Bondoc Ana-Maria

Grupa 141

Proiect Baze de Date

Gestiunea unei companii aeriene



**Cerința 1**

Descrierea modelului real,a utilității acestuia și a regulilor de funcționare.

Companiile aeriene mari utilizează numeroase sisteme software pentru diverse scopuri, cum ar fi planificarea flotei, întreținerea aeronavelor, publicațiile tehnice, planificarea zborurilor, programarea echipajului, planificarea personalului, prețurile, gestionarea veniturilor, rezervările de pasageri, rezervarea online, răspunsul vocal interactiv, managementul relațiilor cu clienții, mărfurile, rezervări, autoservire aeroport, urmărire zboruri, navigare, greutate și centrare, geantă electronică de zbor, contabilitate venituri, registru general, management document etc.

Cele mai multe dintre aceste sisteme sunt furnizate de furnizori externi care utilizează diverse baze de date, inclusiv Oracle, SQL Server și MySQL. Bănuiesc că fiecare companie aeriană importantă din SUA folosește cel puțin o instanță din fiecare sistem major de baze de date, găzduit în sediul propriu sau al furnizorilor lor.

**Cerința 2**

Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului.

Constrângeri:

Un zbor necesită 2 sau mai mulți piloți.

Fiecare pasager are unul sau mai multe bilete.

Fiecare avion îndeplinește mai multe zboruri.

O companie aeriană deține unul sau mai multe avioane.

O companie aeriană realizează mai multe zboruri.

Un bilet implică un boarding pass pentru a putea fi posibilă călătoria.

Id-ul rutei este format din id-ul companiei plus id-ul zborului.

Modelul relațional va impune constrângeri de tip:

- cheie primară (PK)

- cheie străină (FK)

**Cerința 3**

Descrierea entităților, incluzând precizarea cheiiprimare.

|  |  |  |
| --- | --- | --- |
| **ENTITATE** | **CHEIE PRIMARĂ** | **DESCRIERE** |
| Pilot | id\_pilot | Informații privind numele, contact și experiență. |
| Zbor | id\_zbor | Informații specifice despre decolare, aterizare și locul plecării și al sosirii. |
| Boarding Pass | id\_boardingpass | Un boarding pass este un document care permite îmbarcarea pasagerilor în aeronavă. |
| Bilet | id\_bilet | Informații necesare despre zbor. |
| Pasager | id\_pasager | Entitatea conține informații necesare identificării unui pasager. |
| Avion | id\_avion | Informații referitoare la compania căreia îi aparține și modelul. |
| Companie aeriană | id\_companie | Informații privind numele si email-ul. |
| Rută | id\_rută | O rută se referă la locul decolării și locul aterizării. |

**Cerința 4**

Descrierea relațiilor, incluzând precizareacardinalității acestora.

|  |  |  |
| --- | --- | --- |
| **RELAȚIE** | **CARDINALITATE** | **DESCRIERE** |
| Pilot-Zbor | one-to-many  1-M(2) | Un zbor necesită 2 sau mai mulți piloți. |
| Zbor-Boarding Pass | one-to-many  1-M(1) | Un zbor include mai multe boarding pass-uri. |
| Boarding Pass-Pasager | one-to-many  1-M(1) | Un pasager poate deține mai multe boarding pass-uri. |
| Boarding Pass-Bilet | one-to-one  1-1 | Un bilet generează un singur boarding pass, iar un boarding pass poate fi generat pe baza unui singur bilet. |
| Zbor-Avion | one-to-many  1-M(0) | Un avion realizează mai multe zboruri. |
| Comapanie aeriană-Zbor | many-to-many  M(1)-M(1) | O companie aeriană poate efectua mai multe zboruri, iar un zbor poate fi realizat de mai multe companii. |
| Companie aeriană-Avion | one-to-many  1-M(1) | O companie aeriană deține mai multe avioane. |

**Cerința 5**

Descrierea atributelor, incluzând tipul de date și eventualeleconstrângeri, valori implicite, valori posibile ale atributelor.

Pilot

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Tip** | **Descriere** |
| PK id\_pilot | int |  |
| FK id\_zbor | int |  |
| nume | varchar(50) |  |
| prenume | varchar(50) |  |
| nr\_telefon | varchar(10) |  |
| nr\_zboruri | int | Reprezintă experiența pilotului prin numărul de zboruri realizate cu success. |

Zbor

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Tip** | **Descriere** |
| PK id\_zbor | int |  |
| FK id\_avion | int |  |
| data\_decolare | date |  |
| ora\_decolare | varchar(5) |  |
| data\_aterizare | date |  |
| ora\_aterizare | varchar(5) |  |
| plecare | varchar(50) | Orașul din care se face plecarea. |
| sosire | varchar(50) | Orașul cu destinația finală. |

Boarding Pass

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Tip** | **Descriere** |
| PK id\_boardingpass | int |  |
| FK id\_zbor | int |  |
| FK id\_pasager | int |  |
| FK id\_bilet | int |  |
| poartă | int | Poarta la care se face îmbarcarea. |

Bilet

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Tip** | **Descriere** |
| PK id\_bilet | int |  |
| FK id\_pasager | int |  |
| FK id\_zbor | int |  |
| plecare | varchar(50) | Orașul din care se face plecarea. |
| sosire | varchar(50) | Orașul cu destinația finală. |

Pasager

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Tip** | **Descriere** |
| PK id\_pasager | int |  |
| nume | varchar(50) |  |
| prenume | varchar(50) |  |
| nr\_telefon | varchar(10) |  |

Avion

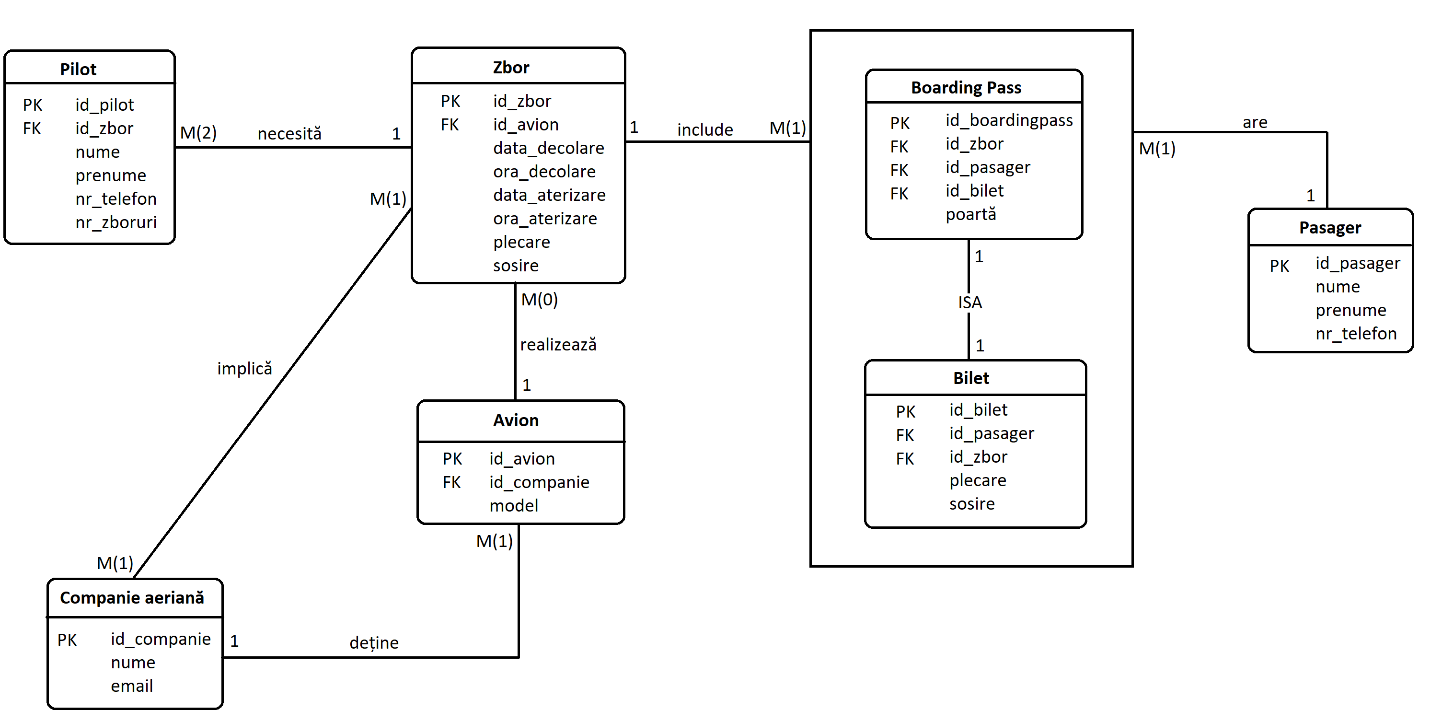
|  |  |  |
| --- | --- | --- |
| **Atribut** | **Tip** | **Descriere** |
| PK id\_avion | int |  |
| FK id\_companie | int |  |
| model | varchar(50) | Modelul avionului. |

Companie aeriană

|  |  |  |
| --- | --- | --- |
| **Atribut** | **Tip** | **Descriere** |
| PK id\_companie | int |  |
| nume | varchar(50) |  |
| email | varchar(50) |  |

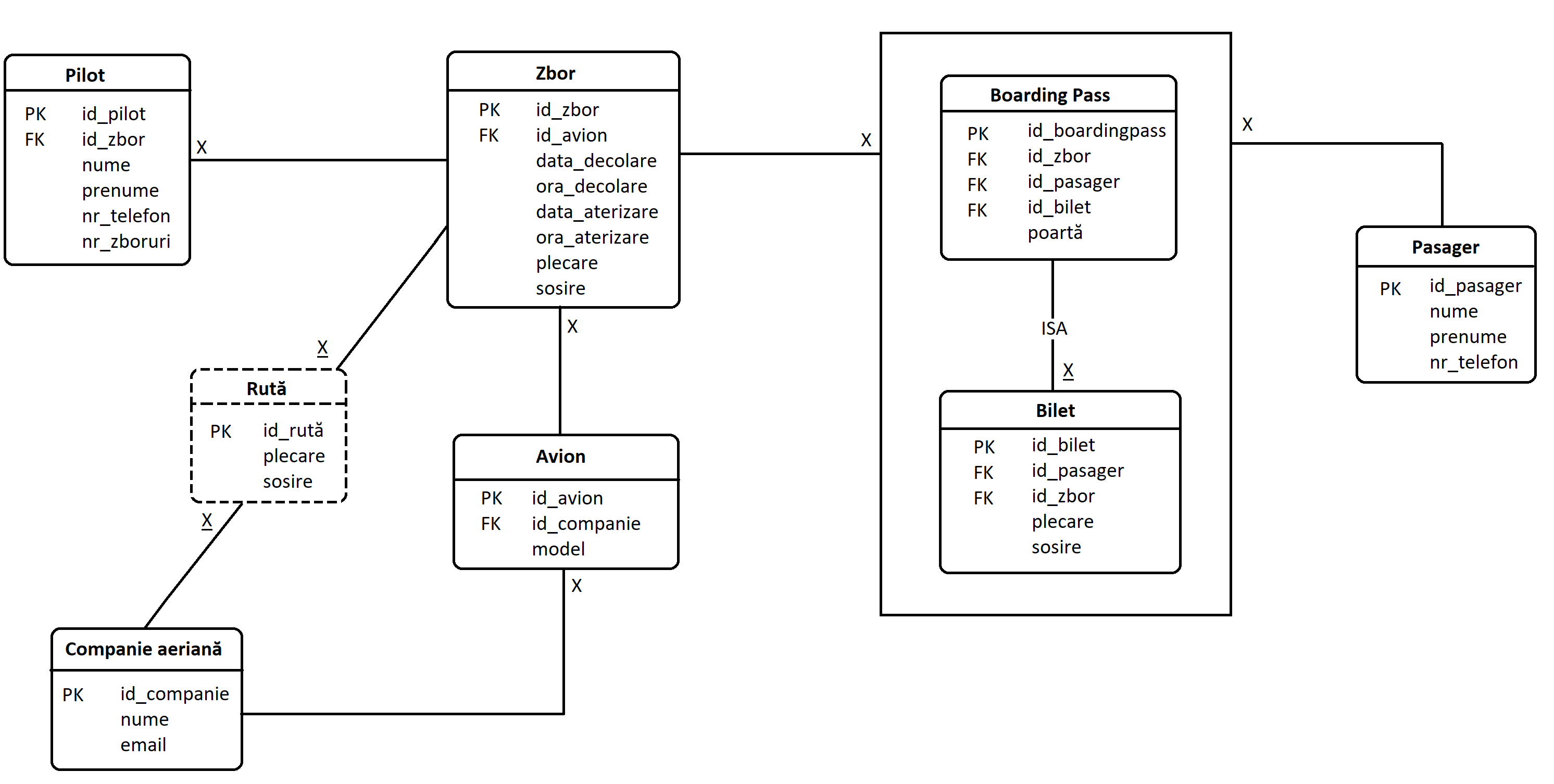
**Cerința 6**

Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



**Cerința 7**

Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



**Cerința 8**

Enumerarea schemelorrelaționalecorespunzătoare diagramei conceptualeproiectate la punctul 7.

Pilot(PK id\_pilot, FK id\_zbor, nume, prenume, nr\_telefon, nr\_zboruri)

Zbor(PK id\_zbor, FK id\_avion, data\_decolare, ora\_decolare, data\_aterizare, ora\_aterizare, plecare, sosire)

Boarding Pass(PK id\_boardingpass, FK id\_zbor, FK id\_pasager, FK id\_bilet, poartă)  
Bilet(PK id\_bilet, FK id\_pasager, FK id\_zbor, plecare, sosire)

Pasager(PK id\_pasager, nume, prenume, nr\_telefon)

Avion(PK id\_avion, FK id\_companie, model)

Companie aeriană(PK id\_companie, nume, email)

Rută(PK id\_rută, plecare, sosire)

**Cerința 9**

Realizarea normalizării până la forma normală 3(FN1-FN3).

NON-FN1

|  |  |  |
| --- | --- | --- |
| **Id\_pasager** | **Nume** | **Nr\_telefon** |
| 300 | Popa Marisa | 0754554123 |
| 301 | Dumitrescu Anais | 0745200014 |
| 302 | Anghelescu Mihail | 0742100667 |

FN1

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_pasager** | **Nume** | **Prenume** | **Nr\_telefon** |
| 300 | Popa | Marisa | 0754554123 |
| 301 | Dumitrescu | Anais | 0745200014 |
| 302 | Anghelescu | Mihail | 0742100667 |

NON-FN2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id\_pasager** | **Id\_zbor** | **Plecare** | **Sosire** | **Poartă** |
| 300 | 10 | Bucuresti | Paris | 7 |
| 301 | 11 | Amsterdam | Barcelona | 9 |
| 302 | 12 | Praga | Londra | 4 |

FN2

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_pasager** | **Id\_zbor** | **Plecare** | **Sosire** |
| 300 | 10 | Bucuresti | Paris |
| 301 | 11 | Amsterdam | Barcelona |
| 302 | 12 | Praga | Londra |

|  |  |  |
| --- | --- | --- |
| **Id\_pasager** | **Id\_zbor** | **Poartă** |
| 300 | 10 | 7 |
| 301 | 11 | 9 |
| 302 | 12 | 4 |

NON-FN3

|  |  |  |
| --- | --- | --- |
| **Id\_zbor** | **Data\_decolare** | **Ora\_decolare** |
| 10 | 15-06-2020 | 10:00 |
| 11 | 02-12-2022 | 23:15 |
| 12 | 23-05-2020 | 15:05 |

FN3

|  |  |
| --- | --- |
| **Id\_zbor** | **Data\_decolare** |
| 10 | 15-06-2020 |
| 11 | 02-12-2022 |
| 12 | 23-05-2020 |

|  |  |
| --- | --- |
| **Data\_decolare** | **Ora\_decolare** |
| 15-06-2020 | 10:00 |
| 02-12-2022 | 23:15 |
| 23-05-2020 | 15:05 |

**Cerința 10**

Crearea tabelelor în SQLși inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare table neasociativ; minimum 10 înregistrări întabelele asociative).

create table companie(

id\_companie int not null primary key,

nume varchar(50) not null,

email varchar(50));

create table avion(

id\_avion int not null primary key,

id\_companie int not null,

model varchar(50),

foreign key(id\_companie) references companie(id\_companie));

create table zbor(

id\_zbor int not null primary key,

id\_avion int not null,

data\_decolare date not null,

ora\_decolare varchar(5) not null,

data\_aterizare date not null,

ora\_aterizare varchar(5) not null,

plecare varchar(50) not null,

sosire varchar(50) not null,

foreign key(id\_avion) references avion(id\_avion));

create table pilot(

id\_pilot int not null primary key,

id\_zbor int not null,

nume varchar(50) not null,

prenume varchar(50) not null,

nr\_telefon varchar(10),

nr\_zboruri int,

foreign key(id\_zbor) references zbor(id\_zbor));

create table pasager(

id\_pasager int not null primary key,

nume varchar(50),

prenume varchar(50),

nr\_telefon varchar(10));

create table bilet(

id\_bilet int not null primary key,

id\_pasager int not null,

id\_zbor int not null,

plecare varchar(50) not null,

sosire varchar(50) not null,

foreign key(id\_pasager) references pasager(id\_pasager),

foreign key(id\_zbor) references zbor(id\_zbor));

create table boardingpass(

id\_boardingpass int not null primary key,

id\_zbor int not null,

id\_pasager int not null,

id\_bilet int not null,

poarta int not null,

foreign key(id\_zbor) references zbor(id\_zbor),

foreign key(id\_pasager) references pasager(id\_pasager),

foreign key(id\_bilet) references bilet(id\_bilet));

create table ruta(

id\_companie int not null,

id\_zbor int not null,

plecare varchar(50),

sosire varchar(50),

constraint id\_ruta primary key(id\_companie, id\_zbor));

alter table ruta add foreign key(id\_companie) references companie(id\_companie);

alter table ruta add foreign key(id\_zbor) references zbor(id\_zbor);

insert into companie(id\_companie, nume, email)

values

(seq\_companie.nextval, 'WizzAir', 'wizzair@gmail.com');

insert into companie(id\_companie, nume, email)

values

(seq\_companie.nextval, 'RyanAir', 'ryanair@gmail.com');

insert into companie(id\_companie, nume, email)

values

(seq\_companie.nextval, 'BlueAir', 'blueair@gmail.com');

insert into companie(id\_companie, nume, email)

values

(seq\_companie.nextval, 'Delta', 'delta@gmail.com');

insert into companie(id\_companie, nume, email)

values

(seq\_companie.nextval, 'Lufthansa', 'lufthansa@gmail.com');

insert into companie(id\_companie, nume, email)

values

(seq\_companie.nextval, 'Tarom', NULL);

insert into avion(id\_avion, id\_companie, model)

values

(200, 100, 'Airbus A300');

insert into avion(id\_avion, id\_companie, model)

values

(201, 101, 'Airbus A321');

insert into avion(id\_avion, id\_companie, model)

values

(202, 102, 'Antonov An-26');

insert into avion(id\_avion, id\_companie, model)

values

(203, 103, 'Boeing 747');

insert into avion(id\_avion, id\_companie, model)

values

(204, 104, 'F6F Hellcat');

insert into zbor(id\_zbor, id\_avion, data\_decolare, ora\_decolare, data\_aterizare, ora\_aterizare, plecare, sosire)

values

(10, 200, to\_date('06/15/2020', 'MM/DD/RRRR'), '10:00', to\_date('06/15/2020', 'MM/DD/RRRR'), '13:45', 'Bucuresti', 'Paris');

insert into zbor(id\_zbor, id\_avion, data\_decolare, ora\_decolare, data\_aterizare, ora\_aterizare, plecare, sosire)

values

(11, 201, to\_date('12/02/2020', 'MM/DD/RRRR'), '23:15', to\_date('12/03/2020', 'MM/DD/RRRR'), '03:00', 'Amsterdam', 'Barcelona');

insert into zbor(id\_zbor, id\_avion, data\_decolare, ora\_decolare, data\_aterizare, ora\_aterizare, plecare, sosire)

values

(12, 202, to\_date('05/23/2020', 'MM/DD/RRRR'), '14:05', to\_date('05/23/2020', 'MM/DD/RRRR'), '16:45', 'Praga', 'Londra');

insert into zbor(id\_zbor, id\_avion, data\_decolare, ora\_decolare, data\_aterizare, ora\_aterizare, plecare, sosire)

values

(13, 203, to\_date('01/20/2020', 'MM/DD/RRRR'), '12:55', to\_date('01/20/2020', 'MM/DD/RRRR'), '17:05', 'Madrid', 'Bucuresti');

insert into zbor(id\_zbor, id\_avion, data\_decolare, ora\_decolare, data\_aterizare, ora\_aterizare, plecare, sosire)

values

(14, 204, to\_date('06/15/2020', 'MM/DD/RRRR'), '05:00', to\_date('06/15/2020', 'MM/DD/RRRR'), '07:35', 'Frankfurt', 'Sibiu');

insert into pilot(id\_pilot, id\_zbor, nume, prenume, nr\_telefon, nr\_zboruri)

values

(1, 10, 'Bondoc', 'Constantin', '0754255678', 123);

insert into pilot(id\_pilot, id\_zbor, nume, prenume, nr\_telefon, nr\_zboruri)

values

(2, 11, 'Popescu', 'Mihai', '0742697412', 345);

insert into pilot(id\_pilot, id\_zbor, nume, prenume, nr\_telefon, nr\_zboruri)

values

(3, 12, 'Diaconescu', 'Denis', '0766542274', 679);

insert into pilot(id\_pilot, id\_zbor, nume, prenume, nr\_telefon, nr\_zboruri)

values

(4, 13, 'Nastasoiu', 'Vasile', '0751004735', 512);

insert into pilot(id\_pilot, id\_zbor, nume, prenume, nr\_telefon, nr\_zboruri)

values

(5, 14, 'Petrescu', 'Alexandru', '0785421119', 289);

insert into pasager(id\_pasager, nume, prenume, nr\_telefon)

values

(seq\_pasager.nextval, 'Popa', 'Marisa', '0754554123');

insert into pasager(id\_pasager, nume, prenume, nr\_telefon)

values

(seq\_pasager.nextval, 'Dumitrescu', 'Anais', '0745200014');

insert into pasager(id\_pasager, nume, prenume, nr\_telefon)

values

(seq\_pasager.nextval, 'Anghelescu', 'Mihail', '0742100667');

insert into pasager(id\_pasager, nume, prenume, nr\_telefon)

values

(seq\_pasager.nextval, 'Barbu', 'Robert', '0798745612');

insert into pasager(id\_pasager, nume, prenume, nr\_telefon)

values

(seq\_pasager.nextval, 'Gaman', 'Andrei', '0755244863');

insert into bilet(id\_bilet, id\_pasager, id\_zbor, plecare, sosire)

values

(400, 300, 10, 'Bucuresti', 'Paris');

insert into bilet(id\_bilet, id\_pasager, id\_zbor, plecare, sosire)

values

(401, 301, 11, 'Amsterdam', 'Barcelona');

insert into bilet(id\_bilet, id\_pasager, id\_zbor, plecare, sosire)

values

(402, 302, 12, 'Praga', 'Londra');

insert into bilet(id\_bilet, id\_pasager, id\_zbor, plecare, sosire)

values

(403, 303, 13, 'Madrid', 'Bucuresti');

insert into bilet(id\_bilet, id\_pasager, id\_zbor, plecare, sosire)

values

(404, 304, 14, 'Frankfurt', 'Sibiu');

insert into boardingpass(id\_boardingpass, id\_zbor, id\_pasager, id\_bilet, poarta)

values

(500, 10, 300, 400, 7);

insert into boardingpass(id\_boardingpass, id\_zbor, id\_pasager, id\_bilet, poarta)

values

(501, 11, 301, 401, 9);

insert into boardingpass(id\_boardingpass, id\_zbor, id\_pasager, id\_bilet, poarta)

values

(502, 12, 302, 402, 4);

insert into boardingpass(id\_boardingpass, id\_zbor, id\_pasager, id\_bilet, poarta)

values

(503, 13, 303, 403, 11);

insert into boardingpass(id\_boardingpass, id\_zbor, id\_pasager, id\_bilet, poarta)

values

(504, 14, 304, 404, 14);

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

values

(100, 10, 'Bucuresti', 'Paris');

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

values

(101, 11, 'Amsterdam', 'Barcelona');

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

values

(102, 12, 'Praga', 'Londra');

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

values

(103, 13, 'Madrid', 'Bucuresti');

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

values

(104, 14, 'Frankfurt', 'Sibiu');

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

values

(100, 14, 'Viena', 'Moscova');

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

values

(101, 13, 'Londra', 'Bucuresti');

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

values

(102, 11, 'Berlin', 'Paris');

insert into ruta(id\_companie, id\_zbor, plecare, sosire)

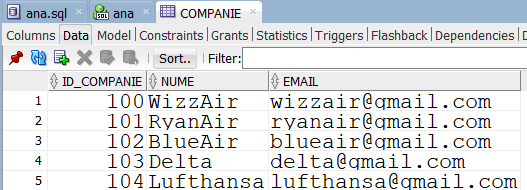
values

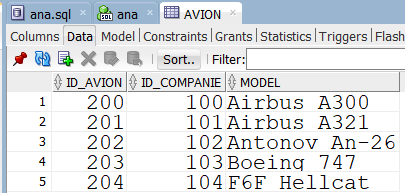
(103, 12, 'Roma', 'Atena');

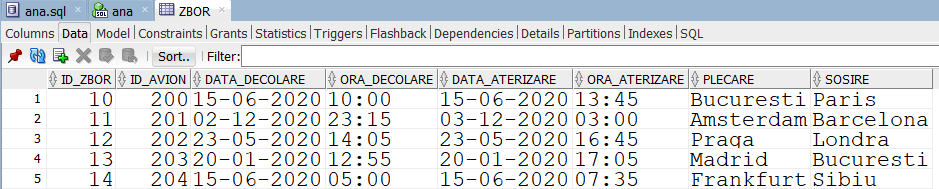
insert into ruta(id\_companie, id\_zbor, plecare, sosire)

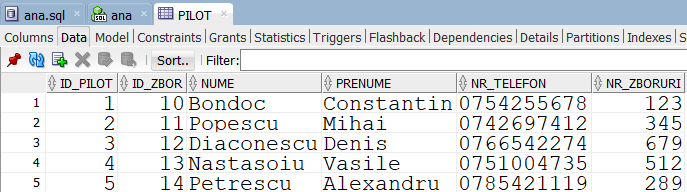
values

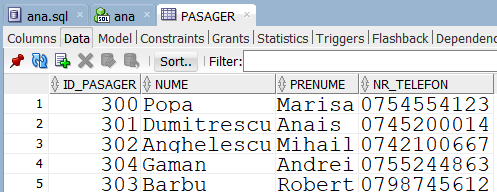
(104, 10, 'Bucuresti', 'Napoli');

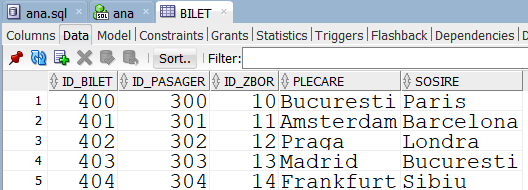


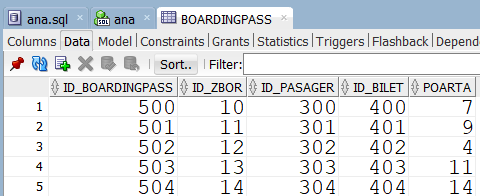


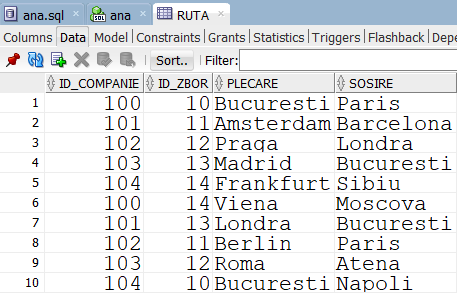












**Cerința 11**

Formulați în limbaj natural și implementați 5cereri SQL complexece vor utiliza, în ansamblul lor, următoarele elemente:

•operație joinpe cel puțin 4 tabele

•filtrare la nivel delinii

•subcereri sincronizateîn care intervin cel puțin 3 tabele

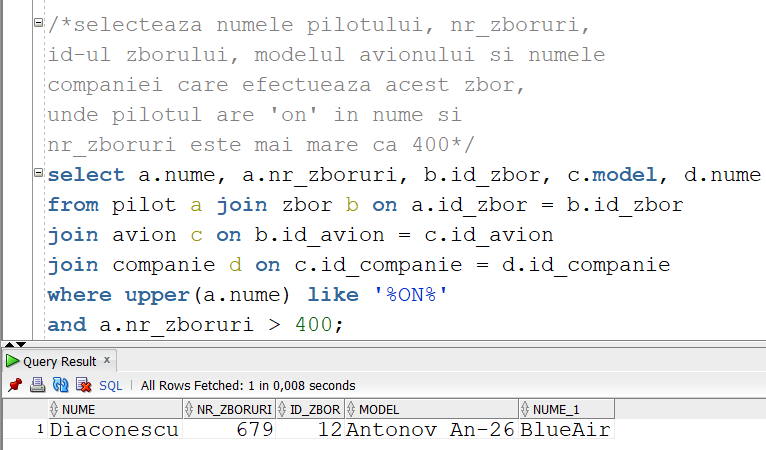
•subcereri nesincronizateîn care intervin cel puțin 3 tabele

•grupări de date, funcții grup, filtrare la nivel degrupuri

•ordonări

•utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE

•utilizarea a cel puțin 1 bloc de cerere(clauza WITH)



select a.nume, a.nr\_zboruri, b.id\_zbor, c.model, d.nume

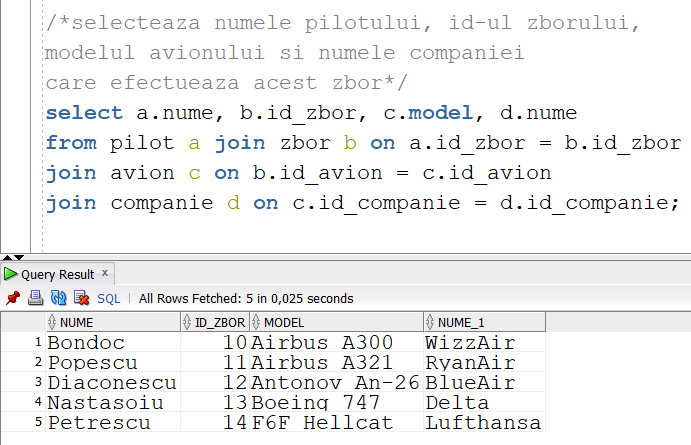
from pilot a join zbor b on a.id\_zbor = b.id\_zbor

join avion c on b.id\_avion = c.id\_avion

join companie d on c.id\_companie = d.id\_companie

where upper(a.nume) like '%ON%'

and a.nr\_zboruri > 400;

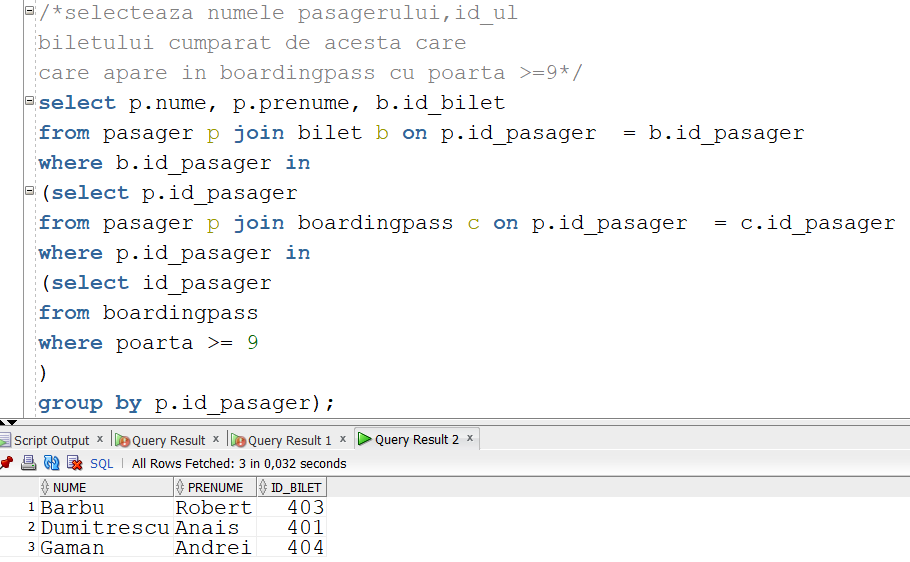


select a.nume, b.id\_zbor, c.model, d.nume

from pilot a join zbor b on a.id\_zbor = b.id\_zbor

join avion c on b.id\_avion = c.id\_avion

join companie d on c.id\_companie = d.id\_companie;



select p.nume, p.prenume, b.id\_bilet

from pasager p join bilet b on p.id\_pasager = b.id\_pasager

where b.id\_pasager in

(select p.id\_pasager

from pasager p join boardingpass c on p.id\_pasager = c.id\_pasager

where p.id\_pasager in

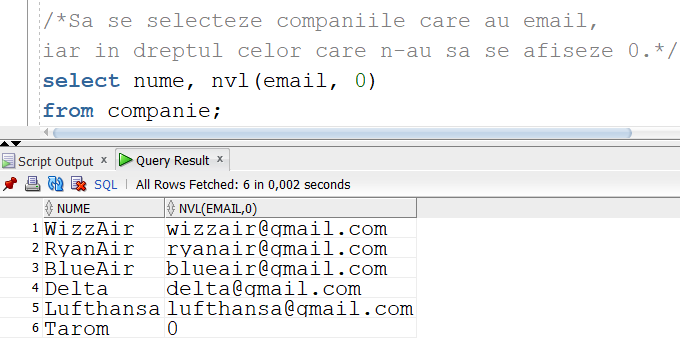
(select id\_pasager

from boardingpass

where poarta >= 9

)

group by p.id\_pasager);



select nume, nvl(email, 0)

from companie;

**Cerința 12**

Implementarea a 3operații de actualizare sau suprimare a datelor utilizând subcereri.

/\*Sa se stearga zborurile care se efectueaza cu avioane

care in model contin litera 'r'\*/

delete from zbor

where id\_avion in (select id\_avion

from avion

where model like '%r%'

);

/\*Sa se actualizeze orasul de sosire

cu 'Bacau' pentru zborurile

al caror id este 13 sau 14\*/

update zbor

set sosire = 'Bacau'

where id\_zbor in(select id\_zbor

from zbor

where id\_zbor=13 or id\_zbor=14);

/\*Sa se stearga zborurile a caror

avioane sunt tinute de compania RyanAir\*/

delete from zbor

where id\_zbor in (select id\_zbor

from avion

where id\_companie=(select id\_companie

from companie

where nume='RyanAir'));

**Cerința 13**

Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele(punctul 10).

create sequence seq\_pasager

increment by 1

start with 300

minvalue 1

maxvalue 99999

nocycle;

create sequence seq\_companie

increment by 1

start with 100

minvalue 1

maxvalue 99999

nocycle;

**Cerința 14**

Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

