

Proiectarea unei Baze de Date a Aeroporturilor din România

Realizat de: Lăzăroiu Teodora-Bianca, grupa 241

Contents

1	Descrierea modelului	2
2	Diagrama Entitate-Relație	3
3	Diagrama Conceptuală	3
4	Crearea tabelelor și a constrângerilor	4
5	Inserarea datelor	8
6	Problema cu subprogram stocat și două tipuri de colecție	26
7	Problema cu subprogram stocat și un cursor	28
8	Problema cu funcție și 3 tabele într-o comandă	30
9	Problema cu procedură și 5 tabele într-o comandă	35
10	Trigger de tip LMD la nivel de comandă	37
11	Trigger de tip LMD la nivel de linie	38
12	Trigger de tip LDD	39
13	Pachet cu obiectele din proiect	40
14	Pachet cu flux de acțiuni integrate	45

1 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.

2 Diagrama Entitate-Relație

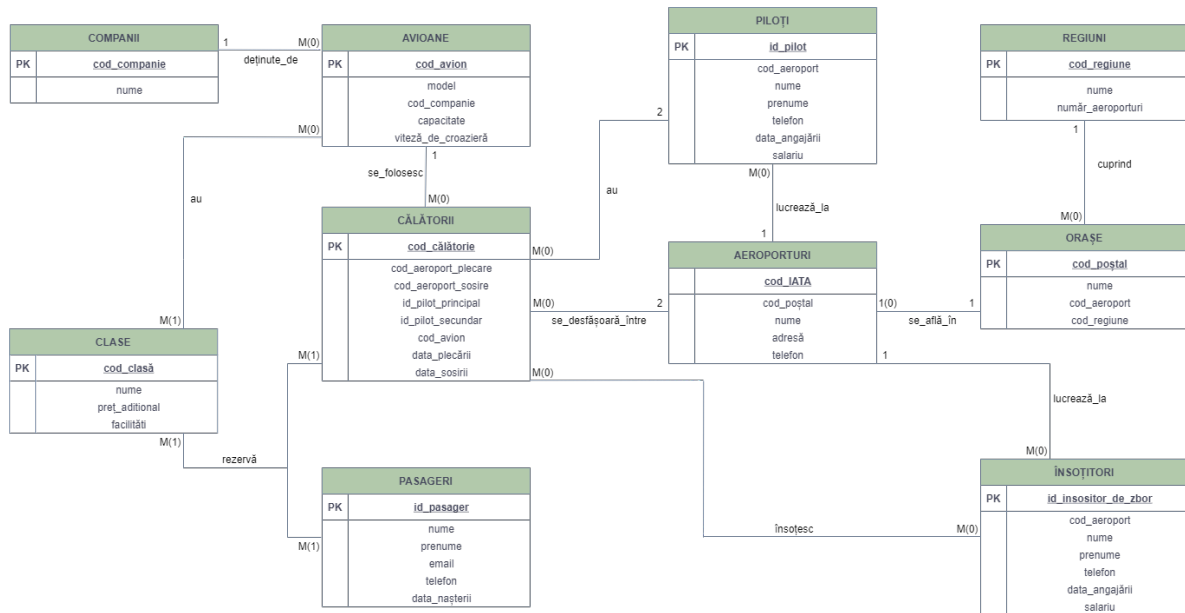


Figure 1: Diagrama Entitate-Relație

3 Diagrama Conceptuală

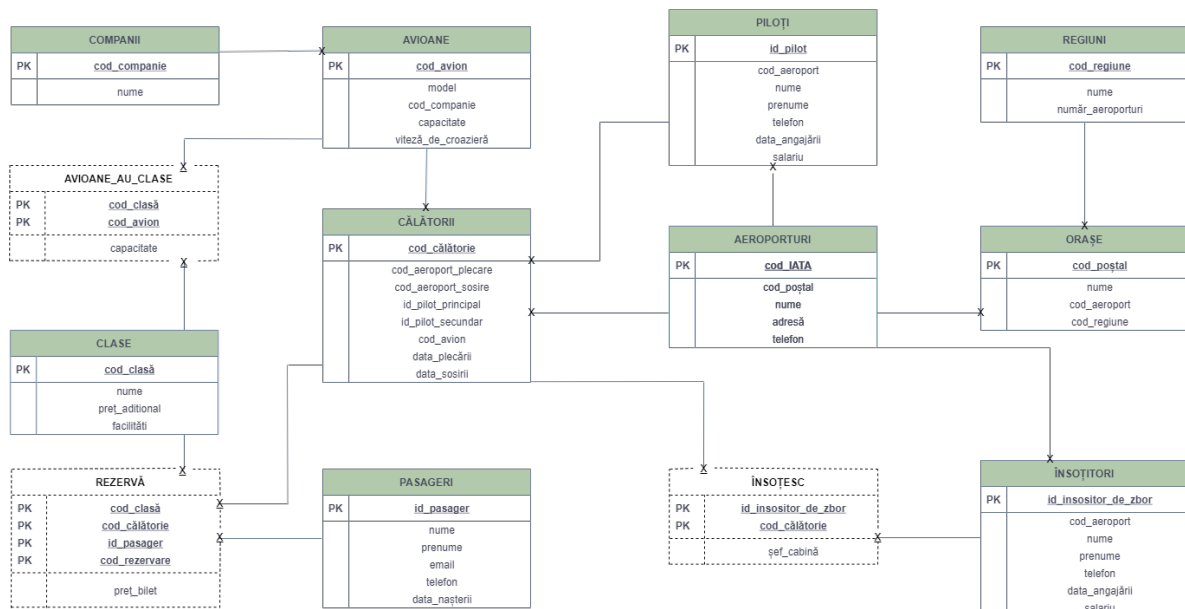


Figure 2: Diagrama Conceptuală

4 Crearea tabelelor și a constrângerilor

--- Vom crea cele 13 tabele în SQL si vom adauga constrângerile pentru modelul de date

```
CREATE TABLE AEROPORTURI
(cod_IATA varchar2(3) CONSTRAINT aeroporturi_pk PRIMARY KEY,
cod_postal varchar2(6) NOT NULL,
nume varchar2(55) NOT NULL,
adresa varchar2(50),
telefon varchar2(10));
```

```
CREATE TABLE ORASE
(cod_postal varchar2(6) CONSTRAINT orase_pk PRIMARY KEY,
nume varchar2(15) NOT NULL,
cod_aeroport varchar2(3),
cod_regiune varchar2(3)); -- cheie externa din tabelul AEROPORTURI
```

```
-- odata ce a fost creat tabelul ORASE putem marca cheia externa in tabelul AEROPORTURI
ALTER TABLE AEROPORTURI
ADD CONSTRAINT aeroporturi_orase_fk FOREIGN KEY(cod_postal)
REFERENCES ORASE(cod_postal);
```

```
CREATE TABLE REGIUNI
(cod_regiune varchar2(3) CONSTRAINT regiuni_pk PRIMARY KEY,
nume varchar2(15) NOT NULL,
numar_aeroporturi number(2));
```

```
CREATE TABLE PILOTI
(id_pilot number(4) CONSTRAINT piloti_pk PRIMARY KEY,
cod_aeroport varchar2(3),
nume varchar2(30) NOT NULL,
prenume varchar2(30),
telefon varchar2(10) NOT NULL,
data_angajarii date DEFAULT sysdate,
salariu number(6),
CONSTRAINT piloti_aeroport_fk FOREIGN KEY(cod_aeroport)
REFERENCES AEROPORTURI(cod_IATA));
```

```
CREATE TABLE INSOTITORI
(id_insotitor_de_zbor number(4) CONSTRAINT insotitori_pk PRIMARY KEY,
cod_aeroport varchar2(3),
nume varchar2(30) NOT NULL,
prenume varchar2(30),
telefon varchar2(10) NOT NULL,
data_angajarii date DEFAULT sysdate,
salariu number(6),
CONSTRAINT insotitori_aeroport_fk FOREIGN KEY(cod_aeroport)
REFERENCES AEROPORTURI(cod_IATA));
```

```

CREATE TABLE CALATORII
(cod_calatorie number(6) CONSTRAINT calatorii_pk PRIMARY KEY,
cod_aeroport_plecure varchar2(3) NOT NULL,
cod_aeroport_sosire varchar2(3) NOT NULL,
id_pilot_principal number(4) NOT NULL,
id_pilot_secundar number(4) NOT NULL,
cod_avion number(4) NOT NULL,
data_plecarii date,
data_sosirii date,
CONSTRAINT calatorii_aeroport_fk1 FOREIGN KEY(cod_aeroport_plecure)
REFERENCES AEROPORTURI(cod_IATA),
CONSTRAINT calatorii_aeroport_fk2 FOREIGN KEY(cod_aeroport_sosire)
REFERENCES AEROPORTURI(cod_IATA),
-- cele doua coduri ale aeroporturilor vor fi chei externe din tabelul AEROPORTURI
CONSTRAINT calatorii_piloti_fk1 FOREIGN KEY(id_pilot_principal)
REFERENCES PILOTI(id_pilot),
CONSTRAINT calatorii_piloti_fk2 FOREIGN KEY(id_pilot_secundar)
REFERENCES PILOTI(id_pilot),
-- cele doua id-uri vor fi chei externe din tabelul PILOTI
CONSTRAINT valid_data_sosirii CHECK(data_sosirii > data_plecarii),
-- data sosirii trebuie sa fie mai mare decat data plecarii
CONSTRAINT distinct_aeroport CHECK (cod_aeroport_plecure != cod_aeroport_sosire),
-- aeroportul de plecare si cel de sosire trebuie sa fie diferite
CONSTRAINT distinct_pilot CHECK (id_pilot_principal != id_pilot_secundar)
-- trebuie sa existe 2 piloti distincti
);

```

```

CREATE TABLE AVIOANE
(cod_avion number(4) CONSTRAINT avioane_pk PRIMARY KEY,
model varchar2(30) NOT NULL,
cod_companie number(4) NOT NULL,
capacitate number(3),
viteza_de_croaziera number(4));

```

```

-- dupa ce am creat tabelul AVIOANE putem sa referim cheia externa "cod_avion"
-- in tabelul CALATORII

```

```

ALTER TABLE CALATORII
ADD CONSTRAINT calatorii_avioane_fk FOREIGN KEY(cod_avion)
REFERENCES AVIOANE(cod_avion);

```

```

CREATE TABLE COMPANII
(cod_companie number(4) CONSTRAINT companii_pk PRIMARY KEY,
nume varchar2(30) NOT NULL);

```

```

-- dupa ce am creat tabelul COMPANII putem sa referim cheia externa
-- "cod_companie" in tabelul AVIOANE

```

```

ALTER TABLE AVIOANE
ADD CONSTRAINT avioane_companii_fk FOREIGN KEY(cod_companie)
REFERENCES COMPANII(cod_companie);

CREATE TABLE CLASE
(cod_clasa number(2) CONSTRAINT clase_pk PRIMARY KEY,
nume varchar2(20) NOT NULL,
pret_aditional number(4),
facilitati varchar2(60));

CREATE TABLE PASAGERI
(id_pasager number(6) CONSTRAINT pasageri_pk PRIMARY KEY,
nume varchar2(30) NOT NULL,
prenume varchar2(30),
email varchar2(30),
telefon varchar2(10) NOT NULL,
data_nasterii date);

CREATE TABLE AVIOANE_AU_CLASE
(cod_clasa number(2),
cod_avion number(4),
capacitate number(3),
CONSTRAINT avioane_au_clase_clase_fk FOREIGN KEY(cod_clasa)
REFERENCES CLASE(cod_clasa),
-- referim cheia externa din tabelul CLASE
CONSTRAINT avioane_au_clase_avioane_fk FOREIGN KEY(cod_avion)
REFERENCES AVIOANE(cod_avion),
-- referim cheia externa din tabelul AVIOANE
CONSTRAINT avioane_au_clase_pk PRIMARY KEY(cod_clasa, cod_avion));
-- cheie primara compusa

CREATE TABLE REZERVA
(cod_clasa number(2),
cod_calatorie number(6),
id_pasager number(6),
cod_rezervare number(8),
pret_bilet number(4),
CONSTRAINT rezerva_clase_fk FOREIGN KEY(cod_clasa)
REFERENCES CLASE(cod_clasa),
-- cheia externa din tabelul CLASE
CONSTRAINT rezerva_calatorii_fk FOREIGN KEY(cod_calatorie)
REFERENCES CALATORII(cod_calatorie),
-- cheia externa din tabelul CALATORII
CONSTRAINT rezerva_pasageri_fk FOREIGN KEY(id_pasager)
REFERENCES PASAGERI(id_pasager),
-- cheia externa din tabelul PASAGERI
CONSTRAINT rezerva_pk PRIMARY KEY(cod_clasa, cod_calatorie, id_pasager, cod_rezervare));

```

-- cheie primara compusa din 4 elemente

```
CREATE TABLE INSOTESC
(id_insotitor_de_zbor number(4),
cod_calatorie number(6),
sef_cabina number(4),
CONSTRAINT insotesc_insotitori_fk1 FOREIGN KEY(id_insotitor_de_zbor)
REFERENCES INSOTITORI(id_insotitor_de_zbor),
-- cheie externa din tabelul INSOTITORI
CONSTRAINT insotesc_calatorii_fk FOREIGN KEY(cod_calatorie)
REFERENCES CALATORII(cod_calatorie),
-- cheie externa din tabelul CALATORII
CONSTRAINT insotesc_insotitori_fk2 FOREIGN KEY(sef_cabina)
REFERENCES INSOTITORI(id_insotitor_de_zbor),
-- cheie externa din tabelul INSOTITORI
CONSTRAINT insotesc_pk PRIMARY KEY(id_insotitor_de_zbor, cod_calatorie));
-- cheie primara compusa
```

5 Inserarea datelor

--- Inserari in tabelul REGIUNI

```
INSERT INTO REGIUNI
```

```
VALUES ('MUN', 'Muntenia', 1);
```

```
INSERT INTO REGIUNI
```

```
VALUES ('TRA', 'Transilvania', 2);
```

```
INSERT INTO REGIUNI
```

```
VALUES ('MOL', 'Moldova', 1);
```

```
INSERT INTO REGIUNI
```

```
VALUES ('DOB', 'Dobrogea', 1);
```

```
INSERT INTO REGIUNI
```

```
VALUES ('BAN', 'Banat', 1);
```

```
INSERT INTO REGIUNI
```

```
VALUES ('OLT', 'Oltenia', 1);
```

```
INSERT INTO REGIUNI
```

```
VALUES ('CRI', 'Crisana', 1);
```

```
INSERT INTO REGIUNI
```

```
VALUES ('MAR', 'Maramures', 0);
```

--- Inserari in tabelul ORASE

```
INSERT INTO ORASE
```

```
VALUES (075150, 'Bucuresti', 'OTP', 'MUN');
```

```
INSERT INTO ORASE
```

```
VALUES (400397, 'Cluj', 'CLJ', 'TRA');
```

```
INSERT INTO ORASE
```

```
VALUES (700750, 'Iasi', 'IAS', 'MOL');
```

```
INSERT INTO ORASE
```

```
VALUES (907195, 'Constanta', 'CND', 'DOB');
```

```
INSERT INTO ORASE
```

```
VALUES (307200, 'Timisoara', 'TSR', 'BAN');
```

```
INSERT INTO ORASE
```

```
VALUES (410223, 'Oradea', 'OMR', 'CRI');
```

```
INSERT INTO ORASE
```

```
VALUES (200621, 'Craiova', 'CRA', 'OLT');
```



```

INSERT INTO ORASE
VALUES (550052, 'Sibiu', 'SBZ', 'TRA');

INSERT INTO ORASE
VALUES (110001, 'Pitesti', NULL, 'MUN');

--- Inserari in tabelul AEROPORTURI
INSERT INTO AEROPORTURI
VALUES('OTP', 075150, 'Henri Coanda International Airport', 'Calea Bucurestilor
↳ 224E', '0212041000');

INSERT INTO AEROPORTURI
VALUES('CLJ', 400397, 'Avram Iancu International Airport', 'Strada Traian Vuia
↳ 149-151', '0264307500');

INSERT INTO AEROPORTURI
VALUES('IAS', 700750, 'Iasi International Airport', 'Strada Moara de Vant 34',
↳ '0232271590');

INSERT INTO AEROPORTURI
VALUES('CND', 907195, 'Mihail Kogalniceanu International Airport', 'Strada
↳ Tudor Vladimirescu', '0241255100');

INSERT INTO AEROPORTURI
VALUES('TSR', 307200, 'Traian Vuia International Airport', 'Strada Aeroportului
↳ 2', '0256386089');

INSERT INTO AEROPORTURI
VALUES('OMR', 410223, 'Oradea International Airport', 'Calea Aradului 80',
↳ '0259416082');

INSERT INTO AEROPORTURI
VALUES('CRA', 200621, 'Craiova International Airport', 'Calea Bucuresti 325',
↳ '0754022508');

INSERT INTO AEROPORTURI
VALUES('SBZ', 550052, 'Sibiu International Airport', 'Soseaua Alba Iulia 73',
↳ '0269253135');

--- Adaugam constrangerea de FOREIGN KEY pentru cod_aeroporturi din tabelul
↳ ORASE
ALTER TABLE ORASE
ADD CONSTRAINT orase_aeroporturi_fk FOREIGN KEY(cod_aeroport)
REFERENCES AEROPORTURI(cod_IATA);

--- Adaugam constrangerea de FOREIGN KEY pentru cod_regiune din tabelul ORASE
ALTER TABLE ORASE

```

```
ADD CONSTRAINT orase_regiuni_fk FOREIGN KEY(cod_regiune)
REFERENCES REGIUNI(cod_regiune);
```

```
--- secventa pentru incrementarea atributului ID_PILOT din tabelul PILOTI
CREATE SEQUENCE SEQ_PILOTI
START WITH 120
INCREMENT by 10
MAXVALUE 9990
NOCYCLE;
```

```
--- Inserari in tabelul PILOTI
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'OTP', 'Popescu', 'Mihai', '0723467897',
↪ to_date('04-05-2018','DD-MM-YYYY'), 100000);
```

```
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'OTP', 'Dumitru', 'Andrei', '0773567890',
↪ to_date('07-09-2018','DD-MM-YYYY'), 150000);
```

```
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'TSR', 'Marinescu', 'Adrian', '0793857856',
↪ to_date('05-02-2016','DD-MM-YYYY'), 550000);
```

```
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'TSR', 'Coman', 'Stefania', '0775852235',
↪ to_date('05-12-2020','DD-MM-YYYY'), 200000);
```

```
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'CLJ', 'Florescu', 'Mihai', '0788451775',
↪ to_date('07-12-2010','DD-MM-YYYY'), 500000);
```

```
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'CLJ', 'Moise', 'Horia', '0788449075',
↪ to_date('23-06-2015','DD-MM-YYYY'), 350000);
```

```
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'CLJ', 'Stoica', 'Irina', '0702390175',
↪ to_date('15-01-2021','DD-MM-YYYY'), 180000);
```

```
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'IAS', 'Naum', 'Florin', '0712345175',
↪ to_date('15-11-2019','DD-MM-YYYY'), 250000);
```

```
INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'IAS', 'Olteanu', 'Paul', '0712743275',
↪ to_date('01-01-2014','DD-MM-YYYY'), 550000);
```

```
INSERT INTO PILOTI
```

```

VALUES (SEQ_PILOTI.NEXTVAL, 'CND', 'Avram', 'Daniel', '0766320275',
↪ to_date('01-06-2020','DD-MM-YYYY'), 222000);

INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'CND', 'Stefan', 'George', '0766320275',
↪ to_date('30-08-2013','DD-MM-YYYY'), 355000);

INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'CND', 'Stancu', 'Elena', '0733332146',
↪ to_date('01-05-2021','DD-MM-YYYY'), 99000);

INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'CRA', 'Mihailescu', 'Cosmin', '0754999146',
↪ to_date('01-05-2018','DD-MM-YYYY'), 340000);

INSERT INTO PILOTI
VALUES (SEQ_PILOTI.NEXTVAL, 'CRA', 'Olteanu', 'Valentin', '0765412146',
↪ to_date('17-04-2010','DD-MM-YYYY'), 450000);

--- secventa pentru incrementarea atributului ID_insotitor_DE_ZBOR din tabelul
↪ INSOTITORI
CREATE SEQUENCE SEQ_INSOTITORI
START WITH 10
INCREMENT by 10
MINVALUE 0
MAXVALUE 9990
NOCYCLE;

---Inserari in tabelul INSOTITORI
INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'OTP', 'Voiculescu', 'Alina', '0745678146',
↪ to_date('04-10-2018','DD-MM-YYYY'), 33000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'OTP', 'Moiescu', 'Denisa', '0792218597',
↪ to_date('13-03-2019','DD-MM-YYYY'), 40000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'OTP', 'Ionescu', 'Florin', '0726688610',
↪ to_date('02-04-2012','DD-MM-YYYY'), 90000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'CLJ', 'Georgescu', 'Calin', '0768216950',
↪ to_date('01-07-2016','DD-MM-YYYY'), 80000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'CLJ', 'Antonie', 'Mihaela', '0783717731',
↪ to_date('06-12-2020','DD-MM-YYYY'), 15000);

```

```

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'TSR', 'Voica', 'Laura', '0766138430',
↪ to_date('31-01-2021','DD-MM-YYYY'), 13000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'TSR', 'Enache', 'Maria', '0778767173',
↪ to_date('25-10-2015','DD-MM-YYYY'), 40000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'IAS', 'Matache', 'Irina', '0762706994',
↪ to_date('20-03-2015','DD-MM-YYYY'), 45000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'IAS', 'Mirea', 'Mihai', '0785072934',
↪ to_date('07-07-2019','DD-MM-YYYY'), 27000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'CND', 'Lazar', 'Nicoleta', '0718538567',
↪ to_date('09-11-2018','DD-MM-YYYY'), 33000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'CND', 'Florea', 'Stefania', '0784725925',
↪ to_date('23-09-2009','DD-MM-YYYY'), 80000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'CRA', 'Miulescu', 'Ana', '0758812546',
↪ to_date('06-06-2020','DD-MM-YYYY'), 12000);

INSERT INTO INSOTITORI
VALUES (SEQ_INSOTITORI.NEXTVAL, 'SBZ', 'Diaconescu', 'Daniela', '0734456845',
↪ to_date('27-04-2018','DD-MM-YYYY'), 16000);

---Inserari in tabelul COMPANII
INSERT INTO COMPANII
VALUES (120, 'Tarom');

INSERT INTO COMPANII
VALUES (121, 'Blue Air');

INSERT INTO COMPANII
VALUES (122, 'Wizz Air');

INSERT INTO COMPANII
VALUES (123, 'KLM');

INSERT INTO COMPANII
VALUES (124, 'Ryanair');

```

```
INSERT INTO COMPANII  
VALUES (125, 'Lufthansa');
```

---Inserari in tabelul AVIOANE

```
INSERT INTO AVIOANE  
VALUES (1000, 'BOEING 737-700', 120, 149, 870);
```

```
INSERT INTO AVIOANE  
VALUES (1001, 'BOEING 737-300', 120, 141, 800);
```

```
INSERT INTO AVIOANE  
VALUES (1002, 'AIRBUS A318-111', 120, 132, 850);
```

```
INSERT INTO AVIOANE  
VALUES (1003, 'ATR 42-500', 120, 52, 550);
```

```
INSERT INTO AVIOANE  
VALUES (1004, 'BOEING 737-300', 121, 141, 800);
```

```
INSERT INTO AVIOANE  
VALUES (1005, 'BOEING 737-500', 121, 126, 785);
```

```
INSERT INTO AVIOANE  
VALUES (1006, 'BOEING 737-800', 121, 189, 823);
```

```
INSERT INTO AVIOANE  
VALUES (1007, 'AIRBUS A320-200', 122, 186, 900);
```

```
INSERT INTO AVIOANE  
VALUES (1008, 'AIRBUS A321-200', 122, 230, 840);
```

```
INSERT INTO AVIOANE  
VALUES (1009, 'AIRBUS A330-200', 123, 185, 913);
```

```
INSERT INTO AVIOANE  
VALUES (1010, 'BOEING 737-800', 124, 162, 946);
```

--- secventa pentru incrementarea atributului COD_CALATORIE din tabelul

↳ CALATORII

```
CREATE SEQUENCE SEQ_CALATORII  
START WITH 1001  
INCREMENT by 1  
MAXVALUE 999999  
NOCYCLE;
```

---Inserari in tabelul CALATORII

```
INSERT INTO CALATORII
```

```
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'));
```

```
INSERT INTO CALATORII
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'));
```

```
INSERT INTO CALATORII
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'));
```

```
INSERT INTO CALATORII
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'));
```

```
INSERT INTO CALATORII
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'));
```

```
INSERT INTO CALATORII
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'));
```

```
INSERT INTO CALATORII
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'));
```

```
INSERT INTO CALATORII
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'));
```

```
INSERT INTO CALATORII
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'));
```

```
INSERT INTO CALATORII
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'));
```

```

INSERT INTO CALATORII
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'));

```

```

INSERT INTO CALATORII
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'));

```

```

INSERT INTO CALATORII
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'));

```

---Inserari in tabelul CLASE

```

INSERT INTO CLASE
VALUES (10, 'Economica', NULL, 'masuta retractabila');

```

```

INSERT INTO CLASE
VALUES (20, 'Business', 300, 'scaune ajustabile, o masa calda');

```

```

INSERT INTO CLASE
VALUES (30, 'First class', 500, 'lounge, meniu variat, paturi');

```

```

INSERT INTO CLASE
VALUES (40, 'Business Premium', 800, 'facilitati business, lounge');

```

```

INSERT INTO CLASE
VALUES (50, 'First Class Premium', 1000, 'facilitati first class, meniu lux');

```

--- secventa pentru incrementarea atributului ID_PASAGER din tabelul PASAGERI

```

CREATE SEQUENCE SEQ_PASAGERI
START WITH 1
INCREMENT by 1
MAXVALUE 999999
NOCYCLE;

```

---Inserari in tabelul PASAGERI

```

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Ionita', 'Alexandra', 'aionita@yahoo.com',
↪ '0741974529', to_date('03-05-2001','dd-mm-yyyy'));

```

```

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Moldoveanu', 'Victor', 'victor_m@yahoo.com',
↪ '0785483431', to_date('08-03-1980','dd-mm-yyyy'));

```

```

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Stoica', 'Valentin', 'valistoica@gmail.com',
↪ '0791459679', to_date('14-12-1996','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Dabija', 'Adrian', 'dabadi@yahoo.com',
↪ '0760523629', to_date('06-07-1990','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Barbu', 'Dumitru', 'dumitruB@yahoo.com',
↪ '0746888320', to_date('13-05-2000','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Teodorescu', 'Eugen', 'eugen_t@yahoo.com',
↪ '0770522210', to_date('23-11-1965','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Lupu', 'Maria', 'maria_lupu@yahoo.com',
↪ '0757241888', to_date('03-12-1977','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Sava', 'Florentina', 'savaflo@gmail.com',
↪ '0766560779', to_date('29-06-2002','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Nistor', 'Mariana', 'mariananistor@gmail.com',
↪ '0745582866', to_date('03-05-2001','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Barbu', 'Ana Maria', 'anamariab@yahoo.com',
↪ '0788821557', to_date('31-01-1992','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Dumitrescu', 'Andreea',
↪ 'andreea_dumi@yahoo.com', '0749267946', to_date('07-07-1979','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Gheorghiu', 'Mihaela', 'mihaela.g@yahoo.com',
↪ '0727774485', to_date('24-11-1969','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Pop', 'Rodica', 'rodica_pop@yahoo.com',
↪ '0791567138', to_date('16-04-1999','dd-mm-yyyy'));

INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Constantinescu', 'Roxana', 'roxana.c@yahoo.com',
↪ '0743479564', to_date('12-12-1982','dd-mm-yyyy'));

```



```
INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Dobre', 'Gabriela', 'gabi_dobre@gmail.com',
↪ '0765870667', to_date('28-02-1997','dd-mm-yyyy'));
```

```
INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Ciobanu', 'Adriana', 'a.ciobanu@yahoo.com',
↪ '0732936930', to_date('03-11-1960','dd-mm-yyyy'));
```

```
INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Toma', 'Claudia', 'claudia.toma@yahoo.com',
↪ '0736630215', to_date('16-09-1995','dd-mm-yyyy'));
```

```
INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Stanescu', 'Constantin',
↪ 'ctin_stanescu@gmail.com', '0756729015',
↪ to_date('08-08-1955','dd-mm-yyyy'));
```

```
INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Dima', 'Gheorghe', 'dima.gh@gmail.com',
↪ '0774034758', to_date('16-04-1968','dd-mm-yyyy'));
```

```
INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Dumitrescu', 'Cristina', 'cristinaD@gmail.com',
↪ '0784550324', to_date('15-05-2000','dd-mm-yyyy'));
```

```
INSERT INTO PASAGERI
VALUES (SEQ_PASAGERI.NEXTVAL, 'Popescu', 'Elena', 'elenapop@yahoo.com',
↪ '0781236689', to_date('01-06-1985','dd-mm-yyyy'));
```

```
---Inserari in tabelul AVIOANE_AU_CLASE
INSERT INTO AVIOANE_AU_CLASE
VALUES (10, 1000, 100);
```

```
INSERT INTO AVIOANE_AU_CLASE
VALUES (20, 1000, 49);
```

```
INSERT INTO AVIOANE_AU_CLASE
VALUES (10, 1001, 141);
```

```
INSERT INTO AVIOANE_AU_CLASE
VALUES (10, 1002, 132);
```

```
INSERT INTO AVIOANE_AU_CLASE
VALUES (10, 1003, 52);
```

```
INSERT INTO AVIOANE_AU_CLASE
VALUES (10, 1004, 80);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (20, 1004, 61);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (10, 1005, 126);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (10, 1006, 100);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (20, 1006, 40);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (30, 1006, 49);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (10, 1007, 160);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (20, 1007, 26);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (10, 1008, 100);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (20, 1008, 100);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (30, 1008, 20);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (40, 1008, 10);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (10, 1009, 185);
```

```
INSERT INTO AVIOANE_AU_CLASE  
VALUES (10, 1010, 162);
```

```
---Inserari in tabelul REZERVA  
INSERT INTO REZERVA  
VALUES (30, 1001, 2, 100001, 300);
```

```
INSERT INTO REZERVA  
VALUES (30, 1001, 3, 100002, 300);
```

```
INSERT INTO REZERVA  
VALUES (10, 1001, 15, 100003, 300);
```

```

INSERT INTO REZERVA
VALUES (10, 1010, 7, 100004, 200);

INSERT INTO REZERVA
VALUES (10, 1010, 10, 100005, 200);

INSERT INTO REZERVA
VALUES (10, 1010, 11, 100006, 200);

INSERT INTO REZERVA
VALUES (20, 1006, 5, 100007, 250);

INSERT INTO REZERVA
VALUES (40, 1006, 12, 100008, 250);

INSERT INTO REZERVA
VALUES (40, 1006, 13, 100009, 250);

INSERT INTO REZERVA
VALUES (30, 1006, 14, 100010, 250);

INSERT INTO REZERVA
VALUES (20, 1012, 1, 100011, 250);

INSERT INTO REZERVA
VALUES (10, 1012, 17, 100012, 250);

INSERT INTO REZERVA
VALUES (10, 1012, 18, 100013, 250);

INSERT INTO REZERVA
VALUES (10, 1012, 20, 100014, 250);

INSERT INTO REZERVA
VALUES (10, 1004, 4, 100015, 270);

INSERT INTO REZERVA
VALUES (10, 1004, 6, 100016, 270);

INSERT INTO REZERVA
VALUES (10, 1003, 16, 100017, 150);

INSERT INTO REZERVA
VALUES (20, 1003, 8, 100018, 150);

INSERT INTO REZERVA
VALUES (20, 1003, 9, 100019, 150);

```

```
INSERT INTO REZERVA
VALUES (10, 1013, 19, 100020, 150);
```

```
INSERT INTO REZERVA
VALUES (10, 1007, 10, 100021, 240);
```

```
INSERT INTO REZERVA
VALUES (10, 1007, 11, 100022, 240);
```

```
INSERT INTO REZERVA
VALUES (10, 1011, 2, 100023, 350);
```

```
INSERT INTO REZERVA
VALUES (10, 1011, 3, 100024, 350);
```

```
INSERT INTO REZERVA
VALUES (10, 1008, 19, 100025, 275);
```

```
INSERT INTO REZERVA
VALUES (10, 1008, 17, 100026, 275);
```

```
INSERT INTO REZERVA
VALUES (10, 1008, 18, 100027, 275);
```

---Inserari in tabelul INSOTESC

```
INSERT INTO INSOTESC
VALUES (10, 1001, 20);
```

```
INSERT INTO INSOTESC
VALUES (20, 1001, NULL);
```

```
INSERT INTO INSOTESC
VALUES (30, 1001, 20);
```

```
INSERT INTO INSOTESC
VALUES (10, 1003, NULL);
```

```
INSERT INTO INSOTESC
VALUES (30, 1003, 10);
```

```
INSERT INTO INSOTESC
VALUES (10, 1004, 20);
```

```
INSERT INTO INSOTESC
VALUES (20, 1004, NULL);
```

```
INSERT INTO INSOTESC
```

```
VALUES (40, 1005, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (40, 1006, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (50, 1006, 40);
```

```
INSERT INTO INSOTESC  
VALUES (80, 1007, 90);
```

```
INSERT INTO INSOTESC  
VALUES (90, 1007, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (90, 1008, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (120, 1009, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (120, 1010, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (60, 1011, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (70, 1011, 60);
```

```
INSERT INTO INSOTESC  
VALUES (60, 1012, 70);
```

```
INSERT INTO INSOTESC  
VALUES (70, 1012, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (20, 1013, NULL);
```

```
INSERT INTO INSOTESC  
VALUES (30, 1013, 20);
```

```
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

6 Problema cu subprogram stocat și două tipuri de colecție

```
--- Pentru fiecare aeroport sa se afiseze numele acestuia
--- si salariul mediu al pilotilor ce lucreaza acolo

/
CREATE OR REPLACE PROCEDURE procedura1 IS

-- Vom crea un tip de date inregistrare pentru a pastra
-- codul aeroportului, numele acestuia si salariul mediu
    TYPE aeroport_record IS RECORD
        (cod AEROPORTURI.cod_iata%TYPE,
         nume AEROPORTURI.nume%TYPE,
         salariu_mediu PILOTI.salariu%TYPE);

-- Datele pentru toate aeroporturile vor fi salvate intr-un
-- tablou imbricat ce contine o coloana de tip inregistrare creat mai sus
    TYPE tablou_imbricat IS TABLE OF aeroport_record;

-- Vom folosi un vector pentru salvarea si calcularea
-- fiecarui salariu mediu al pilotilor in functie de aeroport
    TYPE vector IS VARRAY(8) OF PILOTI.salariu%TYPE;

-- Cu ajutorul unui circlu cursor vom parcurge
-- toate codurile de aeroporturi
    CURSOR c IS
        SELECT cod_iata
        FROM AEROPORTURI;

    t tablou_imbricat := tablou_imbricat();
    r aeroport_record;
    v vector := vector();
BEGIN

    FOR i IN c LOOP

-- Initial vom salva codul si numele aeroporturilor
-- in variabila de tip inregistrare declarata mai sus
        SELECT cod_iata, nume
        INTO r.cod, r.nume
        FROM AEROPORTURI
        WHERE i.cod_iata = cod_iata;

-- Calculam salariul mediu in functie de aeroport si
-- il salvam intr-un element din vector
        v.EXTEND;

        SELECT AVG(salariu)
```

```

        INTO v(v.LAST)
        FROM PILOTI
        WHERE cod_aeroport = i.cod_iata;

-- Copiem valoarea salariului mediu in
-- campul specific din variabila de tip inregistrare
    r.salariu_mediu := v(v.LAST);

-- Adaugam variabila de tip inregistrare curenta in
-- tabloul imbricat pentru a pastra toate valorile
    t.EXTEND;
    t(t.LAST) := r;

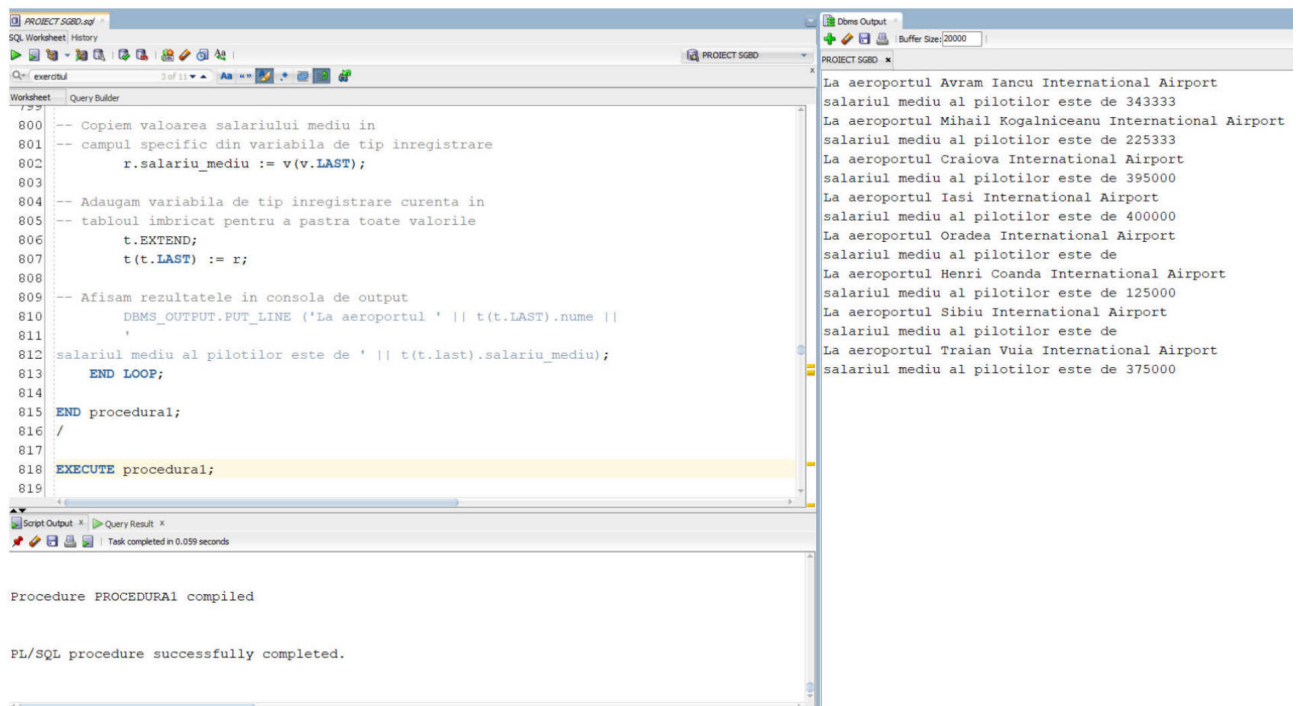
-- Afisam rezultatele in consola de output
    DBMS_OUTPUT.PUT_LINE ('La aeroportul ' || t(t.LAST).nume ||
        ' salariul mediu al pilotilor este de ' || t(t.LAST).salariu_mediu);
END LOOP;

END procedura1;

/

EXECUTE procedura1;

```



7 Problema cu subprogram stocat și un cursor

```
--- Sa se mareasca cu 10% salariul insotitorilor de zbor  
--- care au fost angajati incepand cu anul 2018
```

```
/
```

```
CREATE OR REPLACE PROCEDURE procedura2 IS
```

```
-- Vom folosi un cursor SELECT FOR UPDATE  
-- pentru a bloca liniile inainte ca acestea  
-- sa fie actualizate cu noul salariu  
    CURSOR c IS  
    SELECT * FROM INSOTITORI  
    WHERE TO_CHAR(data_angajarii, 'YYYY') >= 2018  
    FOR UPDATE OF salariu NOWAIT;
```

```
-- Prin folosirea clauzei NOWAIT va fi returnata  
-- o eroare in cazul in care liniile selectate  
-- sunt deja blocate de catre alta comanda
```

```
BEGIN
```

```
-- Se foloseste un ciclu cursor pentru a  
-- actualiza toate valorile selectate  
    FOR i IN c LOOP  
        UPDATE INSOTITORI  
        SET salariu = salariu + ((salariu * 10) / 100)  
        WHERE CURRENT OF c;  
    END LOOP;
```

```
END procedura2;
```

```
/
```

```
SELECT * FROM INSOTITORI;
```

```
EXECUTE procedura2;
```

```
SELECT * FROM INSOTITORI;
```

```
ROLLBACK;
```

PROJECT.SGRO.sql

SQL Worksheet: History

Q= exercitulu 3 of 11

Worksheet Query Builder

```

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= exercitulu 3 of 11

Worksheet Query Builder

```

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

8 Problema cu funcție și 3 tabele într-o comandă

```
-- Citindu-se numele unui pasager de la tastatura, sa se returneze
-- pretul total al rezervarii facute de acesta (pretul biletului +
-- pretul additional pentru clasa, daca este cazul)

-- In cazul in care un pasager are mai multe rezervari,
-- se va returna suma tuturor preturilor totale

-- Se vor arunca erori in cazul in care: exista mai multi pasageri
-- cu numele dat, nu exista un pasager cu numele dat sau nu
-- exista rezervari facute de catre acest pasager

/
CREATE OR REPLACE FUNCTION functie1(v_nume PASAGERI.nume%TYPE)
RETURN REZERVA.pret_bilet%TYPE IS

    pret_total REZERVA.pret_bilet%TYPE;
    contor NUMBER;
BEGIN

-- Vom trata exceptiile la inceputul programului

-- Numaram cati pasageri exista cu numele dat ca parametru
    SELECT COUNT(*) INTO contor
    FROM PASAGERI WHERE INITCAP(v_nume) = INITCAP(nume);

-- Tratatam cazul in care nu exista niciun pasager si
-- cazul in care exista mai multe pasageri cu numele dat
    IF contor = 0
    THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista un pasager cu acest nume');
    ELSIF contor > 1
    THEN RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi pasageri cu acest nume');
    END IF;

-- Folosim aceiasi variabila pentru a numara cate rezervari exista
-- pentru pasagerul dat iar daca nu exista niciuna vom arunca o eroare
    SELECT COUNT(*) INTO contor
    FROM PASAGERI RIGHT OUTER JOIN REZERVA USING (id_pasager)
    WHERE INITCAP(v_nume) = INITCAP(nume);

    IF contor = 0
    THEN RAISE_APPLICATION_ERROR(-20002, 'Nu exista rezervari pe acest nume');
    END IF;

-- Functia de SUM va trata atat cazul in care exista o singura rezervare
-- dar si cazul in care exista mai multe iar functia NVL va inlocui posibila
-- valoare NULL a clasei cu codul 10 cu valoarea 0 pentru a o adauga la suma
```

```

SELECT SUM(pret_bilet + NVL(pret_aditional, 0))
INTO pret_total
FROM REZERVA JOIN CLASE USING (cod_clasa) JOIN PASAGERI USING (id_pasager)
WHERE INITCAP(v_ume) = INITCAP(PASAGERI.ume);

-- Sunt folosite 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 functie1;
/

-- Are o singura rezervare (250 biletul + 300 pretul clasei)
SELECT functie1('IONITA') FROM DUAL;

-- Are doua rezervari: prima (300 biletul + 500 pretul clasei)
-- si a doua (350 biletul) cu totatut de 1150
SELECT functie1('Moldoveanu') FROM DUAL;

-- Nu exista un pasager cu acest nume deci vom primi eroare
SELECT functie1('Lazaroiu') FROM DUAL;

-- Exista 2 pasageri cu numele Barbu
-- deci ne va returna o eroare
SELECT functie1('Barbu') FROM DUAL;

-- Pasagerul exista insa nu are nicio rezervare
-- deci va ridica o eroare
SELECT functie1('popescu') FROM DUAL;

```

PROJECT.SGBD.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

Worksheet Query Builder

```

916 -- Sunt necesare 3 tabele pentru a accesa detalii despre pretul aditional al
917 -- clasei, numele pasagerilor si pretul biletului trecut in rezervare
918
919 -- Functia va returna valoarea calculata mai sus
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

```

Script Output x Query Result x

SQL All Rows Fetched: 1 in 0.005 seconds

FUNCTIE(IONITA)	
1	550

PROJECT SGBD

DBMS Output Buffer Size: 20000

PROJECT.SGBD.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

Worksheet Query Builder

```

916 -- Sunt necesare 3 tabele pentru a accesa detalii despre pretul aditional al
917 -- clasei, numele pasagerilor si pretul biletului trecut in rezervare
918
919 -- Functia va returna valoarea calculata mai sus
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

```

Script Output x Query Result x

SQL All Rows Fetched: 1 in 0.003 seconds

FUNCTIE(MOLDOVEANU)	
1	1150

PROJECT SGBD

DBMS Output Buffer Size: 20000

PROJECT.SGBD.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

Worksheet: Query Builder

```
916 -- Sunt 3 tabele pentru a accesa detalii despre pretul aditional al
917 -- clasei, numele pasagerilor si pretul biletului trecut in rezervare
918
919 -- Functia va returna valoarea calculata mai sus
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
```

Script Output x Query Result x

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

ORA-20000: Nu exista un pasager cu acest nume
ORA-06512: at "LAZAROIU.ODORA.FUNCTIE1", line 17
20000.00000 - "Is"
*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.

PROJECT.SGBD.sql

SQL Worksheet: History

Q= exercitiu 3 of 11

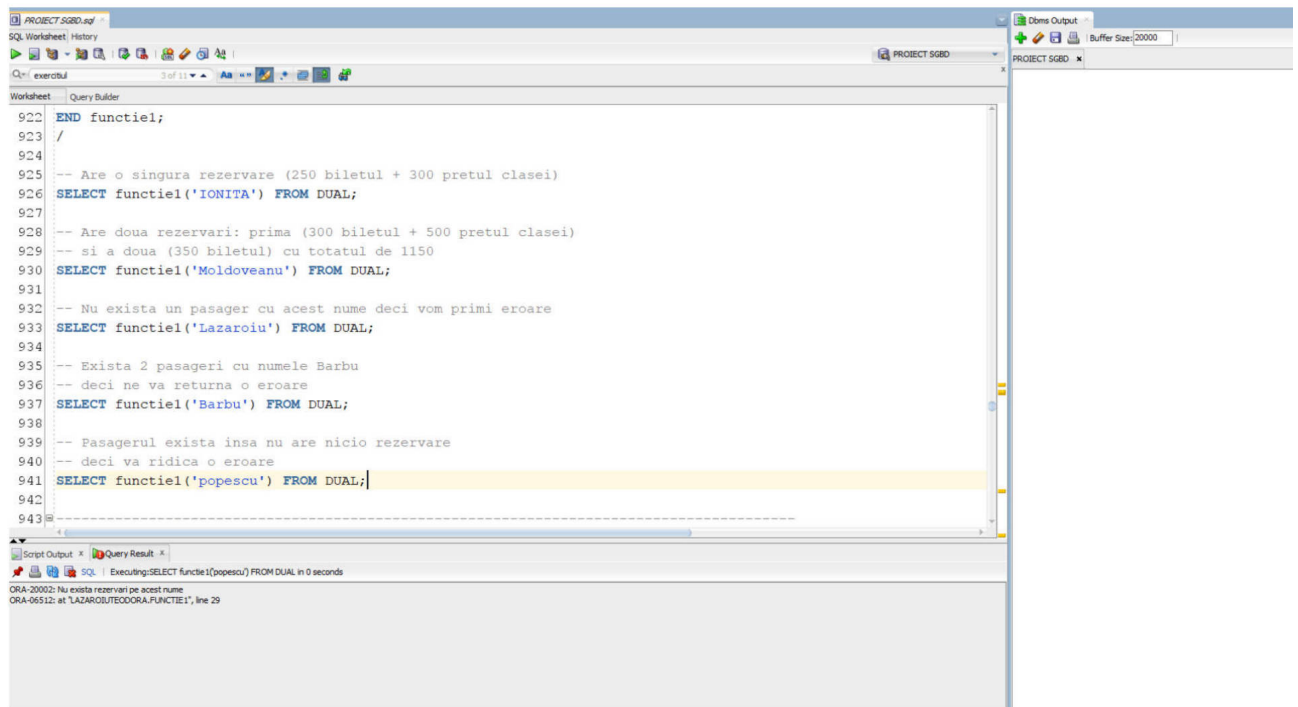
Worksheet: Query Builder

```
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.ODORA.FUNCTIE1", line 19



9 Problema cu procedură și 5 tabele într-o comandă

```
--- Se da numele unui pilot ca parametru. Sa se afiseze numarul de calatorii
--- la care acesta a participat, numele aeroportului la care lucreaza
--- si numele orasului si al regiunii in care se afla aeroportul sau

/
CREATE OR REPLACE PROCEDURE procedura3 (nume_pilot PILOTI.nume%TYPE) IS

-- Vom declara variabilele in care vom salva
-- datele care trebuiesc selectate si afisate
    numar_calatorii NUMBER(4);
    nume_aeroport AEROPORTURI.nume%TYPE;
    nume_oras ORASE.nume%TYPE;
    nume_regiune REGIUNI.nume%TYPE;

BEGIN

-- Vom folosi un RIGHT JOIN intre CALATORII si PILOTI pentru
-- a include si pilotii care nu apar in tabelul de calatorii
-- (adica nu au facut nicio calatorie) dar apartin unui aeroport.
-- Conditia de JOIN intre acestea verifica daca pilotul a fost
-- fie pilot principal, fie secundar si contorizeaza ambele cazuri

-- In continuare se face JOIN simplu cu tabelele AEROPORTURI, ORASE
-- si REGIUNI pentru a putea accesa si selecta numele acestora

    SELECT COUNT(cod_calatorie), a.nume, o.nume, r.nume
    INTO numar_calatorii, nume_aeroport, nume_oras, nume_regiune
    FROM CALATORII c RIGHT JOIN PILOTI p ON (id_pilot = id_pilot_principal OR
    id_pilot = id_pilot_secundar) JOIN AEROPORTURI a ON (p.cod_aeroport = cod_iata)
    JOIN ORASE o ON (o.cod_aeroport = cod_iata) JOIN REGIUNI r USING (cod_regiune)
    WHERE INITCAP(nume_pilot) = INITCAP(p.nume)
    GROUP BY a.nume, o.nume, r.nume;

-- Afisam in consola datele selectate mai sus
    DBMS_OUTPUT.PUT_LINE('Nume pilot: ' || INITCAP(nume_pilot));
    DBMS_OUTPUT.PUT_LINE('Numar de calatorii: ' || numar_calatorii);
    DBMS_OUTPUT.PUT_LINE('Numele aeroportului: ' || nume_aeroport);
    DBMS_OUTPUT.PUT_LINE('Orasul aeroportului in care lucreaza: ' || nume_oras);
    DBMS_OUTPUT.PUT_LINE('Regiunea in care se afla aeroportul: ' || nume_regiune);

EXCEPTION

    WHEN TOO_MANY_ROWS
    THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi piloti cu acest nume');
    WHEN NO_DATA_FOUND
    THEN RAISE_APPLICATION_ERROR(-20004, 'Nu exista piloti cu acest nume');
```

```
END procedura3;  
/
```

```
EXECUTE procedura3('POPESCU');
```

```
-- Nu a facut nicio calatorie, deci va afisa 0  
EXECUTE procedura3('avram');
```

```
-- Exista mai multi piloti cu numele Olteanu  
-- deci programul va arunca o eroare  
EXECUTE procedura3('Olteanu');
```

```
-- Nu exista un pilot cu numele Ionescu deci avem eroare  
EXECUTE procedura3('Ionescu');
```

The screenshot shows the SQL Developer interface with a script named 'PROJECT SG8D.sql' being executed. The script contains several SQL statements and comments. The execution results are displayed in the 'Script Output' window, showing errors for the 'Olteanu' and 'Ionescu' test cases, and successful results for 'Popescu' and 'avram'.

```
995  
996 EXECUTE procedura3('POPESCU');  
997  
998 -- Nu a facut nicio calatorie, deci va afisa 0  
999 EXECUTE procedura3('avram');  
1000  
1001 -- Exista mai multi piloti cu numele Olteanu  
1002 -- deci programul va arunca o eroare  
1003 EXECUTE procedura3('Olteanu');  
1004  
1005 -- Nu exista un pilot cu numele Ionescu deci avem eroare  
1006 EXECUTE procedura3('Ionescu');  
1007
```

Script Output: Task completed in 0.053 seconds

BEGIN procedura3('Olteanu'); END;
Error report -
ORA-20003: Exista mai multi piloti cu acest nume
ORA-06512: at "LAZAROIUTEODORA.PROCEDURA3", line 39
ORA-06512: at line 1

Error starting at line : 1,006 in command -
BEGIN procedura3('Ionescu'); END;
Error report -
ORA-20004: Nu exista piloti cu acest nume
ORA-06512: at "LAZAROIUTEODORA.PROCEDURA3", line 41
ORA-06512: at line 1

PROJECT SG8D

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

Nume pilot: Avram
Numar de calatorii: 0
Numele aeroportului: Mihail Kogalniceanu International Airpor
Orasul aeroportului in care lucreaza: Constanta
Regiunea in care se afla aeroportul: Dobrogea

10 Trigger de tip LMD la nivel de comandă

```
-- Declansator care interzice stergerea  
-- datelor din tabelul REZERVA
```

```
/
```

```
CREATE OR REPLACE TRIGGER trigger1  
BEFORE DELETE ON REZERVA
```

```
BEGIN
```

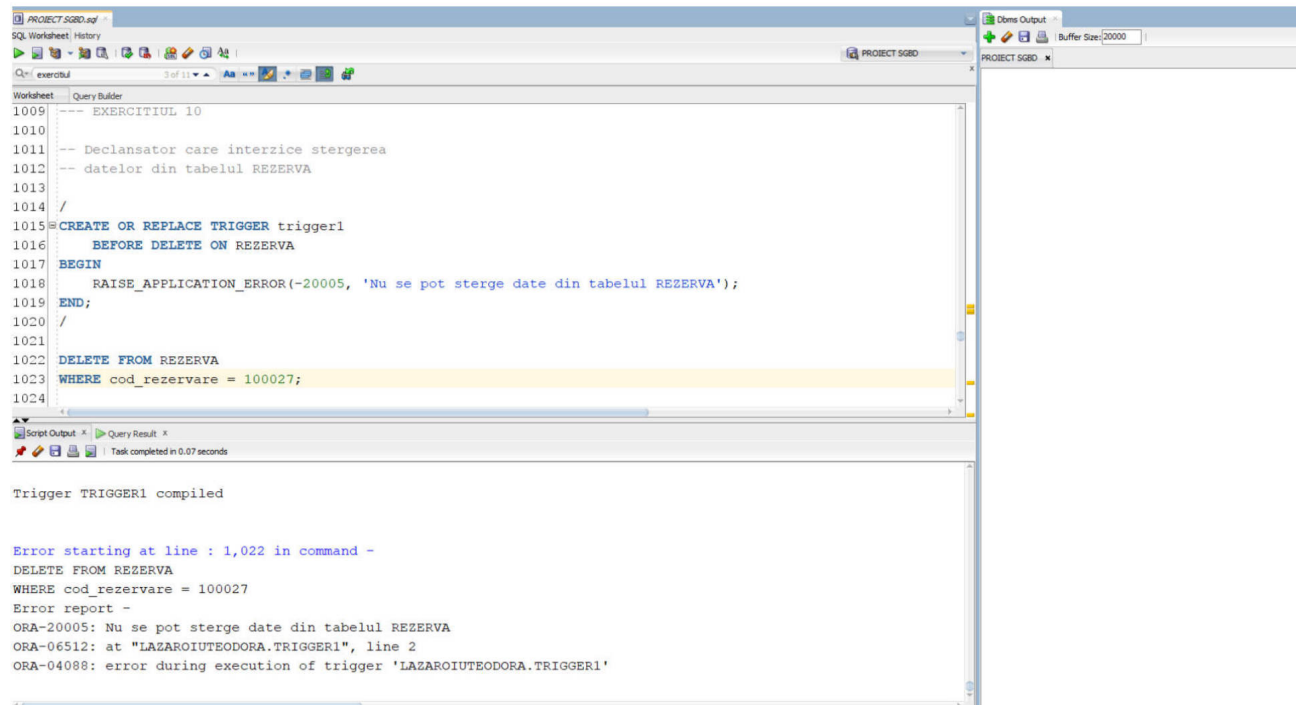
```
RAISE_APPLICATION_ERROR(-20005, 'Nu se pot sterge date din tabelul REZERVA');
```

```
END;
```

```
/
```

```
DELETE FROM REZERVA
```

```
WHERE cod_rezervare = 100027;
```

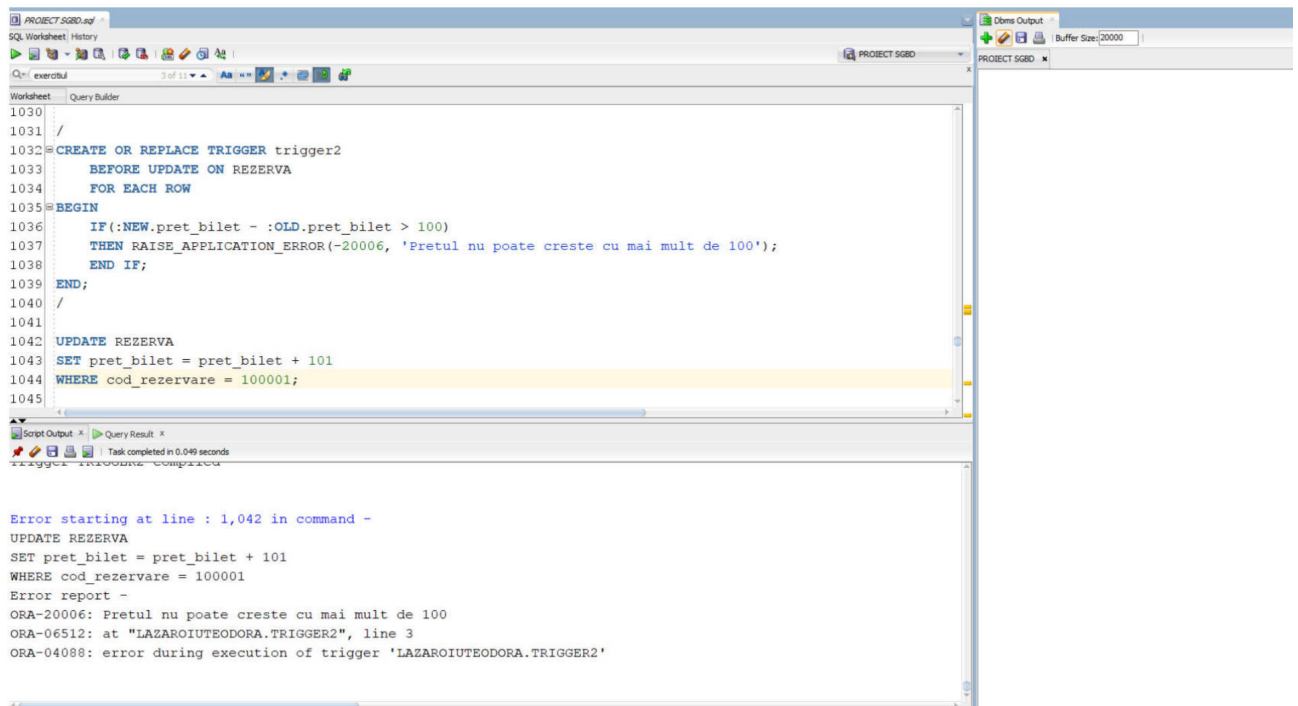


11 Trigger de tip LMD la nivel de linie

```
-- Declansator care interzice cresterea
-- pretului unui bilet cu mai mult de 100

/
CREATE OR REPLACE TRIGGER trigger2
  BEFORE UPDATE ON REZERVA
  FOR EACH ROW
BEGIN
  IF(:NEW.pret_bilet - :OLD.pret_bilet > 100)
  THEN RAISE_APPLICATION_ERROR(-20006, 'Pretul nu poate creste cu mai mult de 100');
  END IF;
END;
/

UPDATE REZERVA
SET pret_bilet = pret_bilet + 101
WHERE cod_rezervare = 100001;
```



12 Trigger de tip LDD

```
-- Declansator care afiseaza un mesaj de fiecare  
-- data cand este rulata o comanda LDD
```

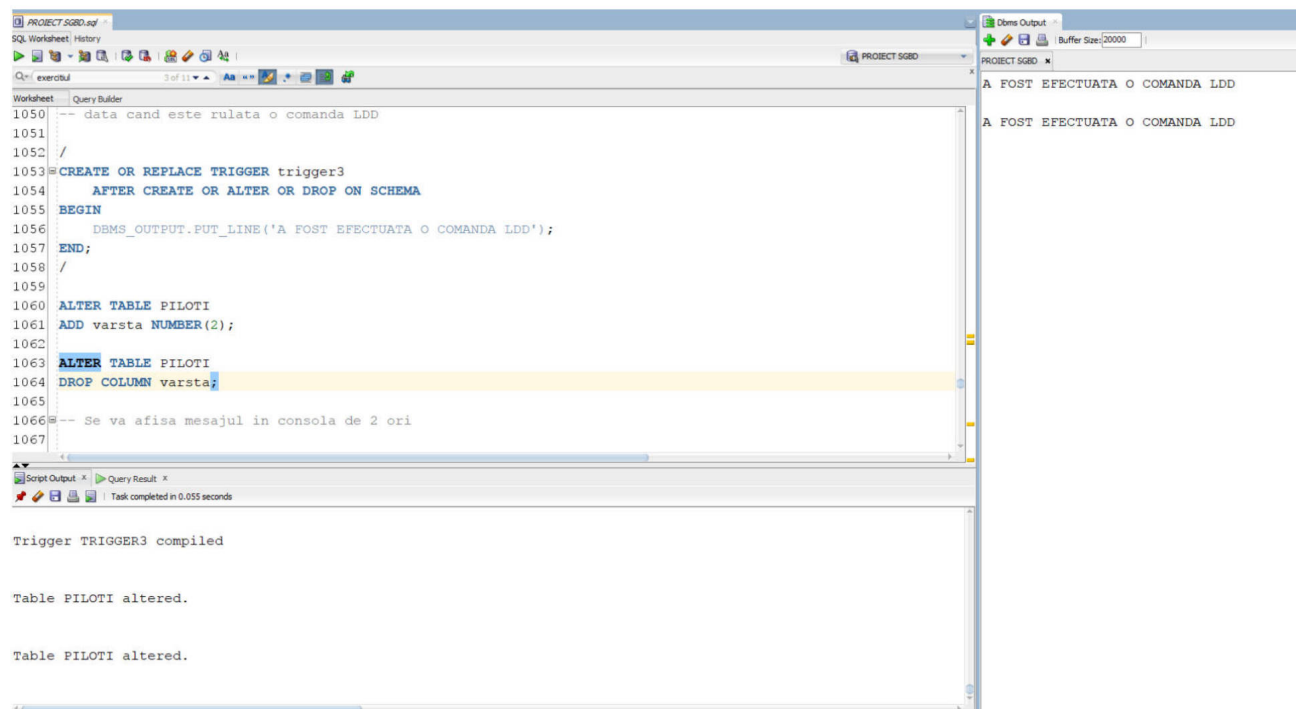
```
/
```

```
CREATE OR REPLACE TRIGGER trigger3  
    AFTER CREATE OR ALTER OR DROP ON SCHEMA  
BEGIN  
    DBMS_OUTPUT.PUT_LINE('A FOST EFECTUATA O COMANDA LDD');  
END;  
/
```

```
ALTER TABLE PILOTI  
ADD varsta NUMBER(2);
```

```
ALTER TABLE PILOTI  
DROP COLUMN varsta;
```

```
-- Se va afisa mesajul in consola de 2 ori
```



13 Pachet cu obiectele din proiect

```
/
CREATE OR REPLACE PACKAGE pachet1 AS
    FUNCTION functie1(v_nume PASAGERI.nume%TYPE) RETURN REZERVA.pret_bilet%TYPE;
    PROCEDURE procedura1;
    PROCEDURE procedura2;
    PROCEDURE procedura3(ume_pilot PILOTI.nume%TYPE);
END pachet1;
/
CREATE OR REPLACE PACKAGE BODY pachet1 AS

-- Pretul total pentru rezervarile facute de un pasager
    FUNCTION functie1(v_nume PASAGERI.nume%TYPE)
    RETURN REZERVA.pret_bilet%TYPE IS
        pret_total REZERVA.pret_bilet%TYPE;
        contor NUMBER;
    BEGIN
        SELECT COUNT(*) INTO contor
        FROM PASAGERI WHERE INITCAP(v_nume) = INITCAP(ume);

        IF contor = 0
        THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista un pasager cu acest nume');
        ELSIF contor > 1
        THEN RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi pasageri cu acest nume');
        END IF;

        SELECT COUNT(*) INTO contor
        FROM PASAGERI RIGHT OUTER JOIN REZERVA USING (id_pasager)
        WHERE INITCAP(v_nume) = INITCAP(ume);

        IF contor = 0
        THEN RAISE_APPLICATION_ERROR(-20002, 'Nu exista rezervari pe acest nume');
        END IF;

        SELECT SUM(pret_bilet + NVL(pret_aditional, 0))
        INTO pret_total
        FROM REZERVA JOIN CLASE USING (cod_clasa) JOIN PASAGERI USING (id_pasager)
        WHERE INITCAP(v_nume) = INITCAP(PASAGERI.nume);

        RETURN pret_total;
    END functie1;

-- Salariul mediu al pilotilor pentru fiecare aeroport
    PROCEDURE procedura1 IS
        TYPE aeroport_record IS RECORD
            (cod AEROPORTURI.cod_iata%TYPE,
             nume AEROPORTURI.nume%TYPE,
```



```

    salariu_mediu PILOTI.salariu%TYPE);

TYPE tablou_imbricat IS TABLE OF aeroport_record;
TYPE vector IS VARRAY(8) OF PILOTI.salariu%TYPE;
CURSOR c IS
    SELECT cod_iata
    FROM AEROPORTURI;

t tablou_imbricat := tablou_imbricat();
r aeroport_record;
v vector := vector();
BEGIN
    FOR i IN c LOOP
        SELECT cod_iata, nume
        INTO r.cod, r.nume
        FROM AEROPORTURI
        WHERE i.cod_iata = cod_iata;

        v.EXTEND;

        SELECT AVG(salariu)
        INTO v(v.LAST)
        FROM PILOTI
        WHERE cod_aeroport = i.cod_iata;

        r.salariu_mediu := v(v.LAST);
        t.EXTEND;
        t(t.LAST) := r;
    END LOOP;
END procedura1;

-- Mareste cu 10% salariul insotitorilor de zbor
-- angajati incepand cu anul 2018
PROCEDURE procedura2 IS
    CURSOR c IS
        SELECT * FROM INSOTITORI
        WHERE TO_CHAR(data_angajarii, 'YYYY') >= 2018
        FOR UPDATE OF salariu NOWAIT;
BEGIN
    FOR i IN c LOOP
        UPDATE INSOTITORI
        SET salariu = salariu + (salariu * 10)/100
        WHERE CURRENT OF c;
    END LOOP;
END procedura2;

-- Detalii despre un pilot dat ca parametru
PROCEDURE procedura3 (nume_pilot PILOTI.nume%TYPE) IS

```

```

    numar_calatorii NUMBER(4);
    nume_aeroport AEROPORTURI.nume%TYPE;
    nume_oras ORASE.nume%TYPE;
    nume_regiune REGIUNI.nume%TYPE;
BEGIN
    SELECT COUNT(cod_calatorie), a.nume, o.nume, r.nume
    INTO numar_calatorii, nume_aeroport, nume_oras, nume_regiune
    FROM CALATORII c RIGHT JOIN PILOTI p ON (id_pilot = id_pilot_principal OR
    id_pilot = id_pilot_secundar) JOIN AEROPORTURI a ON (p.cod_aeroport = cod_iata)
    JOIN ORASE o ON (o.cod_aeroport = cod_iata) JOIN REGIUNI r USING (cod_regiune)
    WHERE INITCAP(nume_pilot) = INITCAP(p.nume)
    GROUP BY a.nume, o.nume, r.nume;

    DBMS_OUTPUT.PUT_LINE('Nume pilot: ' || INITCAP(nume_pilot));
    DBMS_OUTPUT.PUT_LINE('Numar de calatorii: ' || numar_calatorii);
    DBMS_OUTPUT.PUT_LINE('Numele aeroportului: ' || nume_aeroport);
    DBMS_OUTPUT.PUT_LINE('Orasul aeroportului in care lucreaza: ' || nume_oras);
    DBMS_OUTPUT.PUT_LINE('Regiunea in care se afla aeroportul: ' || nume_regiune);
EXCEPTION
    WHEN TOO_MANY_ROWS
    THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi piloti cu acest nume');
    WHEN NO_DATA_FOUND
    THEN RAISE_APPLICATION_ERROR(-20004, 'Nu exista piloti cu acest nume');
END procedura3;

END pachet1;
/

-- Pretul total pentru rezervarile facute de un pasager
SELECT pachet1.functie1('IONITA') FROM DUAL;

-- Salariul mediu al pilotilor pentru fiecare aeroport
EXECUTE pachet1.procedura1;

-- Mareste cu 10% salariul insotitorilor de zbor
-- angajati incepand cu anul 2018
EXECUTE pachet1.procedura2;

-- Detalii despre un pilot
EXECUTE pachet1.procedura3('POPESCU');

```

PROJECT_SGBD.sql

SQL Worksheet: History

Q= exercitulu 3 of 11

Worksheet Query Builder

```

1185 THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi piloti cu acest nume');
1186 WHEN NO_DATA_FOUND
1187 THEN RAISE_APPLICATION_ERROR(-20004, 'Nu exista piloti cu acest nume');
1188 END procedura3;
1189
1190 END pachet1;
1191 /
1192
1193 -- Pretul total pentru rezervarile facute de un pasager
1194 SELECT pachet1.functie1('IONITA') FROM DUAL;
1195
1196 -- Salariul mediu al pilotilor pentru fiecare aeroport
1197 EXECUTE pachet1.procedural;
1198
1199 -- Mareste cu 10% salariul insotitorilor de zbor
1200 -- angajati incepand cu anul 2018
1201 EXECUTE pachet1.procedura2;
1202

```

Script Output x Query Result x

SQL All Rows Fetched: 1 in 0.002 seconds

PACHET1.FUNCTIE1(IONITA)
550

Dbms Output x Buffer Size:20000

PROJECT_SGBD

PROJECT_SGBD.sql

SQL Worksheet: History

Q= PROCEDURA1 3 of 11

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.functie1('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 x Query Result x

SQL Task completed in 0.056 seconds

Package Body PACHET1 compiled

PL/SQL procedure successfully completed.

Dbms Output x Buffer Size:20000

PROJECT_SGBD

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_SGBD.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.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

```

Script Output: Query Result: x

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

PROJECT_SGBD.sql

SQL Worksheet: History

Q= PROCEDURA1

Worksheet: Query Builder

```

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: Query Result: x

Task completed in 0.063 seconds

Rollback complete.

PL/SQL procedure successfully completed.

Dbms Output: PROJECT_SGBD

Buffer Size: 20000

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

14 Pachet cu flux de acțiuni integrate

```
-- Se creeaza un pachet ce contine un tabel imbricat
-- cu o coloana de tip inregistrare ce vor pastra date
-- despre angajatii aeroporturilor (piloti si insotitori)
```

```
-- Vor fi apelate 2 proceduri pentru a se introduce
-- datele in tabelul imbricat si se vor folosi 2 functii
-- pentru a calcula salariul mediul al angajatilor si
-- numarul de angajati care au salariul peste medie.
-- A doua functie se va folosi de prima.
```

```
/
```

```
CREATE OR REPLACE PACKAGE pachet2 AS
```

```
    TYPE angajati_record IS RECORD
        (nume PILOTI.nume%TYPE,
         prenume PILOTI.prenume%TYPE,
         cod_aeroport PILOTI.cod_aeroport%TYPE,
         salariu PILOTI.salariu%TYPE,
         titlu_job VARCHAR2(20));
```

```
    TYPE tablou_angajati IS TABLE OF angajati_record;
```

```
    t tablou_angajati := tablou_angajati();
```

```
    -- introduce date in tablou din tabelul PILOTI
```

```
    PROCEDURE procedura1_pachet;
```

```
    -- introduce date in tablou din tabelul INSOTITORI
```

```
    PROCEDURE procedura2_pachet;
```

```
    -- returneaza media salariilor angajatilor
```

```
    FUNCTION functie1_pachet RETURN NUMBER;
```

```
    -- returneaza numarul de angajati
```

```
    -- care au salariul peste medie
```

```
    FUNCTION functie2_pachet RETURN NUMBER;
```

```
END pachet2;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY pachet2 AS
```

```
    -- preluarea datelor din tabelul PILOTI
```

```
    -- si introducerea lor in tablou
```

```
    PROCEDURE procedura1_pachet IS
```

```
        r angajati_record;
```

```
        CURSOR c IS
```

```
            SELECT id_pilot FROM PILOTI;
```

```
    BEGIN
```

```
        FOR i IN c LOOP
```

```
            SELECT nume, prenume, cod_aeroport, salariu, 'pilot'
```

```

        INTO r.num, r.prenume, r.cod_aeroport, r.salariu, r.titlu_job
        FROM PILOTI WHERE id_pilot = i.id_pilot;

        t.EXTEND;
        t(t.LAST) := r;
    END LOOP;
END procedura1_pachet;

-- preluarea datelor din tabelul INSOTITORI
-- si introducerea lor in tablou
PROCEDURE procedura2_pachet IS
    r angajati_record;
    CURSOR c IS
        SELECT id_insotitor_de_zbor FROM INSOTITORI;
BEGIN
    FOR i IN c LOOP
        SELECT num, prenume, cod_aeroport, salariu, 'insotitor de zbor'
        INTO r.num, r.prenume, r.cod_aeroport, r.salariu, r.titlu_job
        FROM INSOTITORI WHERE id_insotitor_de_zbor = i.id_insotitor_de_zbor;

        t.EXTEND;
        t(t.LAST) := r;
    END LOOP;
END procedura2_pachet;

-- functie ce calculeaza media salariilor angajatilor
FUNCTION functie1_pachet RETURN NUMBER IS
    salariu NUMBER := 0;
BEGIN
    FOR i IN t.FIRST..t.LAST LOOP
        salariu := salariu + t(i).salariu;
    END LOOP;

    salariu := ROUND(salariu / t.COUNT);

    RETURN salariu;
END functie1_pachet;

-- functie ce calculeaza numarul de angajati
-- care au salariul mai mare decat media salariilor
FUNCTION functie2_pachet RETURN NUMBER IS
    contor NUMBER := 0;
BEGIN
    FOR i IN t.FIRST..t.LAST LOOP
        IF (t(i).salariu > functie1_pachet)
        THEN contor := contor + 1;
        END IF;
    END LOOP;
END functie2_pachet;

```

```
        RETURN contor;  
    END functie2_pachet;
```

```
END pachet2;  
/
```

```
-- Se vor rula cele 3 comenzi de mai jos in aceasta ordine  
EXECUTE pachet2.procedura1_pachet;  
EXECUTE pachet2.procedura2_pachet;
```

```
-- Numarul de angajati cu salariul peste medie  
SELECT pachet2.functie2_pachet FROM dual;
```

```
-- Pentru resetarea valorilor din tablou se  
-- va da DROP la pachet si se va recompila  
DROP PACKAGE pachet2;
```

