-09	80000
12	120	CRA	Miulescu	Ana	0758812546	06-JUN-20	12000
13	130	SBZ	Diaconescu	Daniela	0734456845	27-APR-18	16000

PASAGERI:

	✂ ID_PASAGER	✂ NUME	✂ PRENUME	✂ EMAIL	✂ TELEFON	✂ DATA_NASTERII
1	1	Ionita	Alexandra	aionita@yahoo.com	0741974529	03-MAY-01
2	2	Moldoveanu	Victor	victor_m@yahoo.com	0785483431	08-MAR-80
3	3	Stoica	Valentin	valistoica@gmail.com	0791459679	14-DEC-96
4	4	Dabija	Adrian	dabadi@yahoo.com	0760523629	06-JUL-90
5	5	Barbu	Dumitru	dumitruB@yahoo.com	0746888320	13-MAY-00
6	6	Teodorescu	Eugen	eugen_t@yahoo.com	0770522210	23-NOV-65
7	7	Lupu	Maria	maria_lupu@yahoo.com	0757241888	03-DEC-77
8	8	Sava	Florentina	savaflo@gmail.com	0766560779	29-JUN-02
9	9	Nistor	Mariana	mariananistor@gmail.com	0745582866	03-MAY-01
10	10	Barbu	Ana Maria	anamariab@yahoo.com	0788821557	31-JAN-92
11	11	Dumitrescu	Andreea	andreea_dumi@yahoo.com	0749267946	07-JUL-79
12	12	Gheorghiu	Mihaela	mihaela.g@yahoo.com	0727774485	24-NOV-69
13	13	Pop	Rodica	rodica_pop@yahoo.com	0791567138	16-APR-99
14	14	Constantinescu	Roxana	roxana.c@yahoo.com	0743479564	12-DEC-82
15	15	Dobre	Gabriela	gabi_dobre@gmail.com	0765870667	28-FEB-97
16	16	Ciobanu	Adriana	a.ciobanu@yahoo.com	0732936930	03-NOV-60
17	17	Toma	Claudia	claudia.toma@yahoo.com	0736630215	16-SEP-95
18	18	Stanescu	Constantin	ctin_stanescu@gmail.com	0756729015	08-AUG-55
19	19	Dima	Gheorghe	dima.gh@gmail.com	0774034758	16-APR-68
20	20	Dumitrescu	Cristina	cristinaD@gmail.com	0784550324	15-MAY-00
21	21	Popescu	Elena	elenapop@yahoo.com	0781236689	01-JUN-85

CLASE:

	COD_CLASA	NUME	PRET_ADITIOAL	FACILITATI
1	10	Economica	(null)	masuta retractabila
2	20	Business	300	scaune ajustabile, o masa calda
3	30	First class	500	lounge, meniu variat, paturi
4	40	Business Premium	800	business + lounge
5	50	First Class Premium	1000	first class + meniu lux

AVIOANE_AU_CLASE:

ÎNSOȚESC:

	COD_CLASA	COD_AVION	CAPACITATE	ID_INSOSITOR_DE_ZBOR	COD_CALATORIE	SEF_CABINA
1	10	1000	100	1	10	1001 20
2	20	1000	49	2	20	1001 (null)
3	10	1001	141	3	30	1001 20
4	10	1002	132	4	10	1003 (null)
5	10	1003	52	5	30	1003 10
6	10	1004	80	6	10	1004 20
7	20	1004	61	7	20	1004 (null)
8	10	1005	126	8	40	1005 (null)
9	10	1006	100	9	40	1006 (null)
10	20	1006	40	10	50	1006 40
11	30	1006	49	11	80	1007 90
12	10	1007	160	12	90	1007 (null)
13	20	1007	26	13	90	1008 (null)
14	10	1008	100	14	120	1009 (null)
15	20	1008	100	15	120	1010 (null)
16	30	1008	20	16	60	1011 (null)
17	40	1008	10	17	70	1011 60
18	10	1009	185	18	60	1012 70
19	10	1010	162	19	70	1012 (null)
				20	20	1013 (null)
				21	30	1013 20

COMPANII:

	COD_COMPANIE	NUME
1	120	Tarom
2	121	Blue Air
3	122	Wizz Air
4	123	KLM
5	124	Ryanair
6	125	Lufthansa

AVIOANE:

	⚡ COD_AVION	⚡ MODEL	⚡ COD_COMPANIE	⚡ CAPACITATE	⚡ VITEZA_DE_CROAZIERA
1	1000	BOEING 737-700	120	149	870
2	1001	BOEING 737-300	120	141	800
3	1002	AIRBUS A318-111	120	132	850
4	1003	ATR 42-500	120	52	550
5	1004	BOEING 737-300	121	141	800
6	1005	BOEING 737-500	121	126	785
7	1006	BOEING 737-800	121	189	823
8	1007	AIRBUS A320-200	122	186	900
9	1008	AIRBUS A321-200	122	230	840
10	1009	AIRBUS A330-200	123	185	913
11	1010	BOEING 737-800	124	162	946

REZERVĂ:

	⚡ COD_CLASA	⚡ COD_CALATORIE	⚡ ID_PASAGER	⚡ COD_REZERVARE	⚡ PRET_BILET
1	30	1001	2	100001	300
2	30	1001	3	100002	300
3	10	1001	15	100003	300
4	10	1010	7	100004	200
5	10	1010	10	100005	200
6	10	1010	11	100006	200
7	20	1006	5	100007	250
8	40	1006	12	100008	250
9	40	1006	13	100009	250
10	30	1006	14	100010	250
11	20	1012	1	100011	250
12	10	1012	17	100012	250
13	10	1012	18	100013	250
14	10	1012	20	100014	250
15	10	1004	4	100015	270
16	10	1004	6	100016	270
17	10	1003	16	100017	150
18	20	1003	8	100018	150
19	20	1003	9	100019	150
20	10	1013	19	100020	150
21	10	1007	10	100021	240
22	10	1007	11	100022	240
23	10	1011	2	100023	350
24	10	1011	3	100024	350
25	10	1008	19	100025	275
26	10	1008	17	100026	275
27	10	1008	18	100027	275

Problema cu subprogram stocat și două tipuri de colecție

```
1 --- Pentru fiecare aeroport sa se afiseze numele acestuia
2 --- si salariul mediu al pilotilor ce lucreaza acolo
3
4 /
5 CREATE OR REPLACE PROCEDURE procedura1 IS
6
7 -- Vom crea un tip de date inregistrare pentru a pastra
8 -- codul aeroportului, numele acestuia si salariul mediu
9     TYPE aeroport_record IS RECORD
10     (cod AEROPORTURI.cod_iata%TYPE,
11      nume AEROPORTURI.nume%TYPE,
12      salariu_mediu PILOTI.salariu%TYPE);
13
14 -- Datele pentru toate aeroporturile vor fi salvate intr-un
15 -- tablou imbricat ce contine o coloana de tip inregistrare creat mai sus
16     TYPE tablou_imbricat IS TABLE OF aeroport_record;
17
18 -- Vom folosi un vector pentru salvarea si calcularea
19 -- fiecarui salariu mediu al pilotilor in functie de aeroport
20     TYPE vector IS VARRAY(8) OF PILOTI.salariu%TYPE;
21
22 -- Cu ajutorul unui circlu cursor vom parcurge
23 -- toate codurile de aeroporturi
24     CURSOR c IS
25         SELECT cod_iata
26         FROM AEROPORTURI;
27
28     t tablou_imbricat := tablou_imbricat();
29     r aeroport_record;
30     v vector := vector();
31 BEGIN
32
33     FOR i IN c LOOP
34
35 -- Initial vom salva codul si numele aeroporturilor
36 -- in variabila de tip inregistrare declarata mai sus
37         SELECT cod_iata, nume
38         INTO r.cod, r.nume
39         FROM AEROPORTURI
40         WHERE i.cod_iata = cod_iata;
41
42 -- Calculam salariul mediu in functie de aeroport si
43 -- il salvam intr-un element din vector
44         v.EXTEND;
45
46         SELECT AVG(salariu)
47         INTO v(v.LAST)
48         FROM PILOTI
49         WHERE cod_aeroport = i.cod_iata;
50
51 -- Copiem valoarea salariului mediu in
52 -- campul specific din variabila de tip inregistrare
53         r.salariu_mediu := v(v.LAST);
54
55 -- Adaugam variabila de tip inregistrare curenta in
56 -- tabloul imbricat pentru a pastra toate valorile
57         t.EXTEND;
58         t(t.LAST) := r;
59
60 -- Afisam rezultatele in consola de output
61         DBMS_OUTPUT.PUT_LINE ('La aeroportul ' || t(t.LAST).nume ||
```

```

62         ' salariul mediu al pilotilor este de ' || t(t.last).
        salariu_mediu);
63     END LOOP;
64
65 END procedura1;
66 /
67
68 EXECUTE procedura1;

```

The screenshot shows the SQL Developer interface with the following components:

- Worksheet:** Contains the PL/SQL procedure code.


```

800 -- Copiem valoarea salariului mediu in
801 -- campul specific din variabila de tip inregistrare
802     r.salariu_mediu := v(v.LAST);
803
804 -- Adaugam variabila de tip inregistrare curenta in
805 -- tabloul imbricat pentru a pastra toate valorile
806     t.EXTEND;
807     t(t.LAST) := r;
808
809 -- Afisam rezultatele in consola de output
810     DBMS_OUTPUT.PUT_LINE ('La aeroportul ' || t(t.LAST).nume ||
811     '
812     salariul mediu al pilotilor este de ' || t(t.last).salariu_mediu);
813     END LOOP;
814
815 END procedura1;
816 /
817
818 EXECUTE procedura1;
819

```
- DBMS Output:** Displays the output of the procedure execution.


```

La aeroportul Avram Iancu International Airport
salariul mediu al pilotilor este de 343333
La aeroportul Mihail Kogalniceanu International Airport
salariul mediu al pilotilor este de 225333
La aeroportul Craiova International Airport
salariul mediu al pilotilor este de 395000
La aeroportul Iasi International Airport
salariul mediu al pilotilor este de 400000
La aeroportul Oradea International Airport
salariul mediu al pilotilor este de
La aeroportul Henri Coanda International Airport
salariul mediu al pilotilor este de 125000
La aeroportul Sibiu International Airport
salariul mediu al pilotilor este de
La aeroportul Traian Vuia International Airport
salariul mediu al pilotilor este de 375000

```
- Status Bar:** Shows "Task completed in 0.059 seconds".
- Messages:**
 - Procedure PROCEDURAL1 compiled
 - PL/SQL procedure successfully completed.

Problema cu subprogram stocat și un cursor

```
1  --- Sa se mareasca cu 10% salariul insotitorilor de zbor
2  --- care au fost angajati incepand cu anul 2018
3
4  /
5  CREATE OR REPLACE PROCEDURE procedura2 IS
6
7  -- Vom folosi un cursor SELECT FOR UPDATE
8  -- pentru a bloca liniile inainte ca acestea
9  -- sa fie actualizate cu noul salariu
10     CURSOR c IS
11         SELECT * FROM INSOTITORI
12         WHERE TO_CHAR(data_angajarii, 'YYYY') >= 2018
13         FOR UPDATE OF salariu NOWAIT;
14
15 -- Prin folosirea clauzei NOWAIT va fi returnata
16 -- o eroare in cazul in care liniile selectate
17 -- sunt deja blocate de catre alta comanda
18
19 BEGIN
20
21 -- Se foloseste un ciclu cursor pentru a
22 -- actualiza toate valorile selectate
23     FOR i IN c LOOP
24         UPDATE INSOTITORI
25             SET salariu = salariu + ((salariu * 10) / 100)
26             WHERE CURRENT OF c;
27     END LOOP;
28
29 END procedura2;
30 /
31
32 SELECT * FROM INSOTITORI;
33
34 EXECUTE procedura2;
35
36 SELECT * FROM INSOTITORI;
37
38 ROLLBACK;
```


PROJECT.SGRO.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

```

843 -- Se foloseste un ciclu cursor pentru a
844 -- actualiza toate valorile selectate
845 FOR i IN c LOOP
846     UPDATE INSOTITORI
847     SET salariu = salariu + ((salariu * 10) / 100)
848     WHERE CURRENT OF c;
849 END LOOP;
850
851 END procedura2;
852 /
853
854 SELECT * FROM INSOTITORI;
855
856 EXECUTE procedura2;
857
858 SELECT * FROM INSOTITORI;
859

```

Script Output x Query Result x

SQL All Rows Fetched: 13 in 0.002 seconds

ID_INSOTITOR_DE_ZBOR	COD_AEROPORT	NUME	PRENUME	TELEFON	DATA_ANGAJARII	SALARIU
1	10 OTP	Voiculescu	Alina	0745678146	04-OCT-18	33000
2	20 OTP	Moiescu	Denisa	0792218597	13-MAR-19	40000
3	30 OTP	Ionescu	Florin	0726688610	02-APR-12	90000
4	40 CLJ	Georgescu	Calin	0768216950	01-JUL-16	80000
5	50 CLJ	Antonie	Mihaela	0783717731	06-DEC-20	15000
6	60 TSR	Voica	Laura	0766138430	31-JAN-21	13000
7	70 TSR	Enache	Maria	0778767173	25-OCT-15	40000
8	80 IAS	Matache	Irina	0762706994	20-MAR-15	45000
9	90 IAS	Mirea	Mihai	0785072934	07-JUL-19	27000
10	100 CND	Lazar	Nicoleta	0718538567	09-NOV-18	33000
11	110 CND	Florea	Stefania	0784725925	23-SEP-09	80000
12	120 CRA	Miulescu	Ana	0758812546	06-JUN-20	12000
13	130 SBZ	Diaconescu	Daniela	0734456845	27-APR-18	16000

PROJECT.SGRO.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

```

843 -- Se foloseste un ciclu cursor pentru a
844 -- actualiza toate valorile selectate
845 FOR i IN c LOOP
846     UPDATE INSOTITORI
847     SET salariu = salariu + ((salariu * 10) / 100)
848     WHERE CURRENT OF c;
849 END LOOP;
850
851 END procedura2;
852 /
853
854 SELECT * FROM INSOTITORI;
855
856 EXECUTE procedura2;
857
858 SELECT * FROM INSOTITORI;
859

```

Script Output x Query Result x

SQL All Rows Fetched: 13 in 0.002 seconds

ID_INSOTITOR_DE_ZBOR	COD_AEROPORT	NUME	PRENUME	TELEFON	DATA_ANGAJARII	SALARIU
1	10 OTP	Voiculescu	Alina	0745678146	04-OCT-18	36300
2	20 OTP	Moiescu	Denisa	0792218597	13-MAR-19	44000
3	30 OTP	Ionescu	Florin	0726688610	02-APR-12	90000
4	40 CLJ	Georgescu	Calin	0768216950	01-JUL-16	80000
5	50 CLJ	Antonie	Mihaela	0783717731	06-DEC-20	16500
6	60 TSR	Voica	Laura	0766138430	31-JAN-21	14300
7	70 TSR	Enache	Maria	0778767173	25-OCT-15	40000
8	80 IAS	Matache	Irina	0762706994	20-MAR-15	45000
9	90 IAS	Mirea	Mihai	0785072934	07-JUL-19	29700
10	100 CND	Lazar	Nicoleta	0718538567	09-NOV-18	36300
11	110 CND	Florea	Stefania	0784725925	23-SEP-09	80000
12	120 CRA	Miulescu	Ana	0758812546	06-JUN-20	13200
13	130 SBZ	Diaconescu	Daniela	0734456845	27-APR-18	17600

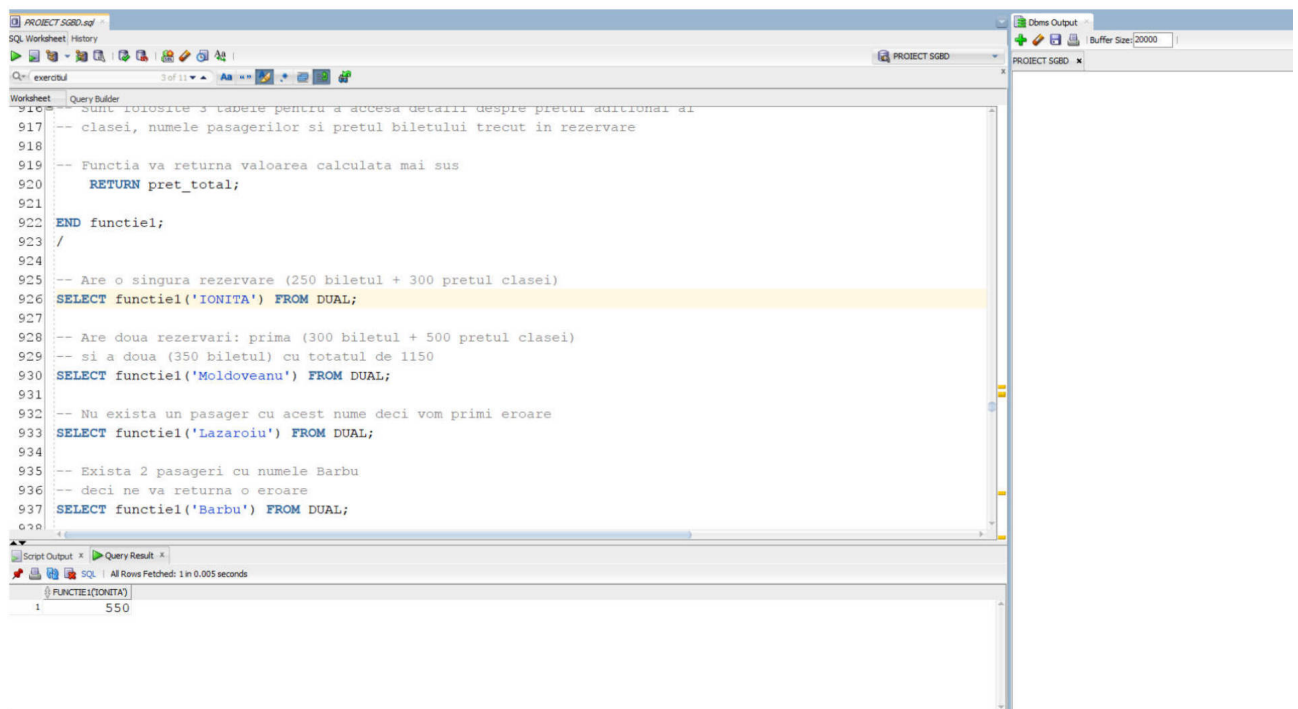
Problema cu funcție și 3 tabele într-o comandă

```
1  -- Citindu-se numele unui pasager de la tastatura, sa se returneze
2  -- pretul total al rezervarii facute de acesta (pretul biletului +
3  -- pretul additional pentru clasa, daca este cazul)
4
5  -- In cazul in care un pasager are mai multe rezervari,
6  -- se va returna suma tuturor preturilor totale
7
8  -- Se vor arunca erori in cazul in care: exista mai multi pasageri
9  -- cu numele dat, nu exista un pasager cu numele dat sau nu
10 -- exista rezervari facute de catre acest pasager
11
12 /
13 CREATE OR REPLACE FUNCTION functie1(v_nume PASAGERI.nume%TYPE)
14 RETURN REZERVA.pret_bilet%TYPE IS
15
16     pret_total REZERVA.pret_bilet%TYPE;
17     contor NUMBER;
18 BEGIN
19
20 -- Vom trata exceptiile la inceputul programului
21
22 -- Numaram cati pasageri exista cu numele dat ca parametru
23     SELECT COUNT(*) INTO contor
24     FROM PASAGERI WHERE INITCAP(v_nume) = INITCAP(nume);
25
26 -- Tratam cazul in care nu exista niciun pasager si
27 -- cazul in care exista mai multe pasageri cu numele dat
28     IF contor = 0
29     THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista un pasager cu acest
30 nume');
31     ELSIF contor > 1
32     THEN RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi pasageri cu
33 acest nume');
34     END IF;
35
36 -- Folosim aceiasi variabila pentru a numara cate rezervari exista
37 -- pentru pasagerul dat iar daca nu exista niciuna vom arunca o eroare
38     SELECT COUNT(*) INTO contor
39     FROM PASAGERI RIGHT OUTER JOIN REZERVA USING (id_pasager)
40     WHERE INITCAP(v_nume) = INITCAP(nume);
41
42     IF contor = 0
43     THEN RAISE_APPLICATION_ERROR(-20002, 'Nu exista rezervari pe acest
44 nume');
45     END IF;
46
47 -- Functia de SUM va trata atat cazul in care exista o singura rezervare
48 -- dar si cazul in care exista mai multe iar functia NVL va inlocui
49 -- valoarea NULL a clasei cu codul 10 cu valoarea 0 pentru a o adauga la
50 -- suma
51     SELECT SUM(pret_bilet + NVL(pret_additional, 0))
52     INTO pret_total
53     FROM REZERVA JOIN CLASE USING (cod_clasa) JOIN PASAGERI USING (
54 id_pasager)
55     WHERE INITCAP(v_nume) = INITCAP(PASAGERI.nume);
56
57 -- Sunt folosite 3 tabele pentru a accesa detalii despre pretul additional
58 -- al
59 -- clasei, numele pasagerilor si pretul biletului trecut in rezervare
```

```

55 -- Functia va returna valoarea calculata mai sus
56     RETURN pret_total;
57
58 END functie1;
59 /
60
61 -- Are o singura rezervare (250 biletul + 300 pretul clasei)
62 SELECT functie1('IONITA') FROM DUAL;
63
64 -- Are doua rezervari: prima (300 biletul + 500 pretul clasei)
65 -- si a doua (350 biletul) cu totalul de 1150
66 SELECT functie1('Moldoveanu') FROM DUAL;
67
68 -- Nu exista un pasager cu acest nume deci vom primi eroare
69 SELECT functie1('Lazaroiu') FROM DUAL;
70
71 -- Exista 2 pasageri cu numele Barbu
72 -- deci ne va returna o eroare
73 SELECT functie1('Barbu') FROM DUAL;
74
75 -- Pasagerul exista insa nu are nicio rezervare
76 -- deci va ridica o eroare
77 SELECT functie1('popescu') FROM DUAL;

```



PROJECT.SGBO.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

Worksheet Query Builder

```

-- Sunt necesare 3 tabele pentru a accesa detalii despre pretul aditional al
-- clasei, numele pasagerilor si pretul biletului trecut in rezervare
-- Functia va returna valoarea calculata mai sus
RETURN pret_total;
END functiei;
/
-- Are o singura rezervare (250 biletul + 300 pretul clasei)
SELECT functiei('IONITA') FROM DUAL;
-- Are doua rezervari: prima (300 biletul + 500 pretul clasei)
-- si a doua (350 biletul) cu totalul de 1150
SELECT functiei('Moldoveanu') FROM DUAL;
-- Nu exista un pasager cu acest nume deci vom primi eroare
SELECT functiei('Lazaroiu') FROM DUAL;
-- Exista 2 pasageri cu numele Barbu
-- deci ne va returna o eroare
SELECT functiei('Barbu') FROM DUAL;

```

Script Output x Query Result x

SQL All Rows Fetched: 1 in 0.003 seconds

1	functiei(MOLDOVEANU)
1	1150

DBMS Output PROJECT.SGBO Buffer Size: 20000

PROJECT.SGBO.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

Worksheet Query Builder

```

-- Sunt necesare 3 tabele pentru a accesa detalii despre pretul aditional al
-- clasei, numele pasagerilor si pretul biletului trecut in rezervare
-- Functia va returna valoarea calculata mai sus
RETURN pret_total;
END functiei;
/
-- Are o singura rezervare (250 biletul + 300 pretul clasei)
SELECT functiei('IONITA') FROM DUAL;
-- Are doua rezervari: prima (300 biletul + 500 pretul clasei)
-- si a doua (350 biletul) cu totalul de 1150
SELECT functiei('Moldoveanu') FROM DUAL;
-- Nu exista un pasager cu acest nume deci vom primi eroare
SELECT functiei('Lazaroiu') FROM DUAL;
-- Exista 2 pasageri cu numele Barbu
-- deci ne va returna o eroare
SELECT functiei('Barbu') FROM DUAL;

```

Script Output x Query Result x

SQL Executing: SELECT functiei('Lazaroiu') FROM DUAL in 0 seconds

ORA-20000: Nu exista un pasager cu acest nume
ORA-06512: at 'LAZAROIU/TEODORA.FUNCTIE1', line 17
20000.00000 - "5a"
Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.

DBMS Output PROJECT.SGBO Buffer Size: 20000

PROJECT_SGBO.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

Worksheet Query Builder

```
920 RETURN pret_total;  
921  
922 END functiei;  
923 /  
924  
925 -- Are o singura rezervare (250 biletul + 300 pretul clasei)  
926 SELECT functiei('IONITA') FROM DUAL;  
927  
928 -- Are doua rezervari: prima (300 biletul + 500 pretul clasei)  
929 -- si a doua (350 biletul) cu totalul de 1150  
930 SELECT functiei('Moldoveanu') FROM DUAL;  
931  
932 -- Nu exista un pasager cu acest nume deci vom primi eroare  
933 SELECT functiei('Lazaroiu') FROM DUAL;  
934  
935 -- Exista 2 pasageri cu numele Barbu  
936 -- deci ne va returna o eroare  
937 SELECT functiei('Barbu') FROM DUAL;  
938  
939 -- Pasagerul exista insa nu are nicio rezervare  
940 -- deci va ridica o eroare  
941 SELECT functiei('popescu') FROM DUAL;  
942
```

Script Output x Query Result x

Executing: SELECT functiei('Barbu') FROM DUAL in 0 seconds

ORA-20001: Exista mai multi pasageri cu acest nume
ORA-06512: at 'LAZAROIU/TEODORA.FUNCTIE1', line 19

PROJECT_SGBO.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

Worksheet Query Builder

```
922 END functiei;  
923 /  
924  
925 -- Are o singura rezervare (250 biletul + 300 pretul clasei)  
926 SELECT functiei('IONITA') FROM DUAL;  
927  
928 -- Are doua rezervari: prima (300 biletul + 500 pretul clasei)  
929 -- si a doua (350 biletul) cu totalul de 1150  
930 SELECT functiei('Moldoveanu') FROM DUAL;  
931  
932 -- Nu exista un pasager cu acest nume deci vom primi eroare  
933 SELECT functiei('Lazaroiu') FROM DUAL;  
934  
935 -- Exista 2 pasageri cu numele Barbu  
936 -- deci ne va returna o eroare  
937 SELECT functiei('Barbu') FROM DUAL;  
938  
939 -- Pasagerul exista insa nu are nicio rezervare  
940 -- deci va ridica o eroare  
941 SELECT functiei('popescu') FROM DUAL;  
942  
943
```

Script Output x Query Result x

Executing: SELECT functiei('popescu') FROM DUAL in 0 seconds

ORA-20002: Nu exista rezervari pe acest nume
ORA-06512: at 'LAZAROIU/TEODORA.FUNCTIE1', line 29

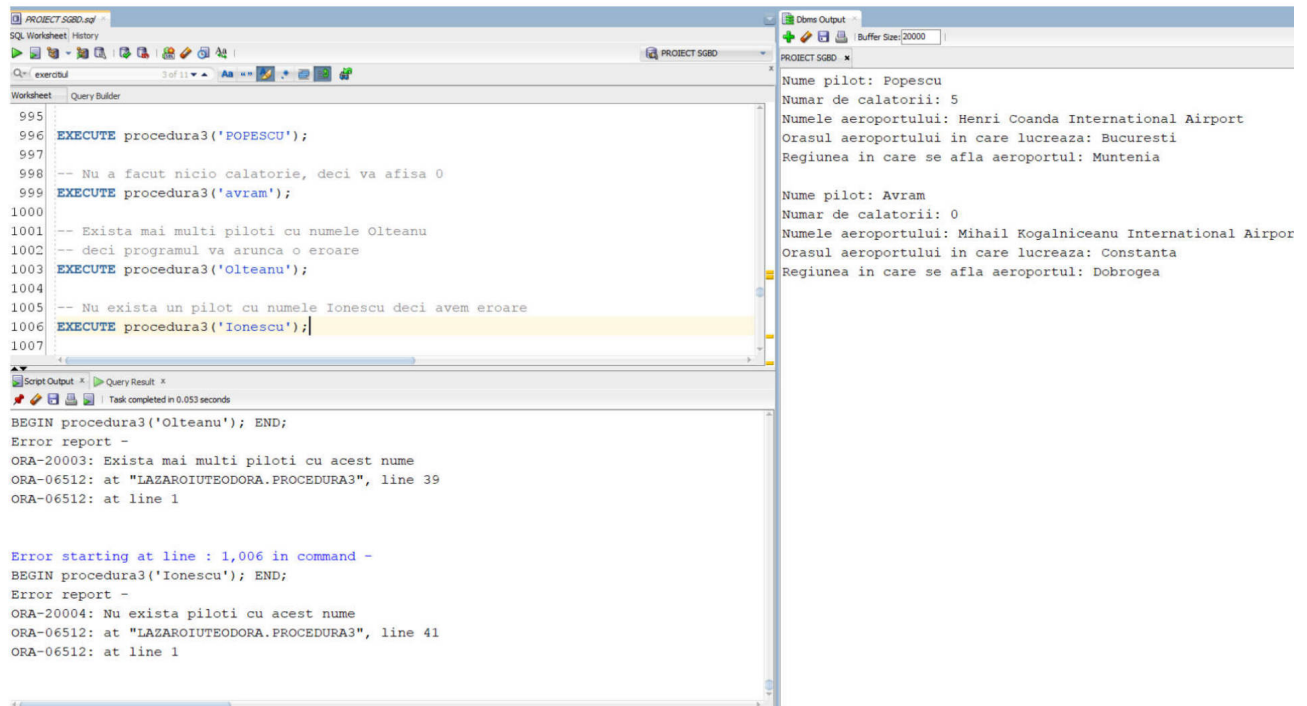
Problema cu procedură și 5 tabele într-o comandă

```
1  --- Se da numele unui pilot ca parametru. Sa se afiseze numarul de
    calatorii
2  --- la care acesta a participat, numele aeroportului la care lucreaza
3  --- si numele orasului si al regiunii in care se afla aeroportul sau
4
5  /
6  CREATE OR REPLACE PROCEDURE procedura3 (nume_pilot PILOTI.nume%TYPE) IS
7
8  -- Vom declara variabilele in care vom salva
9  -- datele care trebuiesc selectate si afisate
10     numar_calatorii NUMBER(4);
11     nume_aeroport AEROPORTURI.nume%TYPE;
12     nume_oras ORASE.nume%TYPE;
13     nume_regiune REGIUNI.nume%TYPE;
14
15 BEGIN
16
17 -- Vom folosi un RIGHT JOIN intre CALATORII si PILOTI pentru
18 -- a include si pilotii care nu apar in tabelul de calatorii
19 -- (adica nu au facut nicio calatorie) dar apartin unui aeroport.
20 -- Conditia de JOIN intre acestea verifica daca pilotul a fost
21 -- fie pilot principal, fie secundar si contorizeaza ambele cazuri
22
23 -- In continuare se face JOIN simplu cu tabelele AEROPORTURI, ORASE
24 -- si REGIUNI pentru a putea accesa si selecta numele acestora
25
26     SELECT COUNT(cod_calatorie), a.nume, o.nume, r.nume
27     INTO numar_calatorii, nume_aeroport, nume_oras, nume_regiune
28     FROM CALATORII c RIGHT JOIN PILOTI p ON (id_pilot =
29     id_pilot_principal OR
30     id_pilot = id_pilot_secundar) JOIN AEROPORTURI a ON (p.cod_aeroport =
31     cod_iata)
32     JOIN ORASE o ON (o.cod_aeroport = cod_iata) JOIN REGIUNI r USING (
33     cod_regiune)
34     WHERE INITCAP(nume_pilot) = INITCAP(p.nume)
35     GROUP BY a.nume, o.nume, r.nume;
36
37 -- Afisam in consola datele selectate mai sus
38     DBMS_OUTPUT.PUT_LINE('Nume pilot: ' || INITCAP(nume_pilot));
39     DBMS_OUTPUT.PUT_LINE('Numar de calatorii: ' || numar_calatorii);
40     DBMS_OUTPUT.PUT_LINE('Numele aeroportului: ' || nume_aeroport);
41     DBMS_OUTPUT.PUT_LINE('Orasul aeroportului in care lucreaza: ' ||
42     nume_oras);
43     DBMS_OUTPUT.PUT_LINE('Regiunea in care se afla aeroportul: ' ||
44     nume_regiune);
45
46 EXCEPTION
47
48     WHEN TOO_MANY_ROWS
49     THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi piloti cu
50     acest nume');
51     WHEN NO_DATA_FOUND
52     THEN RAISE_APPLICATION_ERROR(-20004, 'Nu exista piloti cu acest nume'
53     );
54
55 END procedura3;
56
57 /
58
59 EXECUTE procedura3('POPESCU');
```

```

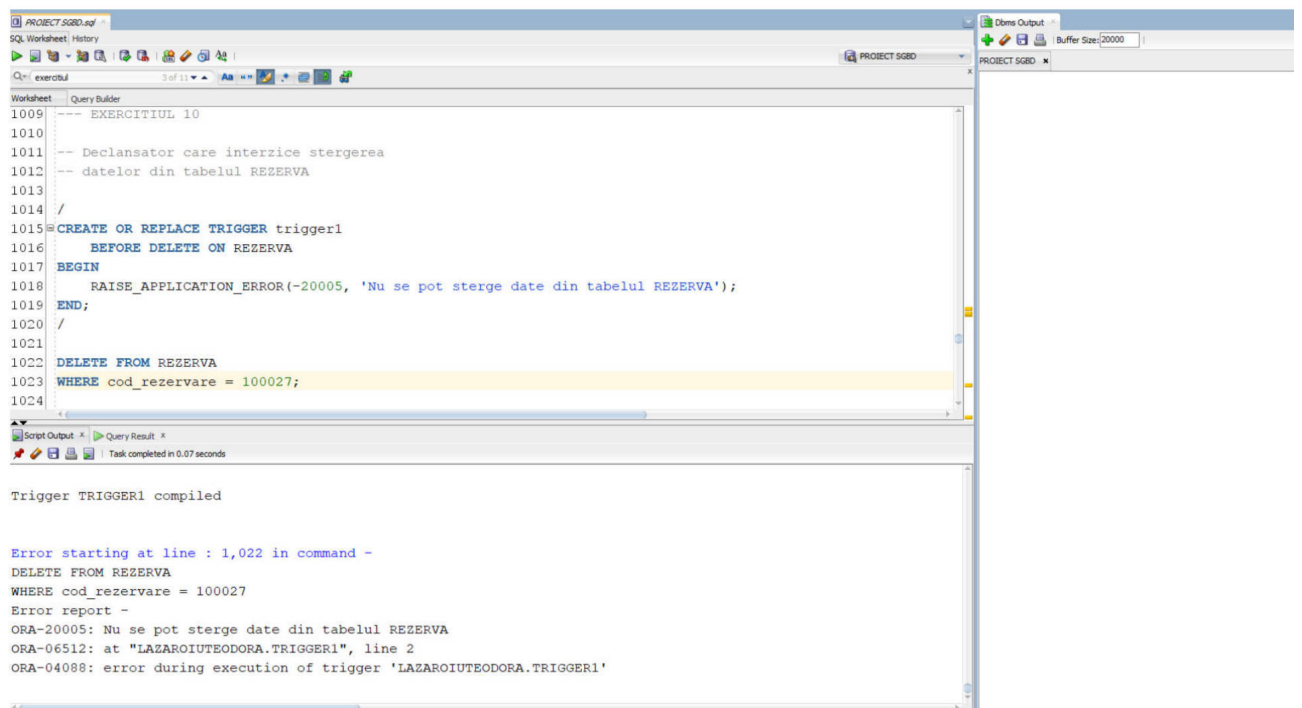
54 EXECUTE procedura3('avram');
55
56 -- Exista mai multi piloti cu numele Olteanu
57 -- deci programul va arunca o eroare
58 EXECUTE procedura3('Olteanu');
59
60 -- Nu exista un pilot cu numele Ionescu deci avem eroare
61 EXECUTE procedura3('Ionescu');

```



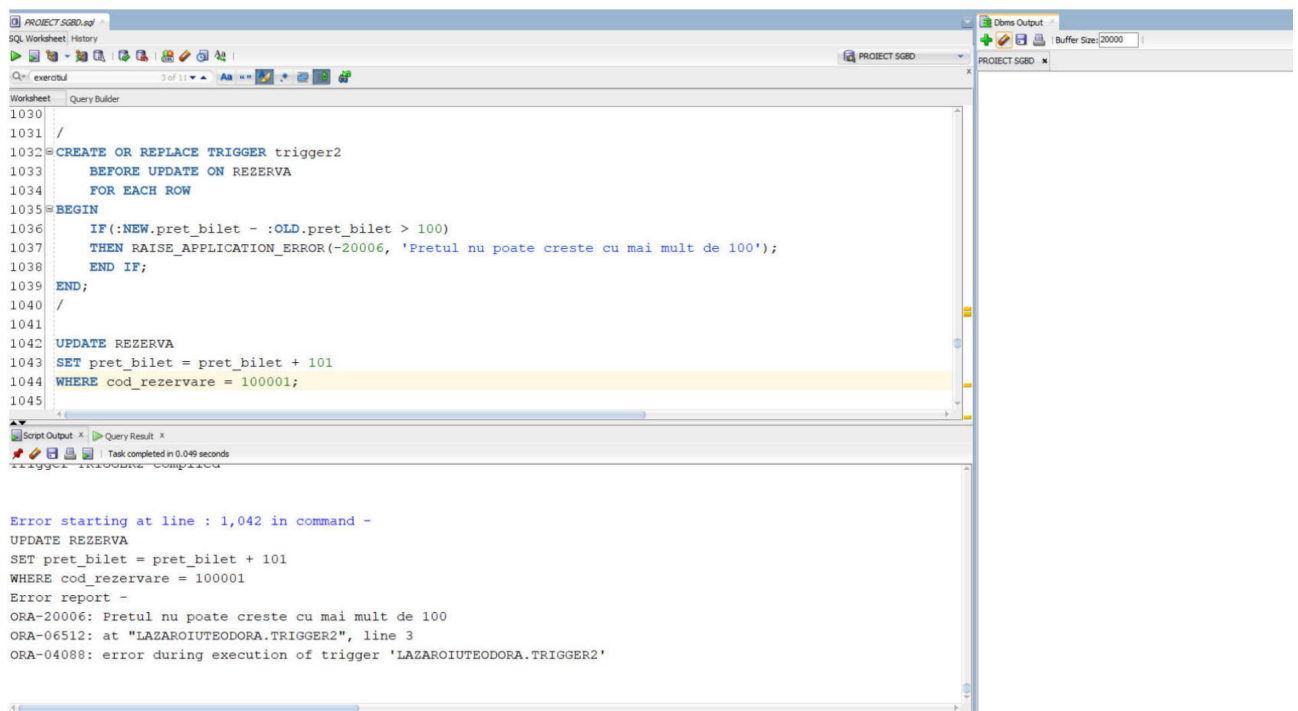
Trigger de tip LMD la nivel de comandă

```
1  -- Declansator care interzice stergerea
2  -- datelor din tabelul REZERVA
3
4  /
5  CREATE OR REPLACE TRIGGER trigger1
6    BEFORE DELETE ON REZERVA
7  BEGIN
8    RAISE_APPLICATION_ERROR(-20005, 'Nu se pot sterge date din tabelul
9    REZERVA');
10 END;
11 /
12 DELETE FROM REZERVA
13 WHERE cod_rezervare = 100027;
```



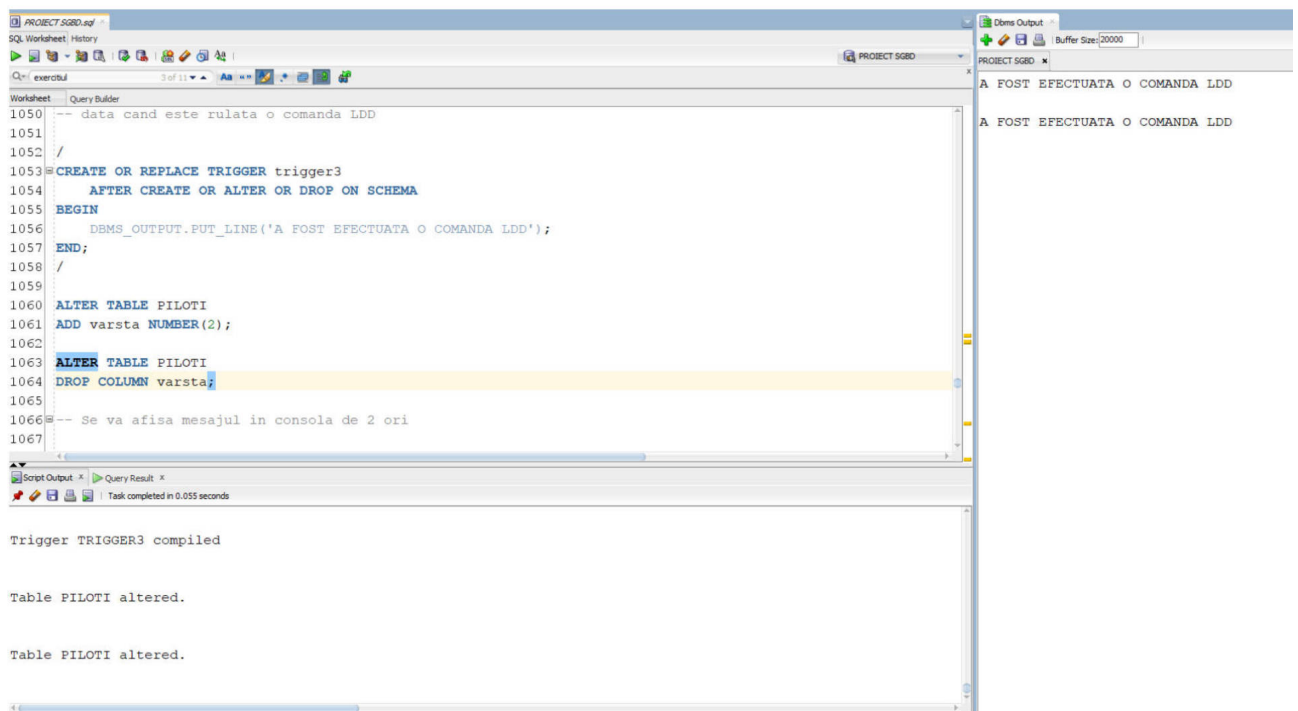
Trigger de tip LMD la nivel de linie

```
1  -- Declansator care interzice cresterea
2  -- pretului unui bilet cu mai mult de 100
3
4  /
5  CREATE OR REPLACE TRIGGER trigger2
6      BEFORE UPDATE ON REZERVA
7      FOR EACH ROW
8  BEGIN
9      IF (:NEW.pret_bilet - :OLD.pret_bilet > 100)
10     THEN RAISE_APPLICATION_ERROR(-20006, 'Pretul nu poate creste cu mai
11     mult de 100');
12     END IF;
13 END;
14 /
15 UPDATE REZERVA
16 SET pret_bilet = pret_bilet + 101
17 WHERE cod_rezervare = 100001;
```



Trigger de tip LDD

```
1  -- Declansator care afiseaza un mesaj de fiecare
2  -- data cand este rulata o comanda LDD
3
4  /
5  CREATE OR REPLACE TRIGGER trigger3
6      AFTER CREATE OR ALTER OR DROP ON SCHEMA
7  BEGIN
8      DBMS_OUTPUT.PUT_LINE('A FOST EFECTUATA O COMANDA LDD');
9  END;
10 /
11
12 ALTER TABLE PILOTI
13 ADD varsta NUMBER(2);
14
15 ALTER TABLE PILOTI
16 DROP COLUMN varsta;
17
18 -- Se va afisa mesajul in consola de 2 ori
```



Pachet cu obiectele din proiect

```
1 /
2 CREATE OR REPLACE PACKAGE pachet1 AS
3     FUNCTION functie1(v_nume PASAGERI.nume%TYPE) RETURN REZERVA.
4     pret_bilet%TYPE;
5     PROCEDURE procedura1;
6     PROCEDURE procedura2;
7     PROCEDURE procedura3(nume_pilot PILOTI.nume%TYPE);
8 END pachet1;
9 /
10 CREATE OR REPLACE PACKAGE BODY pachet1 AS
11 -- Pretul total pentru rezervarile facute de un pasager
12 FUNCTION functie1(v_nume PASAGERI.nume%TYPE)
13 RETURN REZERVA.pret_bilet%TYPE IS
14     pret_total REZERVA.pret_bilet%TYPE;
15     contor NUMBER;
16 BEGIN
17     SELECT COUNT(*) INTO contor
18     FROM PASAGERI WHERE INITCAP(v_nume) = INITCAP(nume);
19
20     IF contor = 0
21     THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista un pasager cu
22     acest nume');
23     ELSIF contor > 1
24     THEN RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi pasageri
25     cu acest nume');
26     END IF;
27
28     SELECT COUNT(*) INTO contor
29     FROM PASAGERI RIGHT OUTER JOIN REZERVA USING (id_pasager)
30     WHERE INITCAP(v_nume) = INITCAP(nume);
31
32     IF contor = 0
33     THEN RAISE_APPLICATION_ERROR(-20002, 'Nu exista rezervari pe
34     acest nume');
35     END IF;
36
37     SELECT SUM(pret_bilet + NVL(pret_aditional, 0))
38     INTO pret_total
39     FROM REZERVA JOIN CLASE USING (cod_clasa) JOIN PASAGERI USING (
40     id_pasager)
41     WHERE INITCAP(v_nume) = INITCAP(PASAGERI.nume);
42
43     RETURN pret_total;
44 END functie1;
45
46 -- Salariul mediu al pilotilor pentru fiecare aeroport
47 PROCEDURE procedura1 IS
48     TYPE aeroport_record IS RECORD
49     (cod AEROPORTURI.cod_iata%TYPE,
50     nume AEROPORTURI.nume%TYPE,
51     salariu_mediu PILOTI.salariu%TYPE);
52
53     TYPE tablou_imbricat IS TABLE OF aeroport_record;
54     TYPE vector IS VARRAY(8) OF PILOTI.salariu%TYPE;
55     CURSOR c IS
56         SELECT cod_iata
57         FROM AEROPORTURI;
```

```

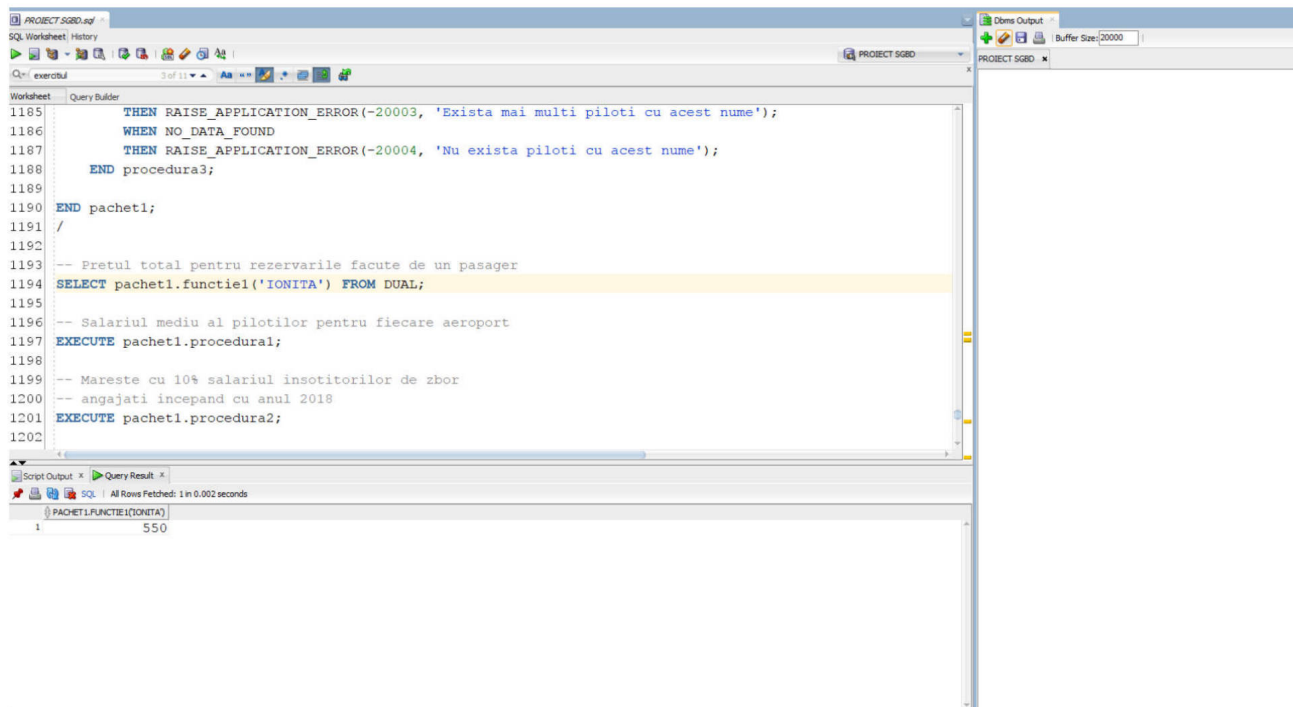
57     v vector := vector();
58 BEGIN
59     FOR i IN c LOOP
60         SELECT cod_iata, nume
61         INTO r.cod, r.nume
62         FROM AEROPORTURI
63         WHERE i.cod_iata = cod_iata;
64
65         v.EXTEND;
66
67         SELECT AVG(salariu)
68         INTO v(v.LAST)
69         FROM PILOTI
70         WHERE cod_aeroport = i.cod_iata;
71
72         r.salariu_mediu := v(v.LAST);
73         t.EXTEND;
74         t(t.LAST) := r;
75     END LOOP;
76 END procedura1;
77
78 -- Mareste cu 10% salariul insotitorilor de zbor
79 -- angajati incepand cu anul 2018
80 PROCEDURE procedura2 IS
81     CURSOR c IS
82     SELECT * FROM INSOTITORI
83     WHERE TO_CHAR(data_angajarii, 'YYYY') >= 2018
84     FOR UPDATE OF salariu NOWAIT;
85 BEGIN
86     FOR i IN c LOOP
87         UPDATE INSOTITORI
88         SET salariu = salariu + (salariu * 10)/100
89         WHERE CURRENT OF c;
90     END LOOP;
91 END procedura2;
92
93 -- Detalii despre un pilot dat ca parametru
94 PROCEDURE procedura3 (nume_pilot PILOTI.nume%TYPE) IS
95     numar_calatorii NUMBER(4);
96     nume_aeroport AEROPORTURI.nume%TYPE;
97     nume_oras ORASE.nume%TYPE;
98     nume_regiune REGIUNI.nume%TYPE;
99 BEGIN
100     SELECT COUNT(cod_calatorie), a.nume, o.nume, r.nume
101     INTO numar_calatorii, nume_aeroport, nume_oras, nume_regiune
102     FROM CALATORII c RIGHT JOIN PILOTI p ON (id_pilot =
103     id_pilot_principal OR
104     id_pilot = id_pilot_secundar) JOIN AEROPORTURI a ON (p.
105     cod_aeroport = cod_iata)
106     JOIN ORASE o ON (o.cod_aeroport = cod_iata) JOIN REGIUNI r USING
107     (cod_regiune)
108     WHERE INITCAP(nume_pilot) = INITCAP(p.nume)
109     GROUP BY a.nume, o.nume, r.nume;
110
111     DBMS_OUTPUT.PUT_LINE('Nume pilot: ' || INITCAP(nume_pilot));
112     DBMS_OUTPUT.PUT_LINE('Numar de calatorii: ' || numar_calatorii);
113     DBMS_OUTPUT.PUT_LINE('Numele aeroportului: ' || nume_aeroport);
114     DBMS_OUTPUT.PUT_LINE('Orasul aeroportului in care lucreaza: ' ||
nume_oras);
115     DBMS_OUTPUT.PUT_LINE('Regiunea in care se afla aeroportul: ' ||
nume_regiune);
116 EXCEPTION
117 WHEN TOO_MANY_ROWS

```

```

115         THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi piloti cu
         acest nume');
116         WHEN NO_DATA_FOUND
117         THEN RAISE_APPLICATION_ERROR(-20004, 'Nu exista piloti cu acest
         nume');
118         END procedura3;
119
120 END pachet1;
121 /
122
123 -- Pretul total pentru rezervarile facute de un pasager
124 SELECT pachet1.functie1('IONITA') FROM DUAL;
125
126 -- Salariul mediu al pilotilor pentru fiecare aeroport
127 EXECUTE pachet1.procedura1;
128
129 -- Mareste cu 10% salariul insotitorilor de zbor
130 -- angajati incepand cu anul 2018
131 EXECUTE pachet1.procedura2;
132
133 -- Detalii despre un pilot
134 EXECUTE pachet1.procedura3('POPESCU');

```



PROJECT.SQ20.sql

SQL Worksheet: History

Q= PROCEDURA1

Worksheet: Query Builder

```

1188 THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi piloti cu acest nume');
1189 WHEN NO_DATA_FOUND
1190 THEN RAISE_APPLICATION_ERROR(-20004, 'Nu exista piloti cu acest nume');
1191 END procedura3;
1192
1193 END pachet1;
1194 /
1195
1196 -- Pretul total pentru rezervarile facute de un pasager
1197 SELECT pachet1.functiei('IONITA') FROM DUAL;
1198
1199 -- Salariul mediu al pilotilor pentru fiecare aeroport
1200 EXECUTE pachet1.procedural;
1201
1202 -- Mareste cu 10% salariul insotitorilor de zbor
1203 -- angajati incepand cu anul 2018
1204 EXECUTE pachet1.procedura2;
1205
1206 -- Detalii despre un pilot
1207 EXECUTE pachet1.procedura3('POPESCU');
1208

```

Script Output: Query Result

Task completed in 0.056 seconds

Package Body PACHET1 compiled

PL/SQL procedure successfully completed.

Dbms Output: PROJECT.SQ20

La aeroportul Avram Iancu International Airport
salariul mediu al pilotilor este de 343333
La aeroportul Mihail Kogalniceanu International Airport
salariul mediu al pilotilor este de 225333
La aeroportul Craiova International Airport
salariul mediu al pilotilor este de 395000
La aeroportul Iasi International Airport
salariul mediu al pilotilor este de 400000
La aeroportul Oradea International Airport
salariul mediu al pilotilor este de
La aeroportul Henri Coanda International Airport
salariul mediu al pilotilor este de 125000
La aeroportul Sibiu International Airport
salariul mediu al pilotilor este de
La aeroportul Traian Vuia International Airport
salariul mediu al pilotilor este de 375000

PROJECT.SQ20.sql

SQL Worksheet: History

Q= PROCEDURA1

Worksheet: Query Builder

```

1189 WHEN NO_DATA_FOUND
1190 THEN RAISE_APPLICATION_ERROR(-20004, 'Nu exista piloti cu acest nume');
1191 END procedura3;
1192
1193 END pachet1;
1194 /
1195
1196 -- Pretul total pentru rezervarile facute de un pasager
1197 SELECT pachet1.functiei('IONITA') FROM DUAL;
1198
1199 -- Salariul mediu al pilotilor pentru fiecare aeroport
1200 EXECUTE pachet1.procedural;
1201
1202 -- Mareste cu 10% salariul insotitorilor de zbor
1203 -- angajati incepand cu anul 2018
1204 EXECUTE pachet1.procedura2;
1205
1206 SELECT * FROM INSOTITORI;
1207
1208 ROLLBACK;
1209

```

Script Output: Query Result

All Rows Fetched: 13 in 0.002 seconds

ID_INSOTITOR_DE_ZBOR	COD_AEROPORT	NUME	PRENUME	TELEFON	DATA_ANGAJARII	SALARIU
1	10 OTP	Voiculescu	Alina	0745678146	04-OCT-18	36300
2	20 OTP	Moisescu	Denisa	0792218597	13-MAR-19	44000
3	30 OTP	Ionescu	Florin	0726688610	02-APR-12	90000
4	40 CLJ	Georgescu	Calin	0768216950	01-JUL-16	80000
5	50 CLJ	Antonie	Mihaela	0783717731	06-DEC-20	16500
6	60 TSR	Voica	Laura	0766138430	31-JAN-21	14300
7	70 TSR	Enache	Maria	0778767173	25-OCT-15	40000
8	80 IAS	Matache	Irina	0762706994	20-MAR-15	45000
9	90 IAS	Mirea	Mihai	0785072934	07-JUL-19	29700

The screenshot displays the Oracle SQL Developer environment. The main window shows a PL/SQL script in the 'Query Builder' tab. The script includes comments in Romanian and SQL/PL/SQL code. The script ends with a rollback and a message indicating successful completion. The 'Script Output' window on the right shows the results of the script execution, including the pilot's name, the number of passengers, the airport name, the city, and the region.

```
1192
1193 END pachet1;
1194 /
1195
1196 -- Pretul total pentru rezervarile facute de un pasager
1197 SELECT pachet1.functie1('IONITA') FROM DUAL;
1198
1199 -- Salariul mediu al pilotilor pentru fiecare aeroport
1200 EXECUTE pachet1.procedura1;
1201
1202 -- Mareste cu 10% salariul insotitorilor de zbor
1203 -- angajati incepand cu anul 2018
1204 EXECUTE pachet1.procedura2;
1205
1206 SELECT * FROM INSOTITORI;
1207
1208 ROLLBACK;
1209
1210 -- Detalii despre un pilot
1211 EXECUTE pachet1.procedura3('POPESCU');
1212
```

Script Output:

```
Nume pilot: Popescu
Numar de calatorii: 5
Numele aeroportului: Henri Coanda International Airport
Orasul aeroportului in care lucreaza: Bucuresti
Regiunea in care se afla aeroportul: Muntenia
```

Rollback complete.

PL/SQL procedure successfully completed.

Pachet cu flux de acțiuni integrate

```
1  -- Se creeaza un pachet ce contine un tabel imbricat
2  -- cu o coloana de tip inregistrare ce vor pastra date
3  -- despre angajatii aeroporturilor (piloti si insotitori)
4
5  -- Vor fi apelate 2 proceduri pentru a se introduce
6  -- datele in tabelul imbricat si se vor folosi 2 functii
7  -- pentru a calcula salariul mediul al angajatilor si
8  -- numarul de angajati care au salariul peste medie.
9  -- A doua functie se va folosi de prima.
10
11 /
12 CREATE OR REPLACE PACKAGE pachet2 AS
13     TYPE angajati_record IS RECORD
14     (nume PILOTI.nume%TYPE,
15      prenume PILOTI.prenume%TYPE,
16      cod_aeroport PILOTI.cod_aeroport%TYPE,
17      salariu PILOTI.salariu%TYPE,
18      titlu_job VARCHAR2(20));
19
20     TYPE tablou_angajati IS TABLE OF angajati_record;
21
22     t tablou_angajati := tablou_angajati();
23
24     -- introduce date in tablou din tabelul PILOTI
25     PROCEDURE procedura1_pachet;
26     -- introduce date in tablou din tabelul INSOTITORI
27     PROCEDURE procedura2_pachet;
28     -- returneaza media salariilor angajatilor
29     FUNCTION functie1_pachet RETURN NUMBER;
30     -- returneaza numarul de angajati
31     -- care au salariul peste medie
32     FUNCTION functie2_pachet RETURN NUMBER;
33 END pachet2;
34 /
35
36 CREATE OR REPLACE PACKAGE BODY pachet2 AS
37
38     -- preluarea datelor din tabelul PILOTI
39     -- si introducerea lor in tablou
40     PROCEDURE procedura1_pachet IS
41         r angajati_record;
42         CURSOR c IS
43             SELECT id_pilot FROM PILOTI;
44     BEGIN
45         FOR i IN c LOOP
46             SELECT nume, prenume, cod_aeroport, salariu, 'pilot'
47             INTO r.nume, r.prenume, r.cod_aeroport, r.salariu, r.
48             titlu_job
49             FROM PILOTI WHERE id_pilot = i.id_pilot;
50
51             t.EXTEND;
52             t(t.LAST) := r;
53         END LOOP;
54     END procedura1_pachet;
55
56     -- preluarea datelor din tabelul INSOTITORI
57     -- si introducerea lor in tablou
58     PROCEDURE procedura2_pachet IS
59         r angajati_record;
60         CURSOR c IS
61             SELECT id_insotitor_de_zbor FROM INSOTITORI;
```



```

61 BEGIN
62     FOR i IN c LOOP
63         SELECT nume, prenume, cod_aeroport, salariu, 'insotitor de
zbor'
64             INTO r.nume, r.prenume, r.cod_aeroport, r.salariu, r.
titlu_job
65         FROM INSOTITORI WHERE id_insotitor_de_zbor = i.
id_insotitor_de_zbor;
66
67         t.EXTEND;
68         t(t.LAST) := r;
69     END LOOP;
70 END procedura2_pachet;
71
72 -- functie ce calculeaza media salariilor angajatilor
73 FUNCTION functie1_pachet RETURN NUMBER IS
74     salariu NUMBER := 0;
75 BEGIN
76     FOR i IN t.FIRST..t.LAST LOOP
77         salariu := salariu + t(i).salariu;
78     END LOOP;
79
80     salariu := ROUND(salariu / t.COUNT);
81
82     RETURN salariu;
83 END functie1_pachet;
84
85 -- functie ce calculeaza numarul de angajati
86 -- care au salariul mai mare decat media salariilor
87 FUNCTION functie2_pachet RETURN NUMBER IS
88     contor NUMBER := 0;
89 BEGIN
90     FOR i IN t.FIRST..t.LAST LOOP
91         IF (t(i).salariu > functie1_pachet)
92             THEN contor := contor + 1;
93         END IF;
94     END LOOP;
95     RETURN contor;
96 END functie2_pachet;
97
98 END pachet2;
99 /
100
101 -- Se vor rula cele 3 comenzi de mai jos in aceasta ordine
102 EXECUTE pachet2.procedura1_pachet;
103 EXECUTE pachet2.procedura2_pachet;
104
105 -- Numarul de angajati cu salariul peste medie
106 SELECT pachet2.functie2_pachet FROM dual;
107
108 -- Pentru resetarea valorilor din tablou se
109 -- va da DROP la pachet si se va recompila
110 DROP PACKAGE pachet2;

```

PROJECT_SGEO.sql

SQL Worksheet: History

PROJECT_SGEO

Worksheet Query Builder

```
1300 -- care au salariul mai mare decat media salariilor
1301 FUNCTION functie2_pachet RETURN NUMBER IS
1302     contor NUMBER := 0;
1303 BEGIN
1304     FOR i IN t.FIRST..t.LAST LOOP
1305         IF (t(i).salariu > functiei1_pachet)
1306             THEN contor := contor + 1;
1307         END IF;
1308     END LOOP;
1309     RETURN contor;
1310 END functie2_pachet;
1311
1312 END pachet2;
1313 /
1314
1315 -- Se vor rula cele 3 comenzi de mai jos in aceasta ordine
1316 EXECUTE pachet2.procedural_pachet;
1317 EXECUTE pachet2.procedura2_pachet;
1318
1319 -- Numarul de angajati cu salariul peste medie
1320 SELECT pachet2.functie2_pachet FROM dual;
```

Script Output x Query Result: x

All Rows Fetched: 1 in 0.004 seconds

1	FUNCTION2_PACHET
1	11