

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

Proiect Baze de Date
Gestiunea datelor unei Linii Aeriene

Profesor coordonator

Vlad Diaconita

Student

Neagu Chris-Marios

Grupa 1055, seria

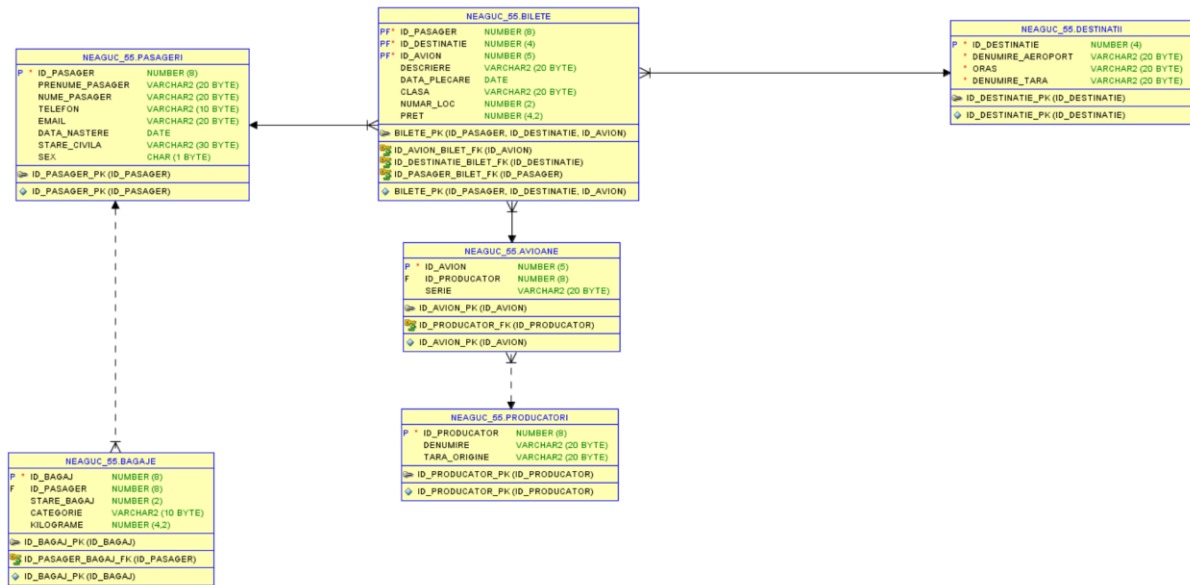
Cuprins

1. Descrierea temei.....	3
2. Schema conceptuala pentru modelarea temei alese.....	4
3. Construirea bazei de date	5
4. Exemple cu operații de actualizare a datelor	8
5. Exemple de interogări cât mai variate și relevante pentru tema aleasă	10

1. Descrierea temei

Am construit o baza de date pentru o linie aeriana. Compania efectueaza transport aerian, iar pasagerii pot alege sa calatoreasca la clasa I, business sau econom. In functie de clasa si de dimensiunea bagajului (daca categoria este de cala si bagajul depaseste 10kg spre exemplu se va plati in plus, iar bagajele de mana trebuie sa fie de maxim 10kg pentru a nu ingreuna avionul) se va stabili valoarea finala a biletului. Astfel, se asigura buna gestiune a calatoriilor, cat si bunastarea pasagerilor aflatii la bordul avionului.

2. Schema conceptuala pentru modelarea temei alese.



3. Construirea bazei de date

1. Instrucțiuni pentru crearea tabelelor – CREATE TABLE sa se creeze tabelele dupa schema conceptuala

```
CREATE TABLE PASAGERI(
```

```
    ID_PASAGER NUMBER(8) CONSTRAINT ID_PASAGER_PK PRIMARY KEY,
```

```
    PRENUME_PASAGER VARCHAR2(20),
```

```
    NUME_PASAGER VARCHAR2(20),
```

```
    TELEFON VARCHAR2(10),
```

```
    EMAIL VARCHAR2(20),
```

```
    DATA_NASTERE DATE,
```

```
    STARE_CIVILA VARCHAR2(30),
```

```
    SEX CHAR,
```

```
    CONSTRAINT TELEFON_CK CHECK (TELEFON not like '%[^0-9]%'),
```

```
    CONSTRAINT EMAIL_CK CHECK (REGEXP_LIKE(EMAIL, '.*@.+.+.+')),
```

```
    CONSTRAINT STARE_CIVILA_CK CHECK(STARE_CIVILA IN  
( 'casatorit', 'singur', 'divortat' )),
```

```
    CONSTRAINT SEX_CK CHECK(SEX IN ('M','F'))
```

```
);
```

```
CREATE TABLE BAGAJE(
```

```
    ID_BAGAJ NUMBER(8) CONSTRAINT ID_BAGAJ_PK PRIMARY KEY,
```

```
    ID_PASAGER NUMBER(8),
```

```
    STARE_BAGAJ NUMBER(2) CONSTRAINT STARE_BAGAJ_CK  
CHECK(STARE_BAGAJ IN ('0','1','2')),
```

```
    CATEGORIE VARCHAR2(10),
```

```
    KILOGRAME NUMBER(4,2),
```

```
    CONSTRAINT ID_PASAGER_BAGAJ_FK FOREIGN KEY(ID_PASAGER)  
REFERENCES PASAGERI(ID_PASAGER)
```

```
);
```

```
CREATE TABLE PRODUCATORI(
```

```
    ID_PRODUCATOR NUMBER(8) CONSTRAINT ID_PRODUCATOR_PK PRIMARY KEY,
```

```
    DENUMIRE VARCHAR2(20),
```

```
    TARA_ORIGINE VARCHAR2(20)
```

```
);
```

```

CREATE TABLE AVIOANE(
ID_AVION NUMBER(5) CONSTRAINT ID_AVION_PK PRIMARY KEY,
ID_PRODUCATOR NUMBER(8),
SERIE VARCHAR2(20),
CONSTRAINT ID_PRODUCATOR_FK FOREIGN KEY(ID_PRODUCATOR) REFERENCES
PRODUCATORI(ID_PRODUCATOR)
);

CREATE TABLE DESTINATII(
ID_DESTINATIE NUMBER(4) CONSTRAINT ID_DESTINATIE_PK PRIMARY KEY,
DENUMIRE_AEROPORT VARCHAR2(20) NOT NULL,
ORAS VARCHAR2(20) NOT NULL,
DENUMIRE_TARA VARCHAR2(20) NOT NULL
);

CREATE TABLE BILETE(
ID_PASAGER NUMBER(8),
ID_DESTINATIE NUMBER(4),
ID_AVION NUMBER(5),
DESCRIERE VARCHAR2(20),
DATA_PLECARE DATE,
CLASA VARCHAR2(20) CONSTRAINT CLASA_CK CHECK(CLASA IN
('I','Business','Econom')),
NUMAR_LOC NUMBER(2),
PRET NUMBER(4,2) CONSTRAINT PRET_CK CHECK(PRET >= 0),
CONSTRAINT ID_PASAGER_BILET_FK FOREIGN KEY(ID_PASAGER) REFERENCES
PASAGERI(ID_PASAGER),
CONSTRAINT ID_DESTINATIE_BILET_FK FOREIGN KEY(ID_DESTINATIE)
REFERENCES DESTINATII(ID_DESTINATIE),
CONSTRAINT ID_AVION_BILET_FK FOREIGN KEY(ID_AVION) REFERENCES
AVIOANE(ID_AVION),
CONSTRAINT BILETE_PK PRIMARY KEY(ID_PASAGER, ID_DESTINATIE, ID_AVION)
);

```

2. Instructiuni pentru stergerea tabelelor – sa se stearga toate tabelele

```
DROP TABLE BAGAJE CASCADE CONSTRAINTS;  
DROP TABLE PASAGERI CASCADE CONSTRAINTS;  
DROP TABLE BILETE CASCADE CONSTRAINTS;  
DROP TABLE AVIOANE CASCADE CONSTRAINTS;  
DROP TABLE PRODUCATORI CASCADE CONSTRAINTS;  
DROP TABLE DESTINATII CASCADE CONSTRAINTS;
```

Sa se restaureze tabelele sterse folosind comanda **FLASHBACK TABLE**.

```
FLASHBACK TABLE BAGAJE TO BEFORE DROP;  
FLASHBACK TABLE PASAGERI TO BEFORE DROP;  
FLASHBACK TABLE BILETE TO BEFORE DROP;  
FLASHBACK TABLE AVIOANE TO BEFORE DROP;  
FLASHBACK TABLE PRODUCATORI TO BEFORE DROP;  
FLASHBACK TABLE DESTINATII TO BEFORE DROP;
```

3. Comanda Alter

Sa se adauge o restrictie check care sa valideze categoria bagajelor ca fiind 'DE MANA' sau 'DE CALA'

```
ALTER TABLE BAGAJE ADD CONSTRAINT CATEGORIE_CK CHECK(CATEGORIE  
IN('DE MANA','DE CALA'));
```

Sa se adauge o restrictie check care sa verifice daca bagajele de mana nu depasesc 10 kilograme sau daca sunt de cala sa nu depaseasca 50 de kilograme pentru a nu ingreuna avionul.

```
ALTER TABLE BAGAJE ADD CONSTRAINT KILOGRAME_CK  
CHECK((CATEGORIE='DE MANA' AND KILOGRAME<=10)OR(CATEGORIE='DE CALA'  
AND KILOGRAME<=50));
```

Sa se modifice tabelul bilete astfel incat coloana **DATA_PLECARE** sa fie nenula si default sa fie ziua curenta.

```
ALTER TABLE BILETE MODIFY DATA_PLECARE DATE DEFAULT SYSDATE NOT  
NULL;
```

Sa se adauge o restrictie not null pe coloana **NUME_PASAGER** din tabelul **PASAGERI** in 2 moduri diferite.

```
ALTER TABLE PASAGERI MODIFY NUME_PASAGER NOT NULL;
```

```
ALTER TABLE PASAGERI ADD CONSTRAINT NUME_PASAGER_NN  
CHECK(NUME_PASAGER IS NOT NULL);
```

Sa se modifice capacitatea emailului din tabelul **PASAGERI** astfel incat sa poata retine 30 de caractere.

```
ALTER TABLE PASAGERI MODIFY EMAIL VARCHAR2(30);
```

Sa se adauge o restrictie not null pe coloana **DENUMIRE** din tabelul **PRODUCATORI**.

```
ALTER TABLE PRODUCATORI MODIFY DENUMIRE NOT NULL;
```

Sa se modifice capacitatea coloanei PRET din tabelul BILETE astfel incat sa permita retinerea datelor de tip number de maxim 8 cifre, cu precizie 2.

```
ALTER TABLE BILETE MODIFY PRET NUMBER(8,2);
```

Sa se mareasca capacitatea coloanei descriere din tabelul BILETE

```
ALTER TABLE BILETE MODIFY DESCRIERE VARCHAR2(100);
```

4. Exemple cu operații de actualizare a datelor

1. Popularea bazei de date cu comanda insert – sa se insereze minim 2 randuri in fiecare tabel

```
INSERT INTO  
PASAGERI(ID_PASAGER,NUME_PASAGER,PRENUME_PASAGER,TELEFON,EMAIL,DATA_NASTERE,STARE_CIVILA,SEX)
```

```
VALUES (1,'NEAGU','CHRIS-  
MARIOS','0757232710','chris.neagu@outlook.com',TO_DATE('22-11-2001','DD-MM-  
YYYY'),'singur','M');
```

```
INSERT INTO  
PASAGERI(ID_PASAGER,NUME_PASAGER,PRENUME_PASAGER,TELEFON,EMAIL,DATA_NASTERE,STARE_CIVILA,SEX)
```

```
VALUES (2,'ANTON','MARIA','0734958232','anton.mihai@gmail.com',TO_DATE('30-10-  
1973','DD-MM-YYYY'),'casatorit','F');
```

```
INSERT INTO BAGAJE VALUES (1,1,0,'DE MANA',2);
```

```
INSERT INTO BAGAJE VALUES (2,2,1,'DE CALA',40);
```

```
INSERT INTO PRODUCATORI VALUES (1,'BOEING','USA');
```

```
INSERT INTO PRODUCATORI VALUES (2,'AIRBUS','FRANTA');
```

```
INSERT INTO AVIOANE VALUES(1,1,'BOEING1');
```

```
INSERT INTO AVIOANE VALUES(2,2,'AIRBUS2');
```

```
INSERT INTO DESTINATII VALUES(1,'HEATHROW','LONDRA','REGATUL UNIT');
```

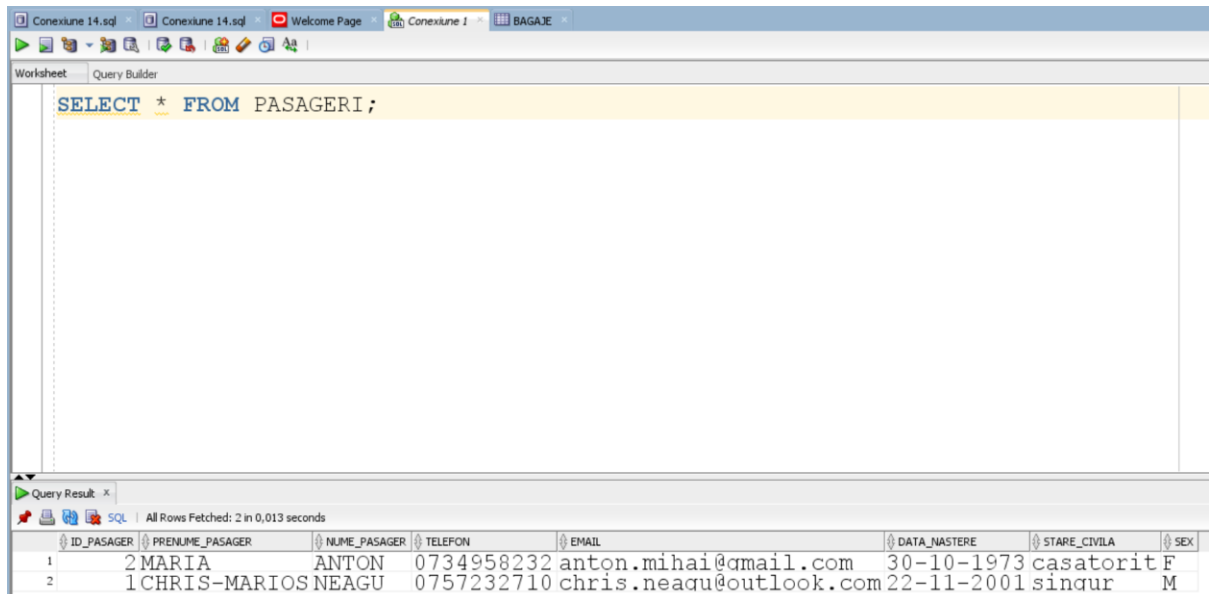
```
INSERT INTO DESTINATII VALUES(2,'OTOPENI','BUCURESTI','ROMANIA');
```

```
INSERT INTO BILETE VALUES(1,1,1,'CALATORIE BUCURESTI-LONDRA LOC NR  
24',TO_DATE('04-01-2022','DD-MM-YYYY'),'T',1,1000);
```

```
INSERT INTO BILETE VALUES(2,2,2,'CALATORIE RETUR PARIS-BUCURESTI LOC  
NR 14',TO_DATE('02-01-2022','DD-MM-YYYY'),'Business',1,600);
```

```
COMMIT;
```


O înregistrare (rând) cu numele studentului după interogarea bazei de date. – SELECT * FROM PASAGERI;



The screenshot shows a database query tool interface. The top part displays the SQL query: `SELECT * FROM PASAGERI;`. Below the query, the results are shown in a table format. The table has 8 columns: ID_PASAGER, PRENUME_PASAGER, NUME_PASAGER, TELEFON, EMAIL, DATA_NASTERE, STARE_CIVILA, and SEX. There are 2 rows of data.

ID_PASAGER	PRENUME_PASAGER	NUME_PASAGER	TELEFON	EMAIL	DATA_NASTERE	STARE_CIVILA	SEX
1	MARIA	ANTON	0734958232	anton.mihai@gmail.com	30-10-1973	casatorit	F
2	CHRIS-MARIOS	NEAGU	0757232710	chris.neagu@outlook.com	22-11-2001	singur	M

Sa se modifice seria avionului identificat prin BOEING1, astfel incat sa fie prescurtata.

```
UPDATE AVIOANE SET SERIE='BO1'
```

```
WHERE SERIE='BOEING1';
```

Sa se modifice starea bagajelor preluate, astfel incat sa apara ca fiind in avion. (stare 0 -> stare 1)

```
UPDATE BAGAJE SET STARE_BAGAJ=1
```

```
WHERE STARE_BAGAJ=0;
```

Sa se modifice clasa inscrisa pe bilete, astfel incat cele Business sa devina Econom.

```
UPDATE BILETE SET CLASA='Econom'
```

```
WHERE CLASA='Business';
```

Sa se modifice costul tuturor biletelor de la clasa Econom, astfel incat sa fie 300.

```
UPDATE BILETE SET PRET=300
```

```
WHERE CLASA='Econom';
```

Sa se modifice aeroportul 'OTOPENI', astfel incat sa devina 'HENRI COANDA'

```
UPDATE DESTINATII SET DENUMIRE_AEROPORT='HENRI COANDA'
```

```
WHERE ORAS='BUCURESTI';
```

Sa se modifice tara de origine astfel incat forma prescurtata USA sa devina STATELE UNITE.

```
UPDATE PRODUCATORI SET TARA_ORIGINE='STATELE UNITE'
WHERE TARA_ORIGINE='USA';
```

Sa se modifice numele pasagerului de la ID_PASAGER 2, astfel incat sa devina 'COMAN'.

```
UPDATE PASAGERI SET NUME_PASAGER = 'COMAN'
WHERE ID_PASAGER = 2;
```

Sa se modifice descrierea cursei pentru a se conforma cu schimbarile de la aeroporturi.

```
UPDATE CURSE SET DESCRIERE='CALATORIE BUCURESTI-LONDRA LOC NR 24
PLECARE DIN AEROPORTUL HENRI COANDA'
WHERE ID_CURSA=1;
```

Sa se utilizeze comanda MERGE pentru a putea modifica costul biletelor pasagerilor care au bagaje de cala si depasesc 10 kilograme. (fiecare kilogram in plus va fi taxat cu 10 unitati monetare)

```
MERGE INTO BILETE USING BAGAJE
ON (BILETE.ID_PASAGER=BAGAJE.ID_PASAGER)
WHEN MATCHED THEN
UPDATE SET BILETE.PRET=BILETE.PRET+(BAGAJE.KILOGRAME-10)*10
WHERE BAGAJE.CATEGORIE='DE CALA';
```

5. Exemple de interogări cât mai variate și relevante pentru tema aleasă

Sa se afiseze toate destinatiile ale caror aeroporturi incep in denumire cu H.

```
SELECT * FROM DESTINATII
WHERE DENUMIRE_AEROPORT LIKE 'H%';
```

Sa se afiseze biletele care au costul intre 1000 si 2000 de unitati monetare.

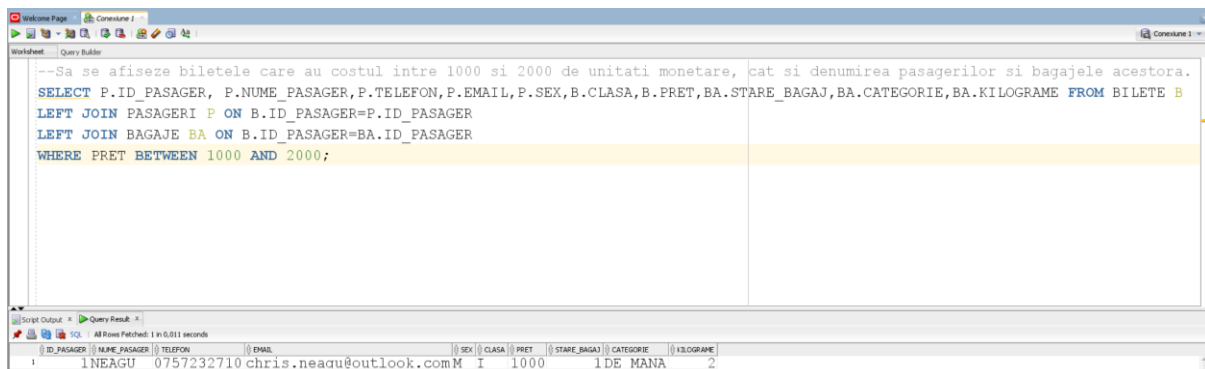
```
SELECT * FROM BILETE
WHERE PRET BETWEEN 1000 AND 2000;
```

Sa se afiseze bagajele si starea lor in format text.

```
SELECT  
ID_BAGAJ,ID_PASAGER,DECODE(STARE_BAGAJ,0,'PRELUAT',1,'IN  
AVION',2,'RETURNAT') STARE_BAGAJ,CATEGORIE, KILOGRAME  
FROM BAGAJE;
```

Sa se afiseze biletele care au costul intre 1000 si 2000 de unitati monetare, cat si denumirea pasagerilor si bagajele acestora.

```
SELECT P.ID_PASAGER,  
P.NUME_PASAGER,P.TELEFON,P.EMAIL,P.SEX,B.CLASA,B.PRET,BA.S  
TARE_BAGAJ,BA.CATEGORIE,BA.KILOGRAME FROM BILETE B  
LEFT JOIN PASAGERI P ON B.ID_PASAGER=P.ID_PASAGER  
LEFT JOIN BAGAJE BA ON B.ID_PASAGER=BA.ID_PASAGER  
WHERE PRET BETWEEN 1000 AND 2000;
```



The screenshot shows a SQL query editor window with a query and its results. The query is as follows:

```
--Sa se afiseze biletele care au costul intre 1000 si 2000 de unitati monetare, cat si denumirea pasagerilor si bagajele acestora.  
SELECT P.ID_PASAGER, P.NUME_PASAGER, P.TELEFON, P.EMAIL, P.SEX, B.CLASA, B.PRET, BA.STARE_BAGAJ, BA.CATEGORIE, BA.KILOGRAME FROM BILETE B  
LEFT JOIN PASAGERI P ON B.ID_PASAGER=P.ID_PASAGER  
LEFT JOIN BAGAJE BA ON B.ID_PASAGER=BA.ID_PASAGER  
WHERE PRET BETWEEN 1000 AND 2000;
```

The results are displayed in a table with the following columns: ID_PASAGER, NUME_PASAGER, TELEFON, EMAIL, SEX, CLASA, PRET, STARE_BAGAJ, CATEGORIE, KILOGRAME. The results are as follows:

ID_PASAGER	NUME_PASAGER	TELEFON	EMAIL	SEX	CLASA	PRET	STARE_BAGAJ	CATEGORIE	KILOGRAME
1	NEAGU	0757232710	chris.neagu@outlook.com	M	I	1000	1	DE MANA	2

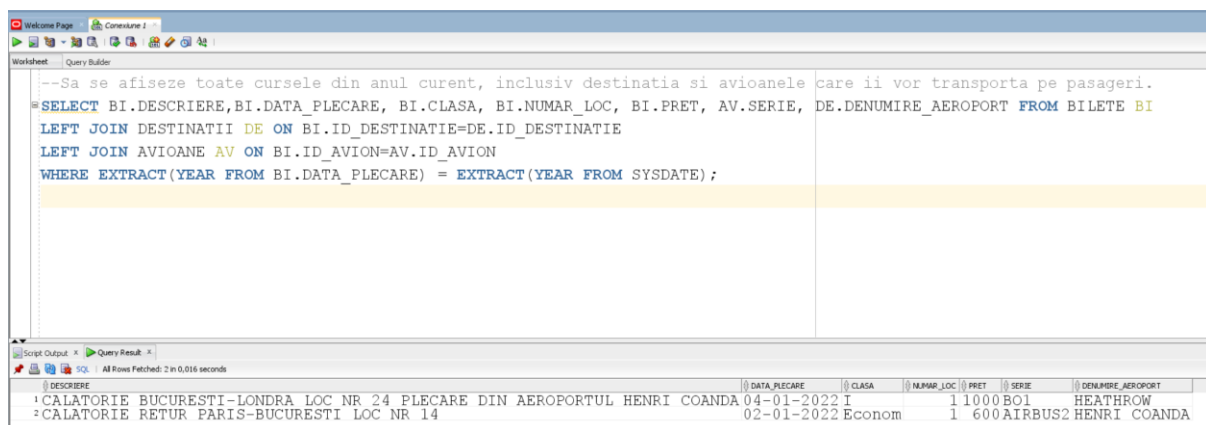
--Sa se afiseze toate cursele din anul curent, inclusiv destinatia si avioanele care ii vor transporta pe pasageri.

```
SELECT BI.DESCRIERE,BI.DATA_PLECARE, BI.CLASA,  
BI.NUMAR_LOC, BI.PRET, AV.SERIE, DE.DENUMIRE_AEROPORT  
FROM BILETE BI
```

```
LEFT JOIN DESTINATII DE ON  
BI.ID_DESTINATIE=DE.ID_DESTINATIE
```

```
LEFT JOIN AVIOANE AV ON BI.ID_AVION=AV.ID_AVION
```

```
WHERE EXTRACT(YEAR FROM BI.DATA_PLECARE) =  
EXTRACT(YEAR FROM SYSDATE);
```



The screenshot shows a SQL query builder window with a query editor on the left and a results pane on the right. The query is as follows:

```
--Sa se afiseze toate cursele din anul curent, inclusiv destinatia si avioanele care ii vor transporta pe pasageri.  
SELECT BI.DESCRIERE,BI.DATA_PLECARE, BI.CLASA, BI.NUMAR_LOC, BI.PRET, AV.SERIE, DE.DENUMIRE_AEROPORT FROM BILETE BI  
LEFT JOIN DESTINATII DE ON BI.ID_DESTINATIE=DE.ID_DESTINATIE  
LEFT JOIN AVIOANE AV ON BI.ID_AVION=AV.ID_AVION  
WHERE EXTRACT(YEAR FROM BI.DATA_PLECARE) = EXTRACT(YEAR FROM SYSDATE);
```

The results pane shows two rows of data:

DESCRIERE	DATA_PLECARE	CLASA	NUMAR_LOC	PRET	SERIE	DENUMIRE_AEROPORT
1 CALATORIE BUCURESTI-LONDRA LOC NR 24 PLECARE DIN AEROPORTUL HENRI COANDA	04-01-2022	1	1000	BO1		HEATHROW
2 CALATORIE RETUR PARIS-BUCURESTI LOC NR 14	02-01-2022	Econom	1	600	AIRBUS2	HENRI COANDA

--Sa se afiseze starea bagajelor in format codificat si decodificat

```
SELECT STARE_BAGAJ,  
CASE WHEN STARE_BAGAJ=0 THEN 'PRELUAT'  
      WHEN STARE_BAGAJ=1 THEN 'IN AVION'  
      ELSE 'RETURNAT'  
END STARE_BAGAJ_CASE  
FROM BAGAJE;
```

--SA SE AFISEZE TOATE AVIOANELE PRODUSE DE PRODUCATORUL BOEING

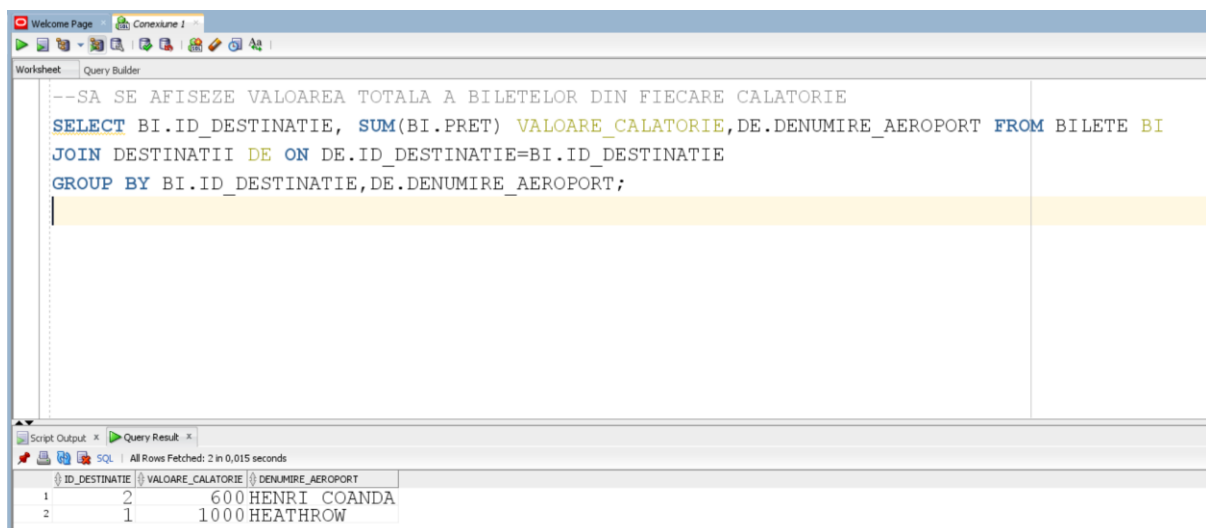
```
SELECT A.ID_AVION,A.SERIE,P.DENUMIRE,P.TARA_ORIGINE FROM  
PRODUCATORI P
```

```
JOIN AVIOANE A ON A.ID_PRODUCATOR=P.ID_PRODUCATOR
```

```
WHERE P.DENUMIRE='BOEING';
```

--SA SE AFISEZE VALOAREA TOTALA A BILETELOR DIN FIECARE CALATORIE

```
SELECT BI.ID_DESTINATIE, SUM(BI.PRET)  
VALOARE_CALATORIE,DE.DENUMIRE_AEROPORT FROM BILETE BI  
JOIN DESTINATII DE ON DE.ID_DESTINATIE=BI.ID_DESTINATIE  
GROUP BY BI.ID_DESTINATIE,DE.DENUMIRE_AEROPORT;
```



The screenshot shows a database query tool interface. The top part displays the SQL query: `--SA SE AFISEZE VALOAREA TOTALA A BILETELOR DIN FIECARE CALATORIE`, `SELECT BI.ID_DESTINATIE, SUM(BI.PRET) VALOARE_CALATORIE,DE.DENUMIRE_AEROPORT FROM BILETE BI`, `JOIN DESTINATII DE ON DE.ID_DESTINATIE=BI.ID_DESTINATIE`, `GROUP BY BI.ID_DESTINATIE,DE.DENUMIRE_AEROPORT;`. The bottom part shows the query results in a table with 2 rows and 3 columns: ID_DESTINATIE, VALOARE_CALATORIE, and DENUMIRE_AEROPORT. The first row has values 2, 600, and HENRI COANDA. The second row has values 1, 1000, and HEATHROW.

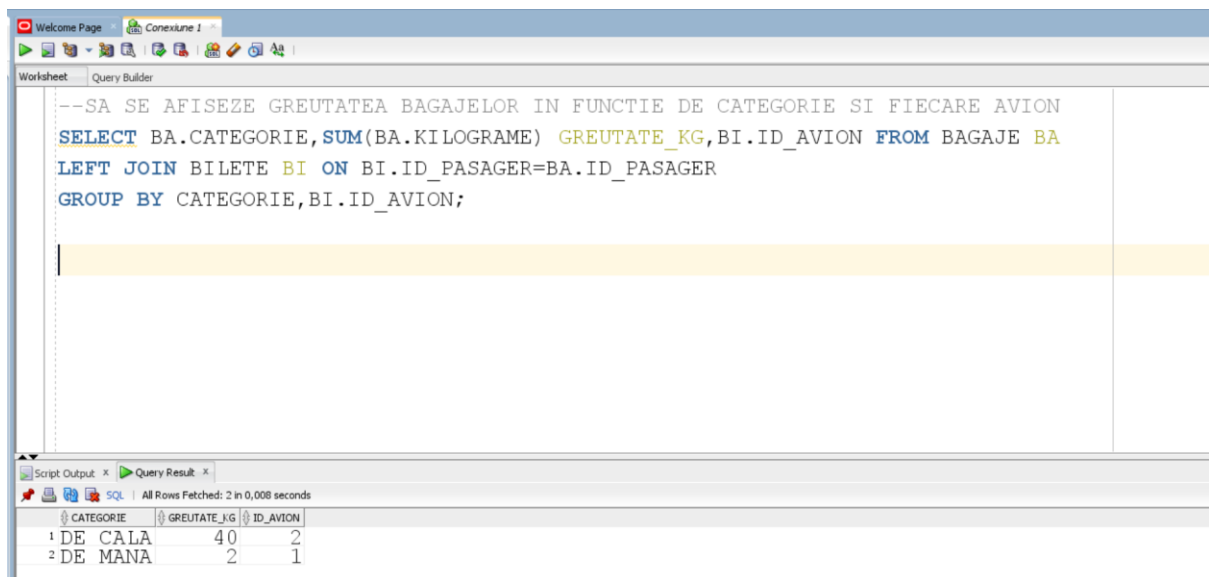
ID_DESTINATIE	VALOARE_CALATORIE	DENUMIRE_AEROPORT
2	600	HENRI COANDA
1	1000	HEATHROW

--sa se afiseze greutatea tuturor bagajelor din fiecare categorie

```
SELECT CATEGORIE,SUM(KILOGRAME) FROM BAGAJE  
GROUP BY CATEGORIE;
```

--SA SE AFISEZE GREUTATEA BAGAJELOR IN FUNCTIE DE CATEGORIE SI FIECARE AVION

```
SELECT BA.CATEGORIE,SUM(BA.KILOGRAME)  
GREUTATE_KG,BI.ID_AVION FROM BAGAJE BA  
LEFT JOIN BILETE BI ON BI.ID_PASAGER=BA.ID_PASAGER  
GROUP BY CATEGORIE,BI.ID_AVION;
```



The screenshot shows a SQL query editor window with a 'Query Builder' tab. The query text is as follows:

```
--SA SE AFISEZE GREUTATEA BAGAJELOR IN FUNCTIE DE CATEGORIE SI FIECARE AVION  
SELECT BA.CATEGORIE,SUM(BA.KILOGRAME) GREUTATE_KG,BI.ID_AVION FROM BAGAJE BA  
LEFT JOIN BILETE BI ON BI.ID_PASAGER=BA.ID_PASAGER  
GROUP BY CATEGORIE,BI.ID_AVION;
```

Below the query editor, the 'Query Result' tab is active, displaying the results of the query. The results are shown in a table with the following columns: CATEGORIE, GREUTATE_KG, and ID_AVION. The table contains two rows of data:

	CATEGORIE	GREUTATE_KG	ID_AVION
1	DE CALA	40	2
2	DE MANA	2	1

--SA SE AFISEZE TOATE DESTINATIILE ALE CAROR 2 LITRE DE INCEPUT SUNT L SI O

```
SELECT * FROM DESTINATII  
WHERE SUBSTR(ORAS,1,2) = 'LO';
```

--SA SE REALIZEZE O TABELA VIRTUALA CU VALOAREA TOTALA A BILETELOR DIN FIECARE CALATORIE

```
CREATE OR REPLACE VIEW VALOARE_BILET_CALATORIE
AS SELECT BI.ID_DESTINATIE, SUM(BI.PRET)
VALOARE_CALATORIE, DE.DENUMIRE_AEROPORT FROM BILETE BI
JOIN DESTINATII DE ON DE.ID_DESTINATIE=BI.ID_DESTINATIE
GROUP BY BI.ID_DESTINATIE, DE.DENUMIRE_AEROPORT;
```

-- SA SE CREEZE UN INDEX PE TABELA PASAGERI PE COLOANA NUME_PASAGER

```
CREATE INDEX INDEX_NUME ON PASAGERI(NUME_PASAGER);
```

Sa se creeze o secventa pentru asigurarea unicitatii cheii primare din tabela BAGAJE.

```
CREATE SEQUENCE seq_nrbagaj
START WITH 3 INCREMENT BY 1
MAXVALUE 10000 NOCYCLE;
```

--Sa se creeze un sinonim pentru tabela BILETE

```
CREATE SYNONYM BI FOR BILETE;
```