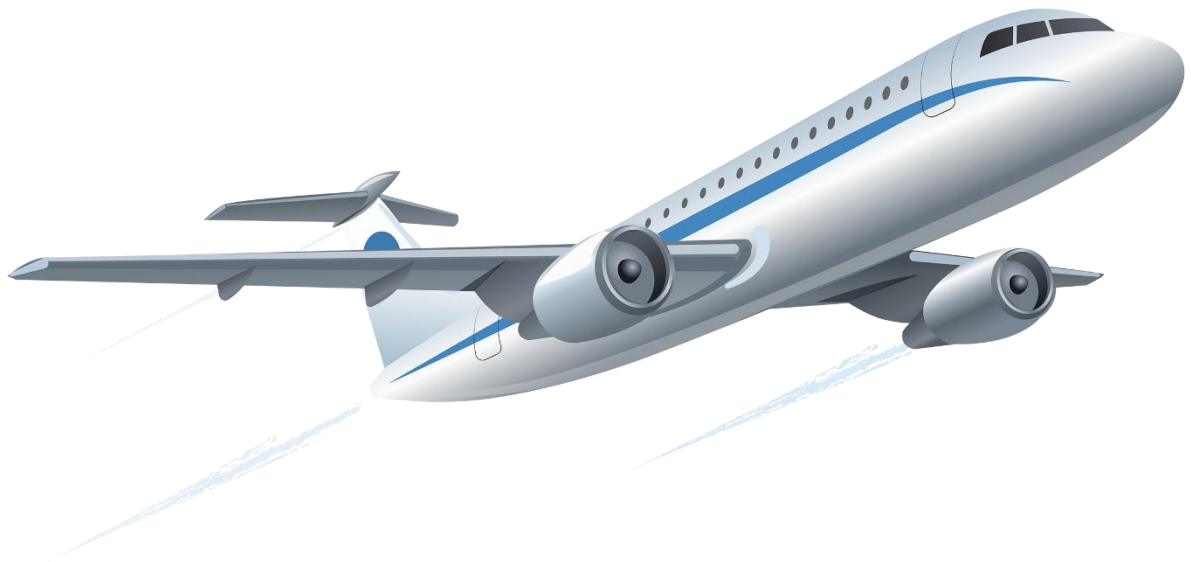


Aplicație pentru  
Gestionarea traficului unui aeroport  
“Aero Manager”



Proiect realizat de:  
Sumanariu Bianca Dorina

C112C

# Cuprins

1. Descriere.....	3
2. Tabele .....	4
3. Crearea bazei de date si a grupului de fisiere aferente .....	5
4. Diagrama bazei de date.....	6
5. Crearea tabelelor.....	7
6. Constrângeri privind cheile străine și primare .....	12
7. Inserarea datelor .....	16
8. Interogari SELECT.....	22
9. Interogari folosind CTE-uri .....	39
10. Actualizari de date (UPDATE) .....	43
11. Stergerea de date (DELETE).....	54
12. Vizualizari.....	64
13. Triggere.....	70
14. Proceduri stocate .....	76
15. Tranzacții .....	82
16. Back-up .....	94
17. Evidenta numarului de query-uri .....	94

## Descriere

Proiectul urmărește realizarea unei aplicații capabile să gestioneze baza de date a unui aeroport care să permită monitorizarea și gestionarea zborurilor, pasagerilor, echipajelor și a altor aspecte legate de traficul aerian. Proiectul urmărește să simplifice procesul de gestionare a traficului și să îmbunătățească eficiența și siguranța zborurilor. Baza de date poate fi utilizată pentru a genera rapoarte și analize cu privire la traficul aerian, incluzând date despre numărul de zboruri, numărul de pasageri, destinațiile populare, companiile aeriene și multe altele. Aceste informații pot fi utile pentru planificarea viitoarelor zboruri, evaluarea performanței și a altor decizii strategice legate de traficul aerian.

În plus, baza de date poate ajuta la gestionarea personalului aeroportului prin urmărirea informațiilor legate de departamente, funcții și sarcini. Personalul poate monitoriza performanța angajaților, asignarea sarcinilor și alocarea resurselor. Aceste informații pot fi folosite pentru a identifica deficiențe în procesele de lucru și pentru a îmbunătăți productivitatea și eficiența personalului.

În consecință, aplicația prezentată anterior va avea ca punct central tabela **Zboruri**.























- Fiecare zbor are o destinație reprezentată de un singur aeroport, astfel tabela **Aeroporturi** va fi legată printr-o cheie străină de tabela **Zboruri**.
- Fiecare aeroport are o locație care se află într-o țară, tabela **Aeroporturi** fiind legată printr-o cheie străină de tabela **Locatii**, aceasta fiind legată la rândul său de tabela **Tari**.
- Pentru organizarea fiecărui zbor va fi nevoie de îndeplinirea unor sarcini stocate în tabela **Tasks**, de care se va ocupa personalul aeroportului prezentat în tabela **Staff**, aceste tabele fiind legate prin tabela de legătură **StaffTasks**. De asemenea, fiecare angajat va îndeplini o funcție, fiind nevoie de o nouă tabelă – **Repartizare\_functii**, pentru ca angajații prezinta posibilitatea de a fi ocupat mai multe funcții într-un departament (legatură prin cheie străină cu **Departamente**).
- Fiecare zbor va dispune de bilete, astfel încât va exista o tabelă **Bilete** care va stoca fiecare bilet, făcându-se legatură printr-o altă tabelă **BiletePerZbor** legată de tabela **Zboruri**. Un călător va primi un bilet, fiindu-i stocate datele în tabela **Calatori**.
- Fiecare zbor va avea un tip de avion stocat în tabela **Avioane**, fiecare aparținând unei companii, fiind stocată în tabela **Companii**.

La nivelul relațiilor M:N:

- Un zbor poate fi alocat la mai multe porți de îmbarcare, iar o poartă de îmbarcare poate fi alocată la mai multe zboruri. Pentru a crea o relație M:N între aceste două tabele, tabelul **Zboruri** este asociat cu tabelul **Gate** prin tabelul **ZboruriGates**.
- Un angajat poate avea asignate mai multe sarcini, iar o sarcină poate fi asignată la mai mulți angajați. Pentru a crea o relație M:N între aceste două tabele există o tabelă **StaffTasks**.
- Un task poate fi alocat la mai multe zboruri, iar un zbor poate avea mai multe task-uri. Pentru a crea o relație M:N între cele două tabele (**Zboruri**, **Tasks**) există tabelul **TasksperZbor**.
- Un angajat poate avea mai multe impozite descries într-un contract, iar mai multe contracte pot avea un impozit, așa că am realizat relația M:N prin tabelul **ContracteImpozite**.

# Tabele

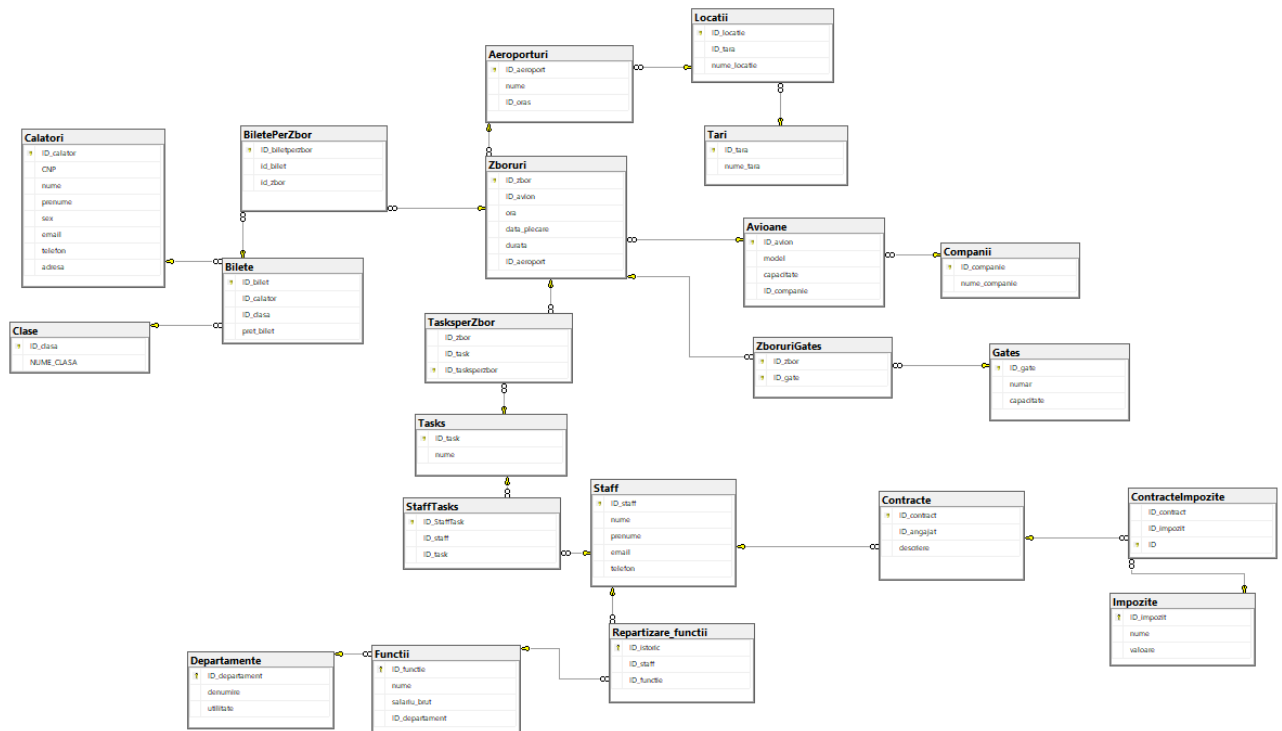
Conform celor prezentate mai sus in ceea ce priveste logica aplicatiei, s-a ajuns la urmatoarea structura de tabele:

- +  dbo.Aeroporturi
- +  dbo.Avioane
- +  dbo.Bilete
- +  dbo.BiletePerZbor
- +  dbo.Calatori
- +  dbo.Clase
- +  dbo.Companii
- +  dbo.Contracte
- +  dbo.ContractelImpozite
- +  dbo.Departamente
- +  dbo.Functii
- +  dbo.Gates
- +  dbo.Impozite
- +  dbo.Locatii
- +  dbo.Repartizare\_functii
- +  dbo.Staff
- +  dbo.StaffTasks
- +  dbo.Tari
- +  dbo.Tasks
- +  dbo.TaskspersZbor
- +  dbo.Zboruri
- +  dbo.ZboruriGates

## Crearea bazei de date si a grupului de fisiere aferente

```
create database GestiuneAeroport3
ON PRIMARY
(
Name = PrimaryData,
FileName = 'C:\DataBase3\Primary.mdf',
size = 10MB, -- KB, Mb, GB, TB
maxsize = unlimited,
filegrowth = 1GB
),
FILEGROUP fgCurrent
(
Name = DataA,
FileName = 'C:\DataBase3\GroupA.ndf',
size = 10MB, -- KB, Mb, GB, TB
maxsize = unlimited,
filegrowth = 1GB
),
(
Name = DataB,
FileName = 'C:\DataBase3\GroupB.ndf',
size = 10MB, -- KB, Mb, GB, TB
maxsize = unlimited,
filegrowth = 1GB
),
FILEGROUP fgArchive
(
Name = ArchiveData,
FileName = 'C:\DataBase3\Archive.ndf',
size = 10MB, -- KB, Mb, GB, TB
maxsize = unlimited,
filegrowth = 1GB
)
LOG ON
(
Name = DataLog,
FileName = 'C:\DataBase3\Log.ldf',
size = 10MB, -- KB, Mb, GB, TB
maxsize = unlimited,
filegrowth = 1024MB)
```

# Diagrama bazei de date



# Crearea tabelelor

1. Crearea tabelii Tari:

```
create table Tari
(ID_tara int IDENTITY(1,1) primary key,
nume_tara varchar(20));
```

2. Crearea tabelii Locatii care va contine orasele (destinatie) :

```
create table Locatii
(ID_locatie int IDENTITY(1,1) primary key,
ID_tara int ,
nume_locatie varchar(20),
constraint ID_tara_fk foreign key(ID_tara) references Tari(ID_tara) ON UPDATE
CASCADE ON DELETE CASCADE);
```

3. Crearea tabelii Aeroporturi va contine aeroporturile ce se afla in orasele destinatie:

```
create table Aeroporturi
(ID_aeroport int IDENTITY(1,1) primary key,
nume varchar(50),
ID_oras int,
constraint FK_ID_locatie foreign key(ID_oras) references Locatii(ID_locatie) ON
UPDATE CASCADE ON DELETE CASCADE);
```

4. Crearea tabelului Companii contine companiile de care apartin avioanele:

```
create table Companii
(ID_companie int IDENTITY(1,1) primary key,
nume_companie varchar(20));
```

5. Crearea tabelului Avioane :

```
create table Avioane
(ID_avion int primary key,
model varchar(20),
capacitate int,
ID_companie int,
CONSTRAINT FK_ID_companie_ FOREIGN KEY (ID_companie) REFERENCES
Companii(ID_companie));
```

6. Crearea tabelii Zboruri :

```
create table Zboruri
( ID_zbor int IDENTITY(1,1) primary key,
  ID_avion int,
  ora time,
  data_plecure date,
  durata int,
  ID_aeroport int,
  constraint FK_ID_avion foreign key(ID_avion) references Avioane(ID_avion) ON
  UPDATE CASCADE ON DELETE CASCADE,
  constraint FK_ID_aeroport foreign key(ID_aeroport) references
  Aeroporturi(ID_aeroport) ON UPDATE CASCADE ON DELETE CASCADE);
```

7. Crearea tabelii Gates care contine terminaalele (portile) de imbarcare asignate fiecarui zbor:

```
create table Gates
(
  ID_gate int IDENTITY(1,1) primary key,
  numar int);
```

8. Crearea tabelii ZboruriGates care contine repartizarile portilor pentru fiecare zbor (valabil si invers):

```
CREATE TABLE ZboruriGates
(
  ID_zbor INT,
  ID_gate INT,
  PRIMARY KEY (ID_zbor, ID_gate),
  constraint FK_ID_zbor_ FOREIGN KEY (ID_zbor) REFERENCES Zboruri(ID_zbor) ON DELETE
  CASCADE,
  constraint FK_ID_gate FOREIGN KEY (ID_gate) REFERENCES Gates(ID_gate) ON DELETE
  CASCADE
);
```

9. Crearea tabelii Clase :

```
create table Clase
(ID_clasa int IDENTITY(1,1) primary key,
NUME_CLASA varchar(10)constraint NUME_CLASA_CK check(NUME_CLASA in('ECONOMIC',
'BUSINESS')));
```

10. Crearea tabelii Calatori care contine pasagerii :

```
create table Calatori
(
  ID_calator int IDENTITY(1,1) primary key,
  CNP varchar(13),
  nume varchar(20),
  prenume varchar(20),
  sex varchar(10),
  email varchar(20),
```



```

telefon varchar(15),
adresa varchar(30),
constraint CNP_UQ unique(CNP),
constraint NUME_nn check(NUME is not null),
constraint SEX_CK check(SEX in('FEMININ', 'MASCULIN')));

```

11. Crearea tabelii Bilete care contine biletele fiecarui calator:

```

create table Bilete
(
ID_bilet int IDENTITY(1,1) primary key,
ID_calator int,
ID_clasa int,
pret_bilet int,
constraint FK_ID_calator foreign key(ID_calator) references Calatori(ID_calator)
ON UPDATE CASCADE ON DELETE CASCADE,
constraint FK_ID_CLASA foreign key(ID_clasa) references Clase(ID_clasa) ON
UPDATE CASCADE ON DELETE CASCADE);

```

12. Crearea tabelii BiletePerZbor care contine legaturile dintre bilete si zboruri:

```

CREATE TABLE BiletePerZbor (
ID_biletperzbor INT IDENTITY(1,1) PRIMARY KEY,
id_bilet INT NOT NULL,
id_zbor INT NOT NULL,
constraint FK_ID_bilet FOREIGN KEY (id_bilet) REFERENCES Bilete(ID_bilet) ON
UPDATE CASCADE ON DELETE CASCADE,
constraint FK_ID_zbor FOREIGN KEY (id_zbor) REFERENCES Zboruri(ID_zbor) ON
UPDATE CASCADE ON DELETE CASCADE
);

```

13. Crearea tabelii Departamente care contine sectiunile din aeroport pe care este impartita munca angajatilor:

```

create table Departamente
(
ID_departament int IDENTITY(1,1) primary key,
denumire varchar(50),
utilitate varchar(50)
);

```

14. Crearea tabelii Functii care contine functiile angajatilor in functie de departament:

```

create table Functii
(
ID_functie int IDENTITY(1,1) primary key,
nume varchar(50) not null,
salariu_brut int,
ID_departament int,
constraint FK_ID_departament foreign key(ID_departament) references
Departamente(ID_departament) ON UPDATE CASCADE ON DELETE CASCADE
);

```

15. Crearea tabela Staff care contine angajatii din aeroport care se ocupa de desfasurarea procesului inaita fiecarui zbor:

```
create table Staff (  
ID_staff INT IDENTITY(1,1) PRIMARY KEY,  
nume VARCHAR(50) NOT NULL,  
prenume VARCHAR(50) NOT NULL,  
email VARCHAR(100) UNIQUE NOT NULL,  
telefon VARCHAR(20),  
);
```

16. Crearea tabelii Repartizare\_functii care face legatura dintre angajati si functiile pe care le ocupa:

```
create table Repartizare_functii  
(  
ID_istoric int IDENTITY(1,1) primary key,  
ID_staff int,  
ID_functie int,  
constraint FK_ID_staff_ foreign key(ID_staff) references Staff(ID_staff) ON  
UPDATE CASCADE ON DELETE CASCADE,  
constraint FK_ID_functie foreign key(ID_functie) references Functii(ID_functie)  
ON UPDATE CASCADE ON DELETE CASCADE  
);
```

17. Crearea tabelii Contracte care contine contractele angajatilor :

```
create table Contracte  
(  
ID_contract int IDENTITY(1,1) primary key,  
ID_angajat int,  
constraint FK_ID_angajat foreign key(ID_angajat) references Staff(ID_staff) ON  
UPDATE CASCADE ON DELETE CASCADE,  
descriere varchar(50)  
);
```

18. Crearea tabelii Impozite care contine taxele aplicate pe salariu:

```
create table Impozite  
(  
ID_impozit int IDENTITY(1,1) primary key,  
nume varchar(50),  
valoare int  
);
```

19. Crearea tabelii ContracteImpozite care face legatura dintre impozite si contractele angajatilor:

```
create table ContracteImpozite  
(  
ID int IDENTITY(1,1) primary key,  
ID_contract int,
```

```

ID_impozit int
constraint FK_ID_contract foreign key(ID_contract) references
Contracte(ID_contract) ON UPDATE CASCADE ON DELETE CASCADE,
constraint FK_ID_impozit foreign key(ID_impozit) references Impozite(ID_impozit)
ON UPDATE CASCADE ON DELETE CASCADE,
);

```

20. Crearea tabelii Tasks care contine operatiunile necesare desfasurarii activitatii in aeroport:

```

create table Tasks
(
ID_task int IDENTITY(1,1) primary key,
nume varchar(50)
);

```

21. Crearea tabelii TaskspersZbor care repartizarile operatiunilor pentru fiecare zbor:

```

create table TaskspersZbor
(
ID_zbor int,
ID_task int,
FOREIGN KEY (ID_zbor) REFERENCES Zboruri(ID_zbor) ON UPDATE CASCADE ON DELETE
CASCADE,
FOREIGN KEY (ID_task) REFERENCES Tasks(ID_task) ON UPDATE CASCADE ON DELETE
CASCADE
);

```

22. Crearea tabelii StaffTasks care contine repartizarile operatiunilor pentru fiecare angajat:

```

create table StaffTasks
(
ID_StaffTask int identity (1,1) primary key,
ID_staff int,
ID_task int,
constraint FK_ID_staff foreign key(ID_staff) references Staff(ID_staff) ON UPDATE
CASCADE ON DELETE CASCADE,
constraint FK_ID_task foreign key(ID_task) references Tasks(ID_task) ON UPDATE
CASCADE ON DELETE CASCADE
);

```

# Constrângeri privind cheile străine și primare

## Tabel Aeroport

PK\_Aeroport\_A327B01759CE0263  
FK\_ID\_locatie

## Tabel Avioane

PK\_Avioane\_CFE99C3C3A35BC32  
FK\_ID\_companie\_

## Tabel Bilete

PK\_Bilet\_BDEC0C9F34234FA6  
FK\_ID\_calator  
FK\_ID\_CLASA

## Tabel BiletePerZbor

PK\_BiletePe\_17996D79E3339D5B  
FK\_ID\_bilet  
FK\_ID\_zbor

## Tabel Calatori

PK\_Calatori\_688B0A5D447387A6  
CNP\_UQ

Tabel Clase

PK\_Clase\_1BBE1A9BD499D83A

Tabel Companii

PK\_Companii\_EF0874BEC6BD7DF4

Tabel Contracte

PK\_Contract\_6877B145E6786624  
FK\_ID\_angajat

Tabel ContracteImpozite

PK\_Contract\_3214EC27BC27A620  
FK\_ID\_contract  
FK\_ID\_impozit

Tabel Departamente

PK\_Departam\_80EAE66CF53FBEFE

Tabel Functii

PK\_Functii\_781106A9979807AB  
FK\_Functii\_ID\_depa\_6E01572D

## Tabel Gates

PK\_Gates\_0D9D047B7DF24165

## Tabel Impozite

PK\_Impozite\_AA10E197C23397FC

## Tabel Locatii

PK\_Locatii\_940ADEB3D0E4A025

ID\_tara\_fk

## Tabel Repartizare\_functii

PK\_Repartiz\_ECC23D6676E5DB4D

FK\_ID\_functie

FK\_ID\_staff\_

## Tabel Staff

PK\_Staff\_E04EF0472DD93B93

UQ\_Staff\_AB6E616491F25A06

## Tabel StaffTaks

PK\_StaffTas\_23A402E2F62BF201

FK\_ID\_staff

FK\_ID\_task

## Tabela Tari

PK\_Tari\_E751DF0FDD74EEEB

## Tabela Tasks

PK\_Tasks\_E75142BB89C0B996

## Tabela TaskspersZbor

PK\_Taskspers\_C61C76D1E208BBBE

FK\_TaskspersZ\_ID\_ta\_31B762FC

FK\_TaskspersZ\_ID\_zb\_30C33EC3

## Tabela Zboruri

PK\_Zboruri\_FEC39E0830BA960B

FK\_ID\_aeroport

FK\_ID\_avion

## Tabela ZboruriGates

PK\_ZboruriG\_5E1A4E4FA6FFFDEA

FK\_ID\_gate

FK\_ID\_zbor\_

# Inserarea datelor

1. Inserarea datelor in tabela Tari:

```
INSERT INTO Tari (nume_tara) VALUES
('Romania'),
('SUA'),
('Spain'),
('Marea Britanie'),
('Germania'),
('Japonia'),
('China'),
('Rusia');
```

2. Inserarea datelor in tabela Locatii:

```
INSERT INTO Locatii (ID_tara, nume_locatie) VALUES
(7, 'Bucuresti'),
(8, 'New York'),
(1, 'Roma'),
(9, 'Barcelona'),
(10, 'Londra'),
(11, 'Berlin'),
(12, 'Tokyo'),
(13, 'Beijing'),
(14, 'Moscova');
```

3. Inserarea datelor in tabela Aeroporturi:

```
INSERT INTO Aeroporturi (nume, ID_oras) VALUES
('Charles de Gaulle', 3),
('Marco Polo', 5),
('Henri Coanda', 13),
('JFK', 14),
('Fiumicino', 15),
('El Prat', 16),
('Heathrow', 17),
('Tegel', 18),
('Narita', 19),
('Beijing Capital', 20);
```

4. Inserare datelor in tabela Companii:

```
INSERT INTO Companii (nume_comanie) VALUES
('Blue Air'),
('American Airlines'),
('Air France'),
('Alitalia'),
('Iberia'),
('British Airways'),
('Lufthansa');
```



```
('ANA'),  
( 'Air China'),  
( 'Aeroflot');
```

5. Inserarea datelor in tabela Avioane:

```
INSERT INTO Avioane (ID_Companie, model, capacitate, ID_avion)  
VALUES (12, 'Boeing 747', 416, 1),  
(13, 'Airbus A380', 853, 2),  
(14, 'Boeing 737', 215, 3),  
(15, 'Airbus A320', 180, 4),  
(16, 'Boeing 777', 396, 5),  
(17, 'Airbus A330', 335, 6),  
(18, 'Embraer E190', 100, 7),  
(19, 'Bombardier CRJ700', 78, 8),  
(20, 'ATR 72', 74, 9),  
(21, 'Sukhoi Superjet 100', 103, 10);
```

6. Inserarea datelor in tabela Gates:

```
INSERT INTO Gates (numar, capacitate) VALUES  
(1,100),(2,110),(3,120),(4,130),(5,140),(6,150),(7,160),(8,170),(9,180),(10,190);
```

7. Inserarea datelor in tabela Clase:

```
INSERT INTO Clase (NUME_CLASA) VALUES ('ECONOMIC');  
INSERT INTO Clase (NUME_CLASA) VALUES ('BUSINESS');
```

8. Inserarea datelor in tabela Calatori:

```
INSERT INTO Calatori (CNP, nume, prenume, sex, email, telefon, adresa) VALUES  
( '1234567890123', 'Popescu', 'Ana', 'FEMININ', 'ana.p@gmail.com', '0721123456',  
'Str. Libertatii nr. 10'),  
( '1234567890124', 'Ionescu', 'Alex', 'MASCULIN', 'alex.i@gmail.com', '0723123456',  
'Str. Independentei nr. 20'),  
( '1234567890125', 'Popa', 'Maria', 'FEMININ', 'maria.p@gmail.com', '0731123456',  
'Str. Unirii nr. 15'),  
( '1234567890126', 'Vasilescu', 'Alexandra', 'FEMININ', 'alex.v@gmail.com',  
'0741123456', 'Str. Mihai Viteazul nr. 5'),  
( '1234567890127', 'Popescu', 'Mihai', 'MASCULIN', 'mihai.p@gmail.com', '0726123456',  
'Str. Republicii nr. 7'),  
( '1234567890128', 'Dumitru', 'Diana', 'FEMININ', 'diana.d@gmail.com', '0751123456',  
'Str. Ion Creanga nr. 12'),  
( '1234567890129', 'Radulescu', 'Andrei', 'MASCULIN', 'andrei.r@gmail.com',  
'0729123456', 'Str. Gheorghe Doja nr. 14'),  
( '1234567890130', 'Georgescu', 'Elena', 'FEMININ', 'elena.g@gmail.com',  
'0742123456', 'Str. Tudor Vladimirescu nr. 17');
```

```
( '1234567890131', 'Petrescu', 'Marian', 'MASCULIN', 'marian.p@gmail.com',
'0732123456', 'Str. Vasile Alecsandri nr. 19'),
( '1234567890132', 'Stefanescu', 'Alina', 'FEMININ', 'alina.s@gmail.com',
'0761123456', 'Str. Stefan cel Mare nr. 9');
```

9. Inserarea datelor in tabela Bilete:

```
INSERT INTO Bilete(ID_calator, ID_clasa, pret_bilet) VALUES
(66,3,100),
(67,4,2000),
(68,3,200),
(69,4,2000),
(70,3,300),
(71,3,200),
(72,4,2000),
(73,3,300),
(74,3,400),
(75,4,2200);
```

10. Inserarea datelor in tabela Departamente:

```
INSERT INTO Departamente (denumire, utilitate) VALUES
('Tehnologia informatiei', 'Se ocupa cu administrarea sistemelor informatice'),
('Siguranta', 'Verificarea pasagerilor si a bagajelor'),
('Operatiuni aeroportuare', 'Asigura pregatirea si gestionarea operatiunilor'),
('Informatii pasageri', 'Asigura furnizarea de informatii catre pasageri'),
('Mentenanata', 'Asigura intretinerea si repararea echipamentelor'),
('Aprovizionare', 'Aprovizionarea aeroportului cu diverse resurse'),
('Financiar-contabil', 'Se ocupa cu administrarea resurselor financiare'),
('Marketing', 'Promovarea serviciilor aeroportuare'),
('Resurse umane', 'Gestionarea resurselor umane ale aeroportului'),
('Planificare si dezvoltare', 'Se ocupa cu planificarea infrastructurii');
```

11. Inserarea in tabela Functii:

```
INSERT INTO Functii (nume, salariu_brut, ID_departament) VALUES
('Manager', 10000, 19),
('Programator', 5000, 10),
('Contabil', 8000, 16),
('Tehnician', 4000, 14),
('Supraveghetor si ajutor imbarcare', 4500, 12),
('Responsabil mentenanata', 4200, 14),
('Casier', 2700, 16),
('Agent de securitate', 3000, 11),
('Inginer aprovizionare', 5500, 15),
('Manager marketing', 6000, 17),
('Manager resurse umane', 5500, 18);
```

12. Inserarea datelor in tabela Impozite:

```

INSERT INTO Impozite (nume, valoare) VALUES
('Impozit pe venitul salarial', 10), -- Impozitul pe venitul salarial
('Contributie la CASS', 5), -- Contribuția la sistemul de asigurări sociale de
sănătate (CASS)
('Contributie la CAS', 25), -- Contribuția la sistemul de asigurări sociale
(CAS)
('Contributie la Fondul de somaj', 2),
('Contributia la Fondul de garantare pentru plata creantelor salariale', 1),
('Contributia la Fondul pentru pensii administrate privat (Pilonul II)', 3),
('Contributia la Fondul de risc si accidente de munca - clasa de risc I', 0.2),
('Contributia la Fondul de risc si accidente de munca - clasa de risc II', 0.3);

```

### 13. Inserarea datelor in tabela Staff:

```

INSERT INTO Staff (nume, prenume, email, telefon) VALUES
('Popescu', 'Alex', 'alex.popescu@example.com', '0744555666'),
('Ionescu', 'Maria', 'maria.ionescu@example.com', '0733666777'),
('Georgescu', 'Andrei', 'andrei.georgescu@example.com', '0711223344'),
('Stanescu', 'Elena', 'elena.stanescu@example.com', '0722333444'),
('Dumitrescu', 'Ion', 'ion.dumitrescu@example.com', '0755667788'),
('Popa', 'Maria', 'maria.popa@example.com', '0766777888'),
('Mihai', 'Vlad', 'vlad.mihai@example.com', '0711223344'),
('Gheorghe', 'Andreea', 'andreea.gheorghe@example.com', '0744555666'),
('Radu', 'Ioana', 'ioana.radu@example.com', '0755667788'),
('Popescu', 'Mihai', 'mihai.popescu@example.com', '0766777888');

```

### 14. Inserarea datelor in tabela Repartizare\_functii:

```

INSERT INTO Repartizare_functii (ID_functie, ID_staff) VALUES
(4,13 ), (5,14), (6,15), (7,16), (8,17), (9,18), (10,19), (11,20), (12,21),
(13, 22)

```

### 15. Inserarea datelor in tabela Zboruri:

```

INSERT INTO Zboruri (ID_avion, ora, data_plecare, durata, ID_aeroport)
VALUES
(1, '09:00', '2023-12-10', 120, 4),
(2, '12:30', '2023-12-11', 180, 5),
(3, '14:45', '2023-12-12', 90, 6),
(4, '08:00', '2023-12-13', 240, 7),
(5, '10:15', '2023-12-14', 150, 8),
(6, '15:30', '2023-12-15', 180, 9),
(7, '11:00', '2023-12-16', 120, 10),
(8, '19:00', '2023-12-17', 360, 11),
(9, '02:00', '2023-12-18', 540, 12),
(10, '05:30', '2023-12-19', 480, 13);

```

### 16. Inserarea datelor in tabela Tasks:

```

INSERT INTO Tasks (nume) VALUES
('Verificarea tehnică a aeronavei înainte de decolare'),
('Asigurarea siguranței pasagerilor și a aeronavei'),

```

```

('Încărcarea și descărcarea bagajelor'),
('Aprovizionarea aeronavei cu combustibil și alte provizii necesare'),
('Îmbarcarea pasagerilor'),
('Verificarea biletelor și a documentelor pasagerilor'),
('Rezolvarea problemelor și întâmpinarea nevoilor pasagerilor'),
('Urmărirea plăților și a încasărilor și întocmirea rapoartelor  
financiare'),
('Gestionarea aprovizionării cu combustibil');

```

17. Inserarea datelor in tabela TasksPerZbor:

```

INSERT INTO TaskspersZbor (ID_task, ID_zbor) VALUES
(9,12), (10,13), (11,14), (12,15), (13,16), (8,17), (8,18), (9,19),
(10,20), (11,21)

```

18. Inserarea datelor in tabela Contracte:

```

INSERT INTO Contracte (ID_angajat, descriere) VALUES
(13, 'Contract de munca pe perioada nedeterminata'),
(14, 'Contract de munca pe perioada determinata'),
(15, 'Contract de colaborare'),
(16, 'Contract de prestari servicii'),
(17, 'Contract de ucenicie'),
(18, 'Contract de munca pe perioada determinata'),
(19, 'Contract de munca in regim de munca flexibil'),
(20, 'Contract de munca pe perioada determinata'),
(21, 'Contract de munca pe perioada nedeterminata'),
(22, 'Contract de munca pe perioada nedeterminata');

```

19. Inserarea datelor in tabela BiletePerZbor:

```

INSERT INTO BiletePerZbor (id_bilet, id_zbor) VALUES
(8, 14),
(9, 14),
(10, 14),
(11, 15),
(12, 15),
(13, 15),
(14, 15),
(15, 16),
(16, 16),
(17, 16);

```

20. Inserarea datelor in tabela StaffTasks:

```

INSERT INTO StaffTasks (ID_staff, ID_task) VALUES
(13,16),
(14,14),
(15,15),
(16,8),
(17,12),

```

```
(17,13),  
(18,8),  
(19,14),  
(20,9),  
(20,10),  
(21,11),  
(22,16);
```

21. Inserarea datelor in tabela ContracteImpozite:

```
INSERT INTO ContracteImpozite (ID_contract, ID_impozit) VALUES  
(7,14),  
(8,14),  
(7,15),  
(7,16),  
(8,15),  
(8,16),  
(9,14),  
(10,14),  
(11,14),  
(12,14),  
(13,14),  
(9,15),  
(16,21),  
(16,15),  
(16,14),  
(14,15),  
(14,18),  
(15,15),  
(15,19);
```

22. Inserarea datelor in tabela ZboruriGates:

```
INSERT INTO ZboruriGates(ID_zbor, ID_gate) VALUES  
(12,14),  
(12,24),  
(13,15),  
(14,16),  
(15,17),  
(16, 18),  
(17,19),  
(18,20),  
(19,21),  
(20,22),  
(21,23);
```

# Interogari SELECT

1. Afisati toate tarile din baza de date.

```
SELECT nume_tara  
FROM Tari;
```

	nume_tara
1	Italia
2	Franta
3	Romania
4	SUA
5	Spain
6	Marea Britanie
7	Germania
8	Japonia
9	China
10	Rusia

2. Afisati toate zborurile care au durata mai mare de 120 de minute.

```
SELECT*  
FROM Zboruri  
WHERE durata > 120
```

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport
1	13	2	12:30:00.0000000	2023-12-11	180	5
2	15	4	08:00:00.0000000	2023-12-13	240	7
3	16	5	10:15:00.0000000	2023-12-14	150	8
4	17	6	15:30:00.0000000	2023-12-15	180	9
5	19	8	19:00:00.0000000	2023-12-17	360	11
6	20	9	02:00:00.0000000	2023-12-18	540	12
7	21	10	05:30:00.0000000	2023-12-19	480	13

3. Afișați numele și prenumele calatorilor a căror număr de telefon se termină într-o cifră pară.

```
SELECT Nume, Prenume
FROM Calatori
WHERE Telefon LIKE '%0' or Telefon LIKE '%2' or Telefon LIKE '%4' or
Telefon LIKE '%6' or Telefon LIKE '%8'
```

	Nume	Prenume	telefon
1	Popescu	Ana	0721123456
2	Ionescu	Alex	0723123456
3	Popa	Maria	0731123456
4	Vasilescu	Alexandra	0741123456
5	Popescu	Mihai	0726123456
6	Dumitru	Diana	0751123456
7	Radulescu	Andrei	0729123456
8	Georgescu	Elena	0742123456
9	Petrescu	Marian	0732123456
10	Stefanescu	Alina	0761123456

4. Afișați avioanele a caror model începe cu 'A'.

```
SELECT*
FROM Avioane
WHERE model like 'A%'
```

	ID_avion	model	capacitate	ID_companie
1	1	Boeing 747	416	12
2	3	Boeing 737	215	14
3	5	Boeing 777	396	16
4	8	Bombardier CRJ700	78	19

5. Afișați zborurile și portile de imbarcare pentru fiecare.

```
SELECT Zboruri.ID_zbor, Zboruri.data_plecare, Zboruri.data_plecare, Zboruri.ora,
Zboruri.durata, Gates.numar
FROM Gates
INNER JOIN ZboruriGates ON Gates.ID_gate = ZboruriGates.ID_gate
INNER JOIN Zboruri ON ZboruriGates.ID_zbor = Zboruri.ID_zbor
```

	ID_zbor	data_plecare	data_plecare	ora	durata	Numar terminal
1	12	2023-12-10	2023-12-10	09:00:00.0000000	120	1
2	12	2023-12-10	2023-12-10	09:00:00.0000000	120	11
3	13	2023-12-11	2023-12-11	12:30:00.0000000	180	2
4	14	2023-12-12	2023-12-12	14:45:00.0000000	90	3
5	15	2023-12-13	2023-12-13	08:00:00.0000000	240	4
6	16	2023-12-14	2023-12-14	10:15:00.0000000	150	5
7	17	2023-12-15	2023-12-15	15:30:00.0000000	180	6
8	18	2023-12-16	2023-12-16	11:00:00.0000000	120	7
9	19	2023-12-17	2023-12-17	19:00:00.0000000	360	8
10	20	2023-12-18	2023-12-18	02:00:00.0000000	540	9
11	21	2023-12-19	2023-12-19	05:30:00.0000000	480	10

6. Afisati toate zborurile impreuna cu numele companiei si destinatia (numele tarii si aeroportului).

```

SELECT Companii.nume_companie AS 'Nume companie', Zboruri.ID_zbor, Zboruri.ora,
Zboruri.data_plecare, Zboruri.durata,
Aeroporturi.nume AS 'Nume aeroport sosire', Tari.nume_tara AS 'Nume tara'
FROM Zboruri
INNER JOIN Aeroporturi ON Zboruri.ID_aeroport = Aeroporturi.ID_aeroport
INNER JOIN Locatii ON Aeroporturi.ID_oras = Locatii.ID_locatie
INNER JOIN Tari ON Tari.ID_tara = Locatii.ID_tara
INNER JOIN Avioane ON Zboruri.ID_avion = Avioane.ID_avion
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie;

```

	Nume companie	ID_zbor	ora	data_plecare	durata	Nume aeroport sosire	Nume tara
1	Blue Air	12	09:00:00.0000000	2023-12-10	120	Charles de Gaulle	Franta
2	American Airlines	13	12:30:00.0000000	2023-12-11	180	Marco Polo	Italia
3	Air France	14	14:45:00.0000000	2023-12-12	90	Henri Coanda	Romania
4	Alitalia	15	08:00:00.0000000	2023-12-13	240	JFK	SUA
5	Iberia	16	10:15:00.0000000	2023-12-14	150	Fiumicino	Italia
6	British Airways	17	15:30:00.0000000	2023-12-15	180	El Prat	Spain
7	Lufthansa	18	11:00:00.0000000	2023-12-16	120	Heathrow	Marea Britanie
8	ANA	19	19:00:00.0000000	2023-12-17	360	Tegel	Germania
9	Air China	20	02:00:00.0000000	2023-12-18	540	Narita	Japonia
10	Aeroflot	21	05:30:00.0000000	2023-12-19	480	Beijing Capital	China



7. Afisati numele si prenumele calatorilor care au achizitionat bilet de la compania 'Air France'.

```
SELECT Calatori.nume, Calatori.prenume, Companii.nume_companie
FROM Calatori
INNER JOIN Bilete ON Calatori.ID_calator = Bilete.ID_calator
INNER JOIN BiletePerZbor ON BiletePerZbor.id_bilet = Bilete.ID_bilet
INNER JOIN Zboruri ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
INNER JOIN Avioane ON Zboruri.ID_avion = Avioane.ID_avion
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.nume_companie = 'Air France'
```

	nume	prenume	nume_companie
1	Popescu	Ana	Air France
2	Ionescu	Alex	Air France
3	Popa	Maria	Air France

8. Afisati numele si prenumele angajatilor care lucreaza la departamentul 'Mentenanata'.

```
SELECT Staff.nume, Staff.prenume, Departamente.denumire
FROM Staff
INNER JOIN Repartizare_functii ON Staff.ID_staff = Repartizare_functii.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN Departamente ON Functii.ID_departament = Departamente.ID_departament
WHERE Departamente.denumire = 'Mentenanata'
```

	nume	prenume	denumire
1	Stanescu	Elena	Mentenanata
2	Popa	Maria	Mentenanata

9. Afisati toate zborurile din luna decembrie in ordine crescatoare in functie de zi, impreuna cu taskurile necesare fiecaruia.

```
SELECT DAY(Zboruri.data_plecare) AS 'Zi plecare', Zboruri.durata, Zboruri.ora, Tasks.nume
AS 'Denumire task'
FROM Zboruri
INNER JOIN TaskspersZbor ON TaskspersZbor.ID_zbor = Zboruri.ID_zbor
INNER JOIN Tasks ON TaskspersZbor.ID_task = Tasks.ID_task
WHERE MONTH(Zboruri.data_plecare) = 12
ORDER BY DAY(Zboruri.data_plecare) asc;
```

	Zi plecare	durata	ora	Denumire task
1	10	120	09:00:00.0000000	Asigurarea siguranței pasagerilor și a aeronavei
2	11	180	12:30:00.0000000	Încarcarea și descarcarea bagajelor
3	12	90	14:45:00.0000000	Aprovizionarea aeronavei cu combustibil și alte ...
4	13	240	08:00:00.0000000	Îmbarcarea pasagerilor
5	14	150	10:15:00.0000000	Verificarea biletelor și a documentelor pasageril...
6	15	180	15:30:00.0000000	Verificarea tehnica a aeronavei înainte de decol...
7	16	120	11:00:00.0000000	Verificarea tehnica a aeronavei înainte de decol...
8	17	360	19:00:00.0000000	Asigurarea siguranței pasagerilor și a aeronavei
9	18	540	02:00:00.0000000	Încarcarea și descarcarea bagajelor
10	19	480	05:30:00.0000000	Aprovizionarea aeronavei cu combustibil și alte ...

10. Afisati numarul de bilete vandute de compania 'Alitalia'.

```
SELECT COUNT(Companii.nume_companie)
FROM Calatori
INNER JOIN Bilete ON Calatori.ID_calator = Bilete.ID_calator
INNER JOIN BiletePerZbor ON BiletePerZbor.id_bilet = Bilete.ID_bilet
INNER JOIN Zboruri ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
INNER JOIN Avioane ON Zboruri.ID_avion = Avioane.ID_avion
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.nume_companie = 'Alitalia'
```

	Bilete vandute de Alitalia
1	4

11. Afisati media costului unui zbor spre 'Roma'.

```
SELECT AVG(Bilete.pret_bilet) AS 'Media pretului unui bilet spre Paris'
FROM Calatori
INNER JOIN Bilete ON Calatori.ID_calator = Bilete.ID_calator
INNER JOIN BiletePerZbor ON BiletePerZbor.id_bilet = Bilete.ID_bilet
INNER JOIN Zboruri ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
INNER JOIN Aeroporturi ON Zboruri.ID_aeroport = Aeroporturi.ID_aeroport
INNER JOIN Locatii ON Aeroporturi.ID_oras = Locatii.ID_locatie
WHERE Locatii.nume_locatie = 'Roma'
```

	Media cost bilet spre Paris
1	966

12. Afisati numarul de bilete vandute pentru fiecare clasa.

```
SELECT C.nume_clasa, COUNT(B.ID_bilet) AS numar_bilete
FROM Clase C
LEFT JOIN Bilete B ON C.ID_clasa = B.ID_clasa
GROUP BY C.nume_clasa;
```

	nume_clasa	numar_bilete
1	BUSINESS	4
2	ECONOMIC	6

13. Afisati numarul de orase pentru fiecare tara.

```
SELECT T.nume_tara, COUNT(L.ID_locatie) AS numar_orase
FROM Tari T
LEFT JOIN Locatii L ON T.ID_tara = L.ID_tara
GROUP BY T.nume_tara;
```

	nume_tara	numar_orase
1	China	1
2	Franta	1
3	Germania	1
4	Italia	2
5	Japonia	1
6	Marea Britanie	1
7	Romania	1
8	Rusia	1
9	Spain	1
10	SUA	1

14. Afisati primele 5 cele mai cautate destinatii (aeroporturi).

```
SELECT TOP 5 a.nume, COUNT(*) AS numar_zboruri
FROM Zboruri z
JOIN Aeroporturi a ON z.ID_aeroport = a.ID_aeroport
GROUP BY a.nume
ORDER BY numar_zboruri DESC
```

	nume	numar_zboruri
1	Henri Coanda	1
2	Heathrow	1
3	Fiumicino	1
4	El Prat	1
5	Charles de Gaulle	1

15. Afisati suma incasata pentru fiecare zbor in ordine descrescatoare.

```
SELECT z.ID_zbor, SUM(b.pret_bilet) AS 'Suma incasata'
FROM Zboruri z JOIN BiletePerZbor bpz ON z.ID_zbor = bpz.id_zbor
JOIN Bilete b ON bpz.id_bilet = b.ID_bilet
GROUP BY z.ID_zbor
ORDER BY 'Suma incasata' DESC;
```

	ID_zbor	Suma incasata
1	15	4500
2	16	2900
3	14	2300

16. Afisati tarile care au mai mult de un oras ca destinatie.

```
SELECT T.nume_tara, COUNT(L.ID_locatie) AS 'Numar Orașe'
FROM Tari T
JOIN Locatii L ON T.ID_tara = L.ID_tara
GROUP BY T.nume_tara
HAVING COUNT(L.ID_locatie) > 1;
```

	nume_tara	Numar Orașe
1	Italia	2

17. Afisati numele si prenumele calatorilor care au cumparat cel putin un bilet si au platit mai mult de 500 de unitati monetare.

```
SELECT CA.nume, CA.prenume
FROM Calatori CA
JOIN Bilete B ON CA.ID_calator = B.ID_calator
GROUP BY CA.nume, CA.prenume
HAVING SUM(B.pret_bilet) > 500;
```

	nume	prenume
1	Ionescu	Alex
2	Vasilescu	Alexandra
3	Stefanescu	Alina
4	Radulescu	Andrei

18. Afisati companiile care au un singur zbor.

```
SELECT C.nume_companie, COUNT(Z.ID_zbor) AS NumarZboruri
FROM Companii C
JOIN Avioane A ON C.ID_companie = A.ID_companie
JOIN Zboruri Z ON A.ID_avion = Z.ID_avion
GROUP BY C.nume_companie
HAVING COUNT(Z.ID_zbor) < 2;
```

	nume_companie	NumarZboruri
1	Aeroflot	1
2	Air China	1
3	Air France	1
4	Alitalia	1
5	American Airlines	1
6	ANA	1
7	Blue Air	1
8	British Airways	1
9	Iberia	1
10	Lufthansa	1

19. Afisati calatorii care au cumparat mai mult de un bilet.

```
SELECT Calatori.nume, Calatori.prenume, COUNT(Bilete.ID_bilet) as total_bilete
FROM Calatori
JOIN Bilete ON Calatori.ID_calator = Bilete.ID_calator
GROUP BY Calatori.nume, Calatori.prenume
HAVING COUNT(Bilete.ID_bilet) > 1;
```

	nume	prenume	total_bilete
1	Popescu	Ana	2

20. Afisati toate zborurile de la data 2023-12-10 in ordine crescatoare in functie de ora.

```
SELECT Zboruri.ID_zbor, Zboruri.ora, Zboruri.durata, Aeroporturi.nume,
Locatii.nume_locatie
FROM Zboruri
INNER JOIN Aeroporturi ON Zboruri.ID_aeroport = Aeroporturi.ID_aeroport
INNER JOIN Locatii ON Aeroporturi.ID_oras = Locatii.ID_locatie
WHERE Zboruri.data_plecure = '2023-12-10'
ORDER BY Zboruri.ora ASC;
```

	ID_zbor	ora	durata	nume	nume_locatie
1	12	09:00:00.0000000	120	Charles de Gaulle	Paris

21. Afisati companiile care detin avioane cu o capacitate mai mare sau egala cu 200.

```
SELECT Companii.nume_comanie
FROM Companii
INNER JOIN Avioane ON Companii.ID_comanie = Avioane.ID_comanie
WHERE Avioane.capacitate >= 200
GROUP BY Companii.nume_comanie;
```

	nume_comanie
1	Air France
2	American Airlines
3	Blue Air
4	British Airways
5	Iberia

22. Afisati departamentele care ofera un salariu mai mare de 2000.

```
SELECT Departamente.denumire
FROM Departamente
INNER JOIN Functii ON Departamente.ID_departament = Functii.ID_departament
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN Staff ON Staff.ID_staff = Repartizare_functii.ID_staff
WHERE Functii.salariu_brut > 2000
GROUP BY Departamente.denumire
```

	denumire
1	Aprovizionare
2	Financiar-contabil
3	Marketing
4	Mentenanata
5	Operatiuni aeroportuare
6	Planificare si dezvoltare
7	Siguranta
8	Tehnologia informatiei

23. Afisati companiile care detin avioane de tipul 'Airbus A340' si cate avioane de acest tip detin.

```
SELECT Companii.nume_comanie, COUNT(Avioane.ID_avion) as numar_avioane
FROM Companii
INNER JOIN Avioane ON Companii.ID_comanie = Avioane.ID_comanie
WHERE Avioane.model = 'Airbus A340'
GROUP BY Companii.nume_comanie;
```

	nume_companie	numar_avioane
1	Air France	1
2	Lufthansa	1

24. Afisati companiile care au zboruri cu o durata mai mare de 100 de minute.

```
SELECT Companii.nume_companie
FROM Companii
INNER JOIN Avioane ON Avioane.ID_companie = Companii.ID_companie
INNER JOIN Zboruri ON Zboruri.ID_avion = Avioane.ID_avion
WHERE Zboruri.durata > 100
Group BY Companii.nume_companie
```

	nume_companie
1	Aeroflot
2	Air China
3	Alitalia
4	American Airlines
5	ANA
6	Blue Air
7	British Airways
8	Iberia
9	Lufthansa

25. Afisati angajatii care platesc Contributie la CAS (impozit).

```
SELECT Staff.nume, Staff.preume
FROM Staff
INNER JOIN Contracte ON Contracte.ID_angajat = Staff.ID_staff
INNER JOIN ContracteImpozite ON ContracteImpozite.ID_contract = Contracte.ID_contract
INNER JOIN Impozite ON Impozite.ID_impozit = Contracte.ID_contract
WHERE Impozite.nume = 'Contributie la CAS'
```

	nume	preume
1	Popescu	Mihai

26. Afisati toate taskurile de la departamentul 'Siguranta'.

```
select Departamente.denumire AS Departament, Tasks.nume AS Task
```

```

FROM Tasks
INNER JOIN StaffTasks ON StaffTasks.ID_task = Tasks.ID_task
INNER JOIN Staff ON StaffTasks.ID_staff = Staff.ID_staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Departamente.denumire = 'Siguranta'

```

	Departament	Task
1	Siguranta	Asigurarea siguran?ei pasagerilor ?i a aeronavei
2	Siguranta	Încarcarea ?i descarcarea bagajelor

27. Afisati toate biletele care au ca destinatie Italia.

```

SELECT Tari.num_e_tara, Bilete.ID_bilet, Bilete.pret_bilet
FROM Tari
INNER JOIN Locatii ON Locatii.ID_tara = Tari.ID_tara
INNER JOIN Aeroporturi ON Aeroporturi.ID_oras = Locatii.ID_locatie
INNER JOIN Zboruri ON Zboruri.ID_aeroport = Aeroporturi.ID_aeroport
INNER JOIN BiletePerZbor ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
INNER JOIN Bilete ON Bilete.ID_bilet = BiletePerZbor.id_bilet
WHERE Tari.num_e_tara = 'Italia'

```

	num_e_tara	ID_bilet	pret_bilet
1	Italia	15	300
2	Italia	16	400
3	Italia	19	300
4	Italia	20	300

28. Afisati toti angajatii care au ca impozit 'Contributia la Fondul de risc si accidente de munca - clasa de risc II'.

```

SELECT Staff.*, Impozite.num_e AS Impozit
FROM IMPOZITE
INNER JOIN ContracteImpozite ON ContracteImpozite.ID_impozit = Impozite.ID_impozit
INNER JOIN Contracte ON Contracte.ID_contract = ContracteImpozite.ID_contract
INNER JOIN Staff ON Staff.ID_staff = Contracte.ID_angajat
WHERE Impozite.num_e = 'Contributia la Fondul de risc si accidente de munca - clasa de risc II'

```

	ID_staff	num_e	prenum_e	email	telefon	Impozit
1	22	Popescu	Mihai	Popescu.M@aero.com	0766777888	Contributia la Fondul de risc si accidente de munca - clasa de risc II



29. Afisati toate zborurile care au ca destinatie Italia si sunt organizate de compania Alitalia.

```
SELECT Zboruri.*, Companii.nume_companie
FROM Companii
INNER JOIN Avioane ON Avioane.ID_companie = Companii.ID_companie
INNER JOIN Zboruri ON Zboruri.ID_avion = Avioane.ID_avion
INNER JOIN Aeroporturi ON Aeroporturi.ID_aeroport = Zboruri.ID_aeroport
INNER JOIN Locatii ON Locatii.ID_locatie = Aeroporturi.ID_oras
INNER JOIN Tari ON Tari.ID_tara = Locatii.ID_tara
WHERE Tari.nume_tara = 'Italia' and Companii.nume_companie = 'Alitalia'
```

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport	nume_companie
1	66	4	08:00:00.0000000	2023-12-13	240	37	Alitalia

30. Afisati angajatii care au mai mult de un task.

```
SELECT STAFF.ID_staff, Staff.nume, Staff.prenume
FROM Staff
INNER JOIN StaffTasks ON StaffTasks.ID_staff = Staff.ID_staff
INNER JOIN Tasks ON StaffTasks.ID_task = Tasks.ID_task
GROUP BY Staff.ID_staff, Staff.nume, Staff.prenume
HAVING COUNT(Tasks.ID_task) > 1
```

	ID_staff	nume	prenume
1	17	Dumitrescu	Ion
2	20	Gheorghe	Andreea

31. Afisati numele companiilor impreuna cu numarul de avioane detinute pentru zboruri in aeroport.

```
SELECT Companii.nume_companie, COUNT (Avioane.ID_avion) AS NR
FROM Companii
INNER JOIN Avioane ON Avioane.ID_companie = Companii.ID_companie
GROUP BY Companii.nume_companie
```

	nume_companie	NR
1	Aeroflot	1
2	Air China	1
3	Air France	2
4	Alitalia	1
5	American Airlines	1
6	ANA	1
7	Blue Air	2
8	British Airways	1
9	Iberia	1
10	Lufthansa	2

32. Afisati numele si prenumele anagajatilor care platesc Contributie la CAS.

```
SELECT Staff.nume , Staff.prenume, Impozite.nume
FROM Impozite
INNER JOIN ContracteImpozite ON ContracteImpozite.ID_impozit = Impozite.ID_impozit
INNER JOIN Contracte ON Contracte.ID_contract = ContracteImpozite.ID_contract
INNER JOIN Staff ON Staff.ID_staff = Contracte.ID_angajat
GROUP BY Staff.nume, Staff.prenume, Impozite.nume
HAVING Impozite.nume = 'Contributie la CAS'
```

	nume	prenume	nume
1	Petrescu	Maria	Contributie la CAS
2	Popescu	Alex	Contributie la CAS

33. Afisati numele si prenumele calatorilor impreuna cu numele companieie la care au cumparat bilet.

```
SELECT Calatori.nume, Calatori.prenume, Companii.nume_companie
FROM Calatori
INNER JOIN Bilete ON Calatori.ID_calator = Bilete.ID_calator
INNER JOIN BiletePerZbor ON Bilete.ID_bilet = BiletePerZbor.id_bilet
INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
INNER JOIN Avioane ON Avioane.ID_avion = Zboruri.ID_avion
INNER JOIN Companii ON Companii.ID_companie =Avioane.ID_companie
```

	nume	prenume	nume_companie
1	Popescu	Ana	American Airlines
2	Ionescu	Alex	American Airlines
3	Popa	Maria	American Airlines
4	Vasilescu	Alexandra	Air France
5	Popescu	Mihai	Air France
6	Dumitru	Diana	Air France
7	Radulescu	Andrei	Air France

34. Afisati clasele disponibile pentru fiecare companie.

```
SELECT DISTINCT Clase.NUME_CLASA, Companii.nume_companie
FROM Clase
INNER JOIN Bilete ON Bilete.ID_clasa = Clase.ID_clasa
INNER JOIN BiletePerZbor ON BiletePerZbor.id_bilet = Bilete.ID_bilet
INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
INNER JOIN Avioane ON Avioane.ID_avion = Zboruri.ID_avion
INNER JOIN Companii ON Companii.ID_companie = Avioane.ID_companie
```

	NUME_CLASA	nume_companie
1	ECONOMIC	Air France
2	ECONOMIC	American Airlines

35. Afisati numarul de taskuri per departament.

```
SELECT Departamente.denumire AS Departament, COUNT(Tasks.ID_task) AS 'Nr tasks'
FROM Departamente
INNER JOIN Functii ON Functii.ID_departament = Departamente.ID_departament
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN Staff ON Staff.ID_staff = Repartizare_functii.ID_staff
INNER JOIN StaffTasks ON Staff.ID_staff = StaffTasks.ID_staff
INNER JOIN Tasks ON Tasks.ID_task = StaffTasks.ID_task
GROUP BY Departamente.denumire
```

	Departament	Nr tasks
1	Financiar-contabil	2
2	Mentenananta	2
3	Operatiuni aeroportuare	2
4	Siguranta	2
5	Tehnologia informatiei	1

36. Afisati destinatiile pe care le poate oferi fiecare companie.

```
SELECT DISTINCT Companii.nume_companie, Tari.nume_tara
FROM Companii
INNER JOIN Avioane ON Avioane.ID_companie = Companii.ID_companie
INNER JOIN Zboruri ON Zboruri.ID_avion = Avioane.ID_avion
INNER JOIN Aeroporturi ON Aeroporturi.ID_aeroport = Zboruri.ID_aeroport
INNER JOIN Locatii ON Locatii.ID_locatie = Aeroporturi.ID_oras
INNER JOIN Tari ON Tari.ID_tara = Locatii.ID_tara
```

	nume_companie	nume_tara
1	Air Aeroflot	Rusia
2	Air China	China
3	Air France	SUA
4	Alitalia	Italia
5	American Airlines	Italia
6	American Airlines	Romania
7	ANA	Japonia
8	Blue Air	Italia
9	British Airways	Marea Britanie
10	Iberia	Spain
11	Lufthansa	Germania

37. Afisati numele si prenumele calatorului cel mai scump bilet.

```
SELECT TOP 1 c.nume, c.prenume, b.pret_bilet
FROM Calatori c
JOIN Bilete b ON c.ID_calator = b.ID_calator
ORDER BY b.pret_bilet DESC;
```

	nume	prenume	pret_bilet
1	Ionescu	Alex	2000

38. Afisati cel mai lung zbor pana in Italia.

```
SELECT TOP 1 z.ID_zbor, z.durata, t.nume_tara
FROM Zboruri z
JOIN Aeroporturi a ON z.ID_aeroport = a.ID_aeroport
JOIN Locatii l ON a.ID_oras = l.ID_locatie
JOIN Tari t ON l.ID_tara = t.ID_tara
WHERE t.nume_tara = 'Italia'
```

ORDER BY z.durata DESC;

	ID_zbor	durata	nume_tara
1	66	240	Italia

39. Afisati cel mai mare salariu brut in functie de functie .

```
SELECT MAX(salariu_brut) AS salariu_maxim, nume AS nume_functie
FROM Functii
GROUP BY nume;
```

	salariu_maxim	nume_functie
1	3000	Agent de securitate
2	2700	Casier
3	8000	Contabil
4	5500	Inginer aprovizionare
5	12000	Manager
6	7260	Manager marketing
7	5500	Manager resurse umane
8	5000	Programator
9	4200	Responsabil mentenanta
10	4500	Supraveghetor si ajutor imbarcare
11	4000	Tehnician

40. Afisati calatoriile impreuna cu pretul biletului cumparat si cu detaliile zborului acestora.

```
SELECT Bilete.pret_bilet, Calatori.nume, Calatori.prenume, Zboruri.ora, Zboruri.durata
FROM Bilete
JOIN Calatori ON Bilete.ID_calator = Calatori.ID_calator
JOIN BiletePerZbor ON Bilete.ID_bilet = BiletePerZbor.ID_bilet
JOIN Zboruri ON BiletePerZbor.ID_zbor = Zboruri.ID_zbor;
```

	pret_bilet	nume	prenume	ora	durata
1	100	Popescu	Ana	12:30:00.00000000	180
2	2000	Ionescu	Alex	12:30:00.00000000	180
3	200	Popa	Maria	12:30:00.00000000	180
4	1971	Vasilescu	Alexandra	14:45:00.00000000	90
5	292	Popescu	Mihai	14:45:00.00000000	90
6	192	Dumitru	Diana	14:45:00.00000000	90
7	1971	Radulescu	Andrei	14:45:00.00000000	90

41. Afisati calatorii impreunca cu clasa la care au achizitionat bilet.

```
SELECT Bilete.ID_bilet, Calatori.nume, Calatori.prenume, Clase.NUME_CLASA
FROM Bilete
JOIN Calatori ON Bilete.ID_calator = Calatori.ID_calator
JOIN Clase ON Bilete.ID_clasa = Clase.ID_clasa;
```

	ID_bilet	nume	prenume	NUME_CLASA
1	8	Popescu	Ana	ECONOMIC
2	9	Ionescu	Alex	ECONOMIC
3	10	Popa	Maria	ECONOMIC
4	11	Vasilescu	Alexandra	ECONOMIC
5	12	Popescu	Mihai	ECONOMIC
6	13	Dumitru	Diana	ECONOMIC
7	14	Radulescu	Andrei	ECONOMIC
8	18	Popescu	Ana	ECONOMIC

42. Afisati numele si prenumele angajatilor de la departamentul 'Siguranta' impreuna cu salariul brut al acestora.

```
SELECT Staff.nume, Staff.prenume, Functii.salariu_brut, Departamente.denumire
FROM Staff
JOIN Repartizare_functii ON Staff.ID_staff = Repartizare_functii.ID_staff
JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
JOIN Departamente ON Functii.ID_departament = Departamente.ID_departament
WHERE Departamente.denumire = 'Siguranta';
```

	nume	prenume	salariu_brut	denumire
1	Gheorghe	Andreea	3000	Siguranta

# Interogari folosind CTE-uri

1. Afisati toti calatorii care au achizitionat bilet la clasa BUSINESS.

```
WITH CTEBilete AS (  
    SELECT Bilete.*, Clase.NUME_CLASA  
    FROM Bilete  
    INNER JOIN Clase ON Clase.ID_clasa = Bilete.ID_clasa  
    WHERE Clase.NUME_CLASA = 'BUSINESS'  
)  
SELECT Calatori.*  
FROM Calatori  
INNER JOIN Bilete ON Bilete.ID_calator = Calatori.ID_calator;
```

	ID_calator	CNP	nume	prenume	sex	email	telefon	adresa
1	66	1234567890123	Popescu	Ana	FEMININ	ana.p@gmail.com	0721123456	Str. Libertatii nr. 10
2	67	1234567890124	Ionescu	Alex	MASCULIN	alex.i@gmail.com	0723123456	Str. Independentei nr. 20
3	68	1234567890125	Popa	Maria	FEMININ	maria.p@gmail.com	0731123456	Str. Unirii nr. 15
4	69	1234567890126	Vasilescu	Alexandra	FEMININ	alex.v@gmail.com	0741123456	Str. Mihai Viteazul nr. 5
5	70	1234567890127	Popescu	Mihai	MASCULIN	mihai.p@gmail.com	0726123456	Str. Republicii nr. 7
6	71	1234567890128	Dumitru	Diana	FEMININ	diana.d@gmail.com	0751123456	Str. Ion Creanga nr. 12
7	72	1234567890129	Radulescu	Andrei	MASCULIN	andrei.r@gmail.com	0729123456	Str. Gheorghe Doja nr. 14
8	73	1234567890130	Georgescu	Elena	FEMININ	elena.g@gmail.com	0742123456	Str. Tudor Vladimirescu nr. 17
9	74	1234567890131	Petrescu	Marian	MASCULIN	marian.p@gmail.com	0732123456	Str. Vasile Alecsandri nr. 19
10	75	1234567890132	Stefanescu	Alina	FEMININ	alina.s@gmail.com	0761123456	Str. Stefan cel Mare nr. 9
11	66	1234567890123	Popescu	Ana	FEMININ	ana.p@gmail.com	0721123456	Str. Libertatii nr. 10

2. Afisati toate zborurile care au ca destinatie orasul Barcelona.

```
WITH OrasDestinatie AS (  
    SELECT ID_locatie  
    FROM Locatii  
    WHERE nume_locatie = 'Barcelona'  
)  
SELECT Zboruri.*  
FROM Zboruri  
INNER JOIN Aeroporturi ON Aeroporturi.ID_aeroport = Zboruri.ID_aeroport  
INNER JOIN OrasDestinatie ON OrasDestinatie.ID_locatie = Aeroporturi.ID_oras;
```

	ID_zbor	ID_avion	ora	data_plecure	durata	ID_aeroport
1	17	6	15:30:00.0000000	2023-12-15	180	9

3. Afisati toate zborurile efectuate cu acelasi tip de avion.

```
WITH ZboruriAvion AS (  
    SELECT z.ID_zbor, a.model, z.data_plecure, z.ora  
    FROM Zboruri z  
    JOIN Avioane a ON z.ID_avion = a.ID_avion  
)  
SELECT * FROM ZboruriAvion WHERE model = 'Boeing 747';
```

	ID_zbor	model	data_plecure	ora
1	12	Boeing 747	2023-12-10	09:00:00.0000000

4. Afisati toate zborurile impreuna cu numele aeroportului destinatie, modelul avionului si numele companiei.

```
WITH InfoZboruri AS (  
    SELECT Z.ID_zbor, Z.data_plecure, Z.ora, A1.num AS aeroport_destinatie, C.num_companie,  
    A.model  
    FROM Zboruri Z  
    JOIN Aeroporturi A1 ON Z.ID_aeroport=A1.ID_aeroport  
    JOIN Avioane A ON Z.ID_avion=A.ID_avion  
    JOIN Companii C ON A.ID_companie=C.ID_companie)  
SELECT *  
FROM InfoZboruri;
```

	ID_zbor	data_plecure	ora	aeroport_destinatie	nume_companie	model
1	12	2023-12-10	09:00:00.0000000	Charles de Gaulle	Blue Air	Boeing 747
2	13	2023-12-11	12:30:00.0000000	Marco Polo	American Airlines	Airbus A380
3	14	2023-12-12	14:45:00.0000000	Henri Coanda	Air France	Boeing 737
4	15	2023-12-13	08:00:00.0000000	JFK	Alitalia	Airbus A320
5	16	2023-12-14	10:15:00.0000000	Fiumicino	Iberia	Boeing 777
6	17	2023-12-15	15:30:00.0000000	El Prat	British Airways	Airbus A330
7	18	2023-12-16	11:00:00.0000000	Heathrow	Lufthansa	Embraer E190
8	19	2023-12-17	19:00:00.0000000	Tegel	ANA	Bombardier CRJ700
9	20	2023-12-18	02:00:00.0000000	Narita	Air China	ATR 72
10	21	2023-12-19	05:30:00.0000000	Beijing Capital	Aeroflot	Sukhoi Superjet 100



5. Afisati numarul maxim de bilete vandute pentru fiecare luna.\*

```
WITH CTEmax AS
(
    SELECT MONTH(Zboruri.data_plecure) AS Luna, COUNT(Bilete.ID_bilet) AS Maxim
    FROM Zboruri
    INNER JOIN BiletePerZbor ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
    INNER JOIN Bilete ON BiletePerZbor.id_bilet = Bilete.ID_bilet
    GROUP BY MONTH(Zboruri.data_plecure)
)
SELECT Luna, Maxim
FROM CTEmax
```

	Luna	Maxim
1	11	1
2	12	10

6. Afisati numarul de calatori pentru fiecare zbor.

```
WITH CTE_NumarCalatoriPerZbor AS (
SELECT Z.ID_zbor, COUNT(B.ID_bilet) AS NumarCalatori
FROM Zboruri Z
INNER JOIN BiletePerZbor BPZ ON Z.ID_zbor = BPZ.id_zbor
INNER JOIN Bilete B ON BPZ.id_bilet = B.ID_bilet
GROUP BY Z.ID_zbor
)
SELECT * FROM CTE_NumarCalatoriPerZbor;
```

	ID_zbor	NumarCalatori
1	14	3
2	15	4
3	16	3
4	21	1

7. Afisati cel mai lung zbor cu cel mai mic pret.

```
WITH CTE AS (
SELECT TOP 1 WITH TIES z.ID_zbor, z.durata, b.pret_bilet
FROM Zboruri z
JOIN BiletePerZbor bz ON bz.id_zbor = z.ID_zbor
JOIN Bilete b ON b.ID_bilet = bz.id_bilet
```

```

ORDER BY z.durata DESC, b.pret_bilet ASC
)
SELECT z.ID_zbor, a.num AS aeroport, l.num_locatie AS destinatie, c.num_companie,
av.model AS model_avion, z.ora, z.data_plecure, z.durata, b.pret_bilet
FROM Zboruri z
JOIN CTE ON CTE.ID_zbor = z.ID_zbor
JOIN Aeroporturi a ON a.ID_aeroport = z.ID_aeroport
JOIN Locatii l ON l.ID_locatie = a.ID_oras
JOIN Avioane av ON av.ID_avion = z.ID_avion
JOIN Companii c ON c.ID_companie = av.ID_companie
JOIN BiletePerZbor bz ON bz.id_zbor = z.ID_zbor
JOIN Bilete b ON b.ID_bilet = bz.id_bilet;

```

	ID_zbor	aeroport	destinatie	num_companie	model_avion	ora	data_plecure	durata	pret_bilet
1	21	Beijing Capital	Beijing	Aeroflot	Sukhoi Superjet 100	09:00:00.0000000	2023-11-19	480	300

## Actualizari de date (UPDATE)

1. Schimbarea numelui de familie a angajatei Ionescu Maria in urma casatoriei.

UPDATE Staff

SET nume = 'Petrescu'

WHERE nume = 'Ionescu' and prenume = 'Maria'

	ID_staff	nume	prenume	email	telefon
1	13	Popescu	Alex	alex.popescu@example.com	0744555666
2	14	Ionescu	Maria	maria.ionescu@example.com	0733666777
3	15	Georgescu	Andrei	andrei.georgescu@example.com	0711223344
4	16	Stanescu	Elena	elena.stanescu@example.com	0722333444
5	17	Dumitrescu	Ion	ion.dumitrescu@example.com	0755667788
6	18	Popa	Maria	maria.popa@example.com	0766777888
7	19	Mihai	Vlad	vlad.mihai@example.com	0711223344
8	20	Gheorghe	Andreea	andreea.gheorghe@example.com	0744555666
9	21	Radu	Ioana	ioana.radu@example.com	0755667788
10	22	Popescu	Mihai	mihai.popescu@example.com	0766777888

	ID_staff	nume	prenume	email	telefon
1	13	Popescu	Alex	alex.popescu@example.com	0744555666
2	14	Petrescu	Maria	maria.ionescu@example.com	0733666777
3	15	Georgescu	Andrei	andrei.georgescu@example.com	0711223344
4	16	Stanescu	Elena	elena.stanescu@example.com	0722333444
5	17	Dumitrescu	Ion	ion.dumitrescu@example.com	0755667788
6	18	Popa	Maria	maria.popa@example.com	0766777888
7	19	Mihai	Vlad	vlad.mihai@example.com	0711223344
8	20	Gheorghe	Andreea	andreea.gheorghe@example.com	0744555666
9	21	Radu	Ioana	ioana.radu@example.com	0755667788
10	22	Popescu	Mihai	mihai.popescu@example.com	0766777888

2. Se doreste adaugarea unui discount la biletele care sunt vandute de compania 'Alitalia.

```
DECLARE @newdiscount REAL;
SET @newdiscount = 0.05;
UPDATE Bilete
SET pret_bilet -= @newdiscount*pret_bilet
FROM Bilete
INNER JOIN BiletePerZbor ON Bilete.ID_bilet = BiletePerZbor.id_bilet
INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
INNER JOIN Avioane ON Avioane.ID_avion = Zboruri.ID_avion
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.nume_companie = 'Alitalia'
SELECT @newdiscount
```

	nume_companie	pret_bilet
1	Alitalia	2075
2	Alitalia	308
3	Alitalia	203
4	Alitalia	2075

	nume_companie	pret_bilet
1	Alitalia	1971
2	Alitalia	292
3	Alitalia	192
4	Alitalia	1971

3. Actualizati adresa de email a tuturor anagajatilor cu adrese de email create de catre aeroport.\*

```
select * from Staff
DECLARE @Counter INT ,
@MaxId INT,
@email as varchar(50);
SELECT @Counter = min(Staff.ID_staff) ,
@MaxId = max(Staff.ID_staff)
FROM Staff
WHILE( @Counter <= @MaxId)
BEGIN
    update S
    set S.email = S.nume+ '.' + SUBSTRING(S.prenume,1,1) + '@aero.com'
    from Staff as S
    select @email = email from Staff where Staff.ID_staff = @Counter
```

```

PRINT CONVERT(VARCHAR,@Counter) + '. employee's email address is ' + @email
SET @Counter = @Counter + 1
END
select * from Staff

```

1	13	Popescu	Alex	alex.popescu@example.com	0744555666
2	14	Petrescu	Maria	maria.ionescu@example.com	0733666777
3	15	Georgescu	Andrei	andrei.georgescu@example.com	0711223344
4	16	Stanescu	Elena	elena.stanescu@example.com	0722333444
5	17	Dumitrescu	Ion	ion.dumitrescu@example.com	0755667788
6	18	Popa	Maria	maria.popa@example.com	0766777888
7	19	Mihai	Vlad	vlad.mihai@example.com	0711223344
8	20	Gheorghe	Andreea	andreea.gheorghe@example.com	0744555666

	ID_staff	nume	prenume	email	telefon
3	15	Georgescu	Andrei	Georgescu.A@aero.com	0711223344
4	16	Stanescu	Elena	Stanescu.E@aero.com	0722333444
5	17	Dumitrescu	Ion	Dumitrescu.I@aero.com	0755667788
6	18	Popa	Maria	Popa.M@aero.com	0766777888
7	19	Mihai	Vlad	Mihai.V@aero.com	0711223344
8	20	Gheorghe	Andreea	Gheorghe.A@aero.com	0744555666
9	21	Radu	Ioana	Radu.I@aero.com	0755667788
10	22	Popescu	Mihai	Popescu.M@aero.com	0766777888

4. Modificati numele angajatilor astfel incat sa se concateneze un 'M' in fata prenumelor celor care lucreaza in departamentul 'Mentenanța'.

```

DECLARE @newprenume varchar(30);
UPDATE Staff
SET Staff.prenume = CONCAT('M', prenume)---Substring(prenume,3,100)--CONCAT('M', prenume)
SELECT @newprenume = prenume
FROM Staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Departamente.denumire = 'Mentenanța'

select Staff.*, Departamente.denumire
from Staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Departamente.denumire = 'Mentenanța'

```

	ID_staff	nume	prenume	email	telefon	denumire
1	16	Stanescu	MElena	Stanescu.E@aero.com	0722333444	Mentenanata
2	18	Popa	MMaria	Popa.M@aero.com	0766777888	Mentenanata

5. Modificati clasa BUSSINES in ECONOMIC celor care calatoresc cu 'Alitalia'.

```

SELECT Bilete.ID_bilet, Companii.num_companie, Clase.NUME_CLASA
FROM Bilete
INNER JOIN Clase ON Clase.ID_clasa = Bilete.ID_clasa
INNER JOIN BiletePerZbor ON Bilete.ID_bilet = BiletePerZbor.id_bilet
INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
INNER JOIN Avioane ON Avioane.ID_avion = Zboruri.ID_avion
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.num_companie = 'Alitalia'

```

```

UPDATE Clase
SET Clase.NUME_CLASA = 'ECONOMIC'
FROM Bilete
INNER JOIN Clase ON Clase.ID_clasa = Bilete.ID_clasa
INNER JOIN BiletePerZbor ON Bilete.ID_bilet = BiletePerZbor.id_bilet
INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
INNER JOIN Avioane ON Avioane.ID_avion = Zboruri.ID_avion
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.num_companie = 'Alitalia'

```

```

SELECT Bilete.ID_bilet, Companii.num_companie, Clase.NUME_CLASA
FROM Bilete
INNER JOIN Clase ON Clase.ID_clasa = Bilete.ID_clasa
INNER JOIN BiletePerZbor ON Bilete.ID_bilet = BiletePerZbor.id_bilet
INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
INNER JOIN Avioane ON Avioane.ID_avion = Zboruri.ID_avion
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.num_companie = 'Alitalia'

```

	ID_bilet	nume_companie	NUME_CLASA
1	11	Alitalia	BUSINESS
2	12	Alitalia	ECONOMIC
3	13	Alitalia	ECONOMIC
4	14	Alitalia	BUSINESS

	ID_bilet	nume_companie	NUME_CLASA
1	11	Alitalia	ECONOMIC
2	12	Alitalia	ECONOMIC
3	13	Alitalia	ECONOMIC
4	14	Alitalia	ECONOMIC

6. Actualizati modelul unui avion in urma schimbarii acestuia de catre companie.

```
SELECT*
FROM Avioane

UPDATE Avioane
SET model = 'Airbus A380', capacitate = '400'
WHERE ID_avion = 9;

SELECT*
FROM Avioane
```

5	5	Boeing 777	396	16
6	6	Airbus A330	335	17
7	7	Embraer E190	100	18
8	8	Bombardier CRJ700	78	19
9	9	ATR 72	74	20
10	10	Sukhoi Superjet 100	103	21
11	11	Airbus A340	380	18
12	12	Airbus A340	380	14

	ID_avion	model	capacitate	ID_companie
4	4	Airbus A320	180	15
5	5	Boeing 777	396	16
6	6	Airbus A330	335	17
7	7	Embraer ...	100	18
8	8	Bombardi...	78	19
9	9	Airbus A380	400	20
10	10	Sukhoi Su...	103	21
11	11	Airbus A340	380	18

7. Actualizati durata zborului cu ID 13.

```
select*  
from zboruri
```

```
UPDATE Zboruri  
SET durata = 120  
WHERE ID_zbor = 13;
```

```
select*  
from zboruri
```

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport
1	12	1	09:00:00.0000000	2023-12-10	120	4
2	13	2	12:30:00.0000000	2023-12-11	180	5
3	14	3	14:45:00.0000000	2023-12-12	90	6
4	15	4	08:00:00.0000000	2023-12-13	240	7
5	16	5	10:15:00.0000000	2023-12-14	150	8
6	17	6	15:30:00.0000000	2023-12-15	180	9
7	18	7	11:00:00.0000000	2023-12-16	120	10
8	19	8	19:00:00.0000000	2023-12-17	360	11

---

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport
1	12	1	09:00:00.0000000	2023-12-10	120	4
2	13	2	12:30:00.0000000	2023-12-11	120	5
3	14	3	14:45:00.0000000	2023-12-12	90	6
4	15	4	08:00:00.0000000	2023-12-13	240	7
5	16	5	10:15:00.0000000	2023-12-14	150	8
6	17	6	15:30:00.0000000	2023-12-15	180	9
7	18	7	11:00:00.0000000	2023-12-16	120	10
8	19	8	19:00:00.0000000	2023-12-17	360	11

8. Actualizati capacitatea tuturor avioanelor companiei Air France la 300.

```
select Avioane.capacitate, Companii.nume_companie  
FROM Avioane  
JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
```

```
UPDATE Avioane  
SET capacitate = 300  
FROM Avioane  
JOIN Companii ON Avioane.ID_companie = Companii.ID_companie  
WHERE Companii.nume_companie = 'Air France';
```

```
select Avioane.capacitate, Companii.nume_companie  
FROM Avioane  
JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
```



	capacitate	nume_companie
1	416	Blue Air
2	853	American Airlines
3	215	Air France
4	180	Alitalia
5	396	Iberia
6	335	British Airways
7	100	Lufthansa
8	78	ANA

---

	capacitate	nume_companie
1	416	Blue Air
2	853	American Airlines
3	300	Air France
4	180	Alitalia
5	396	Iberia
6	335	British Airways
7	100	Lufthansa
8	78	ANA

9. Mariti salariul angajatilor care lucreaza la departamentul Marketing cu 10%.

```
SELECT Staff.*, Departamente.denumire AS Departament, Functii.salariu_brut
FROM Staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff =Staff.ID_staff
INNER JOIN Functii ON Functii.ID_functie = Repartizare_functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Departamente.denumire = 'Marketing'
```

```
UPDATE Functii
SET salariu_brut = salariu_brut * 1.1
FROM Functii
JOIN Departamente ON Functii.ID_departament = Departamente.ID_departament
WHERE Departamente.denumire = 'Marketing';
```

```
SELECT Staff.*, Departamente.denumire AS Departament, Functii.salariu_brut
FROM Staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff =Staff.ID_staff
INNER JOIN Functii ON Functii.ID_functie = Repartizare_functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Departamente.denumire = 'Marketing'
```

	ID_staff	nume	prenume	email	telefon	Departament	salariu_brut
1	22	Popescu	MMihai	Popescu.M@aero.com	0766777888	Marketing	6600

	ID_staff	nume	prenume	email	telefon	Departament	salariu_brut
1	22	Popescu	MMihai	Popescu.M@aero.com	0766777888	Marketing	7260

10. Actualizati ora zborurilor cu durata mai mare de 400 min.

```
select*
from Zboruri

UPDATE Zboruri
SET durata = durata + 30
WHERE durata > 400;

select*
from Zboruri
```

	ID_zbor	ID_avion	ora	data_plecare	dur...	ID_aeropon
1	12	1	09:00:00.0000000	2023-12-10	120	4
2	13	2	12:30:00.0000000	2023-12-11	120	5
3	14	3	14:45:00.0000000	2023-12-12	90	6
4	15	4	08:00:00.0000000	2023-12-13	240	7
5	16	5	10:15:00.0000000	2023-12-14	150	8
6	17	6	15:30:00.0000000	2023-12-15	180	9
7	18	7	11:00:00.0000000	2023-12-16	120	10
8	19	8	19:00:00.0000000	2023-12-17	360	11
9	20	9	02:00:00.0000000	2023-12-18	540	12
10	21	10	09:00:00.0000000	2023-11-19	480	13

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeropo
1	12	1	09:00:00.0000000	2023-12-10	120	4
2	13	2	12:30:00.0000000	2023-12-11	120	5
3	14	3	14:45:00.0000000	2023-12-12	90	6
4	15	4	08:00:00.0000000	2023-12-13	240	7
5	16	5	10:15:00.0000000	2023-12-14	150	8
6	17	6	15:30:00.0000000	2023-12-15	180	9
7	18	7	11:00:00.0000000	2023-12-16	120	10
8	19	8	19:00:00.0000000	2023-12-17	360	11
9	20	9	02:00:00.0000000	2023-12-18	570	12
10	21	10	09:00:00.0000000	2023-11-19	510	13

11. Actualizati task-urile angajatilor care lucreaza in departamentul Marketing.

```
UPDATE Tasks
SET nume = 'Actualizarea strategiei de marketing'
FROM Tasks
INNER JOIN StaffTasks ON Tasks.ID_task = StaffTasks.ID_task
--INNER JOIN Staff ON Staff.ID_staff = StaffTasks.ID_staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = StaffTasks.ID_staff
INNER JOIN Functii ON Functii.ID_functie = Repartizare_functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Departamente.denumire= 'Marketing'
```

12. Actualizati ora tuturor zborurilor companiei 'Lufthansa'.

```
UPDATE Zboruri
SET ora = '10:00'
FROM Zboruri
INNER JOIN Avioane ON Zboruri.ID_avion = Avioane.ID_avion
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.nume_companie = 'Lufthansa';
```

(1 row affected)

Completion time: 2023-05-06T21:10:30.2288143+03:00

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport	nume_companie
1	18	7	10:00:00.0000000	2023-12-16	120	10	Lufthansa

13. Amanati toate zborurile cu o ora pentru o anumita zi.

```
select*
from Zboruri
```

```
UPDATE Zboruri
SET ora = DATEADD(hour, 1, ora)
WHERE data_plecare = '2023-12-10'
```

```
select*
from Zboruri
```

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport
1	12	1	09:00:00.0000000	2023-12-10	120	4
2	13	2	12:30:00.0000000	2023-12-11	120	5
3	14	3	14:45:00.0000000	2023-12-12	90	6
4	15	4	08:00:00.0000000	2023-12-13	240	7
5	16	5	10:15:00.0000000	2023-12-14	150	8
6	17	6	15:30:00.0000000	2023-12-15	180	9
7	18	7	10:00:00.0000000	2023-12-16	120	10
8	19	8	19:00:00.0000000	2023-12-17	360	11

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport
1	12	1	10:00:00.0000000	2023-12-10	120	4
2	13	2	12:30:00.0000000	2023-12-11	120	5
3	14	3	14:45:00.0000000	2023-12-12	90	6
4	15	4	08:00:00.0000000	2023-12-13	240	7
5	16	5	10:15:00.0000000	2023-12-14	150	8
6	17	6	15:30:00.0000000	2023-12-15	180	9
7	18	7	10:00:00.0000000	2023-12-16	120	10
8	19	8	19:00:00.0000000	2023-12-17	360	11

14. Actualizati numele tarilor care incep cu litera mica astfel incat sa inceapa cu litera mare (in cazul in care s-a strecurat vreo greseala).

```
select*
from Tari

UPDATE Tari
SET nume_tara = UPPER(LEFT(nume_tara, 1)) + SUBSTRING(nume_tara, 2, LEN(nume_tara))
WHERE LOWER(LEFT(nume_tara, 1)) = LEFT(nume_tara, 1)

select*
from Tari
```

4	8	SUA
5	9	Spain
6	10	Marea Britanie
7	11	Germania
8	12	Japonia
9	13	China
10	14	Rusia
11	31	armenia

	ID_tara	nume_tara
4	8	SUA
5	9	Spain
6	10	Marea Br...
7	11	Germania
8	12	Japonia
9	13	China
10	14	Rusia
11	31	Armenia

15. Actualizarea descrierii contractului prin adaugarea de informatii.

```
select Contracte.descriere, Functii.nume AS Functie
from Contracte
INNER JOIN Staff ON Contracte.ID_angajat = Staff.ID_staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
WHERE Functii.nume = 'Manager'
```

```
UPDATE Contracte
SET descriere = CONCAT(descriere, '. 40 de ore de munca pe saptamana')
FROM Contracte
INNER JOIN Staff ON Contracte.ID_angajat = Staff.ID_staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
WHERE Functii.nume = 'Manager'
```

```
select Contracte.descriere, Functii.nume AS Functie
from Contracte
INNER JOIN Staff ON Contracte.ID_angajat = Staff.ID_staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
WHERE Functii.nume = 'Manager'
```

	descriere	Functie
1	Contract de munca pe perioada nedeterminata	Manager

	descriere	Functie
1	Contract de munca pe perioada nedeterminata. 40 d...	Manager

# Stergerea de date (DELETE)

1. Stergerea locatiile care nu au o tara asociata.

```
SET IMPLICIT_TRANSACTIONS ON
DELETE
FROM Locatii
WHERE Locatii.ID_tara is NULL
ROLLBACK;
```

---

(0 rows affected)

Completion time: 2023-05-08T09:49:10.1234805+03:00

2. Stergeti biletul pentru un anumit calator cu un anumit CNP.

```
select*
from Calatori
```

```
select*
from Bilete
```

```
DELETE Bilete
FROM Bilete
INNER JOIN Calatori ON Bilete.ID_calator = Calatori.ID_calator
WHERE Calatori.CNP = '1234567890132'
```

3	68	1234567890125	Popa	Maria	FEMININ	maria.p@gmail.com	0731123456	Str. Unirii nr. 15
4	69	1234567890126	Vasilescu	Alexandra	FEMININ	alex.v@gmail.com	0741123456	Str. Mihai Viteazul nr. 5
5	70	1234567890127	Popescu	Mihai	MASCULIN	mihai.p@gmail.com	0726123456	Str. Republicii nr. 7
6	71	1234567890128	Dumitru	Diana	FEMININ	diana.d@gmail.com	0751123456	Str. Ion Creanga nr. 12
7	72	1234567890129	Radulescu	Andrei	MASCULIN	andrei.r@gmail.com	0729123456	Str. Gheorghe Doja nr. 14
8	73	1234567890130	Georgescu	Elena	FEMININ	elena.g@gmail.com	0742123456	Str. Tudor Vladimirescu nr. 17
9	74	1234567890131	Petrescu	Marian	MASCULIN	marian.p@gmail.com	0732123456	Str. Vasile Alecsandri nr. 19
10	75	1234567890132	Stefanescu	Alina	FEMININ	alina.s@gmail.com	0761123456	Str. Stefan cel Mare nr. 9

ID_bilet	ID_calator	ID_clasa	pret_bilet
4	11	69	3
5	12	70	3
6	13	71	3
7	14	72	4
8	15	73	3
9	16	74	3
10	17	75	3
11	18	66	3

3	68	1234567890125	Popa	Maria	FEMININ	maria.p@gmail.com	0731123456	Str. Unirii nr. 15
4	69	1234567890126	Vasilescu	Alexandra	FEMININ	alex.v@gmail.com	0741123456	Str. Mihai Viteazul nr. 5
5	70	1234567890127	Popescu	Mihai	MASCULIN	mihai.p@gmail.com	0726123456	Str. Republicii nr. 7
6	71	1234567890128	Dumitru	Diana	FEMININ	diana.d@gmail.com	0751123456	Str. Ion Creanga nr. 12
7	72	1234567890129	Radulescu	Andrei	MASCULIN	andrei.r@gmail.com	0729123456	Str. Gheorghe Doja nr. 14
8	73	1234567890130	Georgescu	Elena	FEMININ	elena.g@gmail.com	0742123456	Str. Tudor Vladimirescu nr. 17
9	74	1234567890131	Petrescu	Marian	MASCULIN	marian.p@gmail.com	0732123456	Str. Vasile Alecsandri nr. 19
10	75	1234567890132	Stefanescu	Alina	FEMININ	alina.s@gmail.com	0761123456	Str. Stefan cel Mare nr. 9

	ID_bilet	ID_calator	ID_clasa	pret_bilet
3	10	68	3	200
4	11	69	3	1971
5	12	70	3	292
6	13	71	3	192
7	14	72	4	1971
8	15	73	3	300
9	16	74	3	400
10	18	66	3	300

3. Stergeti toate zborurile care au loc intr-o anumita zi.

```
select*
from Zboruri

DELETE
FROM Zboruri
WHERE Zboruri.data_plecare = '2023-11-01'
```

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport
1	63	1	09:00:00.0000000	2023-12-10	120	34
2	64	2	12:30:00.0000000	2023-12-11	180	35
3	65	3	14:45:00.0000000	2023-12-12	90	36
4	66	4	08:00:00.0000000	2023-12-13	240	37
5	67	5	10:15:00.0000000	2023-12-14	150	38
6	68	6	15:30:00.0000000	2023-12-15	180	39
7	69	7	11:00:00.0000000	2023-12-16	120	40
8	70	8	19:00:00.0000000	2023-12-17	360	41
9	71	9	02:00:00.0000000	2023-12-18	540	42
10	72	10	05:30:00.0000000	2023-12-19	480	43
11	73	1	12:30:00.0000000	2023-11-01	120	34
12	74	2	12:00:00.0000000	2023-11-01	180	35

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport
1	63	1	09:00:00.0000000	2023-12-10	120	34
2	64	2	12:30:00.0000000	2023-12-11	180	35
3	65	3	14:45:00.0000000	2023-12-12	90	36
4	66	4	08:00:00.0000000	2023-12-13	240	37
5	67	5	10:15:00.0000000	2023-12-14	150	38
6	68	6	15:30:00.0000000	2023-12-15	180	39
7	69	7	11:00:00.0000000	2023-12-16	120	40
8	70	8	19:00:00.0000000	2023-12-17	360	41
9	71	9	02:00:00.0000000	2023-12-18	540	42
10	72	10	05:30:00.0000000	2023-12-19	480	43

4. Stergeti un angajat in urma terminarii contractului. (stergere contract, legatura din ContracteImpozite si cea din StaffTasks)

```
DELETE Staff
FROM Staff
INNER JOIN Contracte ON Staff.ID_staff = Contracte.ID_angajat
INNER JOIN StaffTasks ON Staff.ID_staff = StaffTasks.ID_staff
WHERE Contracte.ID_contract = 15
```

```
select*
from Contracte
```

```
select*
FROM Staff
INNER JOIN Contracte ON Staff.ID_staff = Contracte.ID_angajat
INNER JOIN StaffTasks ON Staff.ID_staff = StaffTasks.ID_staff
WHERE Contracte.ID_contract = 15
```

	ID_staff	nume	prenume	email	telefon	ID_contract	ID_angajat	descriere	ID_StaffTask	ID_staff	ID_task
1	21	Radu	Oana	Radu.I@aero.com	0755667788	15	21	Contract de munca pe perioada nedeterminata	14	21	11

	ID_contract	ID_angajat	descriere
1	7	13	Contract de munca pe perioada nedeterminata. 40 ...
2	8	14	Contract de munca pe perioada determinata
3	9	15	Contract de colaborare
4	10	16	Contract de prestari servicii
5	11	17	Contract de ucenicie
6	12	18	Contract de munca pe perioada determinata
7	13	19	Contract de munca in regim de munca flexibil
8	14	20	Contract de munca pe perioada determinata
9	16	22	Contract de munca pe perioada nedeterminata

ID_staff	nume	prenume	email	telefon	ID_contract	ID_angajat	descriere	ID_StaffTask	ID_staff	ID_task
----------	------	---------	-------	---------	-------------	------------	-----------	--------------	----------	---------

5. Stergeti toate taskurile angajatilor din departamentul 'Marketing'. \*

```
DELETE Tasks
FROM Tasks
INNER JOIN StaffTasks ON StaffTasks.ID_task = Tasks.ID_task
INNER JOIN Staff ON StaffTasks.ID_staff = Staff.ID_staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Departamente.denumire = 'Marketing'
```

```
select Departamente.denumire AS Departament, Tasks.nume AS Task
```



```

FROM Tasks
INNER JOIN StaffTasks ON StaffTasks.ID_task = Tasks.ID_task
INNER JOIN Staff ON StaffTasks.ID_staff = Staff.ID_staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Departamente.denumire = 'Marketing'

```

	Departament	Task
1	Marketing	Actualizarea strategiei de marketing

(1 row affected)

Completion time: 2023-05-09T16:59:02.0040531+03:00

Departament	Task
Click to select the w	

6. Stergeti biletele zborului care are destinatia 'Italia'.

```

SELECT Tari.num_e_tara, Bilete.ID_bilet, Bilete.pret_bilet
FROM Tari
INNER JOIN Locatii ON Locatii.ID_tara = Tari.ID_tara
INNER JOIN Aeroporturi ON Aeroporturi.ID_oras = Locatii.ID_locatie
INNER JOIN Zboruri ON Zboruri.ID_aeroport = Aeroporturi.ID_aeroport
INNER JOIN BiletePerZbor ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
INNER JOIN Bilete ON Bilete.ID_bilet = BiletePerZbor.id_bilet
WHERE Tari.num_e_tara = 'Italia'

```

```

DELETE Bilete
FROM Tari
INNER JOIN Locatii ON Locatii.ID_tara = Tari.ID_tara
INNER JOIN Aeroporturi ON Aeroporturi.ID_oras = Locatii.ID_locatie
INNER JOIN Zboruri ON Zboruri.ID_aeroport = Aeroporturi.ID_aeroport
INNER JOIN BiletePerZbor ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
INNER JOIN Bilete ON Bilete.ID_bilet = BiletePerZbor.id_bilet
WHERE Tari.num_e_tara = 'Italia'

```

```

SELECT Tari.num_e_tara, Bilete.ID_bilet, Bilete.pret_bilet
FROM Tari
INNER JOIN Locatii ON Locatii.ID_tara = Tari.ID_tara
INNER JOIN Aeroporturi ON Aeroporturi.ID_oras = Locatii.ID_locatie

```

```

INNER JOIN Zboruri ON Zboruri.ID_aeroport = Aeroporturi.ID_aeroport
INNER JOIN BiletePerZbor ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
INNER JOIN Bilete ON Bilete.ID_bilet = BiletePerZbor.id_bilet
WHERE Tari.nume_tara = 'Italia'

```

	nume_tara	ID_bilet	pret_bilet
1	Italia	15	300
2	Italia	16	400
3	Italia	19	300
4	Italia	20	300

nume_tara	ID_bilet	pret_bilet
-----------	----------	------------

7. Stergeti angajatii care au functie de Manager si nu au niciun task.

```

select Staff.*
FROM Staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN StaffTasks ON StaffTasks.ID_staff = Staff.ID_staff
WHERE StaffTasks.ID_task is null

```

	ID_staff	nume	prenume	email	telefon
1	26	Andronic	Catrina	andronic.c@gmail.com	0787546366

```

DELETE Staff
FROM Staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Repartizare_functii.ID_functie = Functii.ID_functie
INNER JOIN StaffTasks ON StaffTasks.ID_staff = Staff.ID_staff
WHERE Functii.nume = 'Manager' and StaffTasks.ID_task is null

```

(1 row affected)

Completion time: 2023-05-09T20:50:56.3758763+03:00

ID_staff	nume	prenume	email	telefon
----------	------	---------	-------	---------

8. Stergeti calatorii care zboara cu o anumita companie.

```
select Calatori.nume, Calatori.prenume ,Companii.nume_companie
FROM Calatori
INNER JOIN Bilete ON Bilete.ID_calator = Calatori.ID_calator
INNER JOIN BiletePerZbor ON BiletePerZbor.id_bilet = Bilete.ID_bilet
INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
INNER JOIN Avioane ON Zboruri.ID_zbor = Zboruri.ID_zbor
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.nume_companie = 'Iberia'
```

	nume	prenume	nume_companie
1	Popescu	Ana	Iberia
2	Ionescu	Alex	Iberia
3	Popa	Maria	Iberia
4	Vasilescu	Alexandra	Iberia
5	Popescu	Mihai	Iberia
6	Dumitru	Diana	Iberia
7	Radulescu	Andrei	Iberia

```
DELETE Calatori
FROM Calatori
INNER JOIN Bilete ON Bilete.ID_calator = Calatori.ID_calator
INNER JOIN BiletePerZbor ON BiletePerZbor.id_bilet = Bilete.ID_bilet
INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
INNER JOIN Avioane ON Zboruri.ID_zbor = Zboruri.ID_zbor
INNER JOIN Companii ON Avioane.ID_companie = Companii.ID_companie
WHERE Companii.nume_companie = 'Air France'
ROLLBACK;
```

9. Stergeti contractele din departamentul 'Marketing' in care exista contracte pe o perioada nedeterminata.

```
Select Contracte.descriere, Departamente.denumire
FROM Contracte
INNER JOIN Staff ON Staff.ID_staff = Contracte.ID_angajat
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Functii.ID_functie = Repartizare_functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Contracte.descriere = 'Contract de munca pe perioada nedeterminata' and
Departamente.denumire = 'Marketing'
```

	descriere	denumire
1	Contract de munca pe perioada nedeterminata	Marketing

```

DELETE Contracte
FROM Contracte
INNER JOIN Staff ON Staff.ID_staff = Contracte.ID_angajat
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Functii.ID_functie = Repartizare_functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
WHERE Contracte.descriere = 'Contract de munca pe perioada nedeterminata' and
Departamente.denumire = 'Marketing'
ROLLBACK;

```

(1 row affected)

Completion time: 2023-05-10T00:11:29.9261108+03:00

10. Stergeti zborurile care au destinatia Italia si apartin companiei Alitalia.

```

SELECT Zboruri.*, Companii.nume_comanie
FROM Companii
INNER JOIN Avioane ON Avioane.ID_comanie = Companii.ID_comanie
INNER JOIN Zboruri ON Zboruri.ID_avion = Avioane.ID_avion
INNER JOIN Aeroporturi ON Aeroporturi.ID_aeroport = Zboruri.ID_aeroport
INNER JOIN Locatii ON Locatii.ID_locatie = Aeroporturi.ID_oras
INNER JOIN Tari ON Tari.ID_tara = Locatii.ID_tara
WHERE Tari.nume_tara = 'Italia' and Companii.nume_comanie = 'Alitalia'

```

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport	nume_comanie
1	66	4	08:00:00.0000000	2023-12-13	240	37	Alitalia

SET IMPLICIT\_TRANSACTIONS ON

```

DELETE Zboruri
FROM Companii
INNER JOIN Avioane ON Avioane.ID_comanie = Companii.ID_comanie
INNER JOIN Zboruri ON Zboruri.ID_avion = Avioane.ID_avion
INNER JOIN Aeroporturi ON Aeroporturi.ID_aeroport = Zboruri.ID_aeroport
INNER JOIN Locatii ON Locatii.ID_locatie = Aeroporturi.ID_oras
INNER JOIN Tari ON Tari.ID_tara = Locatii.ID_tara
WHERE Tari.nume_tara = 'Italia' and Companii.nume_comanie = 'Alitalia'
ROLLBACK;

```

(0 rows affected)

(1 row affected)

Completion time: 2023-05-10T12:27:06.1782309+03:00

11. Stergeti angajatii carora li se aplica impozite de tipul 'Contributia la Fondul de risc si accidente de munca - clasa de risc II'.

```
SELECT Staff.*, Impozite.ume AS Impozit
FROM IMPOZITE
INNER JOIN ContracteImpozite ON ContracteImpozite.ID_impozit = Impozite.ID_impozit
INNER JOIN Contracte ON Contracte.ID_contract = ContracteImpozite.ID_contract
INNER JOIN Staff ON Staff.ID_staff = Contracte.ID_angajat
WHERE Impozite.ume = 'Contributia la Fondul de risc si accidente de munca - clasa
de risc II'
```

	ID_staff	nume	prenume	email	telefon	Impozit
1	22	Popescu	Mihai	Popescu.M@aero.com	0766777888	Contributia la Fondul de risc si accidente de munca - clasa de risc II

```
DELETE Staff
FROM IMPOZITE
INNER JOIN ContracteImpozite ON ContracteImpozite.ID_impozit = Impozite.ID_impozit
INNER JOIN Contracte ON Contracte.ID_contract = ContracteImpozite.ID_contract
INNER JOIN Staff ON Staff.ID_staff = Contracte.ID_angajat
WHERE Impozite.ume = 'Contributia la Fondul de risc si accidente de munca - clasa
de risc II'
ROLLBACK;
```

(1 row affected)

Completion time: 2023-05-10T12:40:10.4915786+03:00

12. Stergeti angajatii care au mai mult un task.

```
SELECT STAFF.ID_staff, Staff.numa, Staff.prenume
FROM Staff
INNER JOIN StaffTasks ON StaffTasks.ID_staff = Staff.ID_staff
INNER JOIN Tasks ON StaffTasks.ID_task = Tasks.ID_task
GROUP BY Staff.ID_staff, Staff.numa, Staff.prenume
HAVING COUNT(Tasks.ID_task) > 1
```

	ID_staff	numa	prenume
1	17	Dumitrescu	Ion
2	20	Gheorghe	Andreea

```
DELETE Staff
FROM Staff
INNER JOIN StaffTasks ON StaffTasks.ID_staff = Staff.ID_staff
INNER JOIN Tasks ON StaffTasks.ID_task = Tasks.ID_task
WHERE (SELECT COUNT(*) FROM Tasks WHERE StaffTasks.ID_task = Tasks.ID_task) > 1
ROLLBACK;
```

13. Stergeti companiile care au mai multe de un avion.

```
SELECT Companii.numa_companie, COUNT (Avioane.ID_avion) AS NR
FROM Companii
INNER JOIN Avioane ON Avioane.ID_companie = Companii.ID_companie
GROUP BY Companii.numa_companie
```

	numa_companie	NR
1	Aeroflot	1
2	Air China	1
3	Air France	2
4	Alitalia	1
5	American Airlines	1
6	ANA	1
7	Blue Air	1
8	British Airways	1
9	Iberia	1
10	Lufthansa	2

```
DELETE Avioane
FROM Avioane
WHERE (SELECT COUNT(*) FROM Companii WHERE Avioane.ID_companie = Companii.ID_companie)
> 1
ROLLBACK;
```

14. Stergeti calatorii cu numarul de telefon '0770985467'

```
DELETE Calatori
FROM Calatori
WHERE Calatori.telefon = '0770985467'
ROLLBACK;
```

(1 row affected)

Completion time: 2023-05-10T13:59:09.1076164+03:00

15. Stergeti toate locatiile pentru tara asociata 'SUA'.

```
DELETE Locatii
FROM Locatii
INNER JOIN Tari ON Tari.ID_tara = Locatii.ID_tara
WHERE Tari.nume_tara = 'SUA'
ROLLBACK;
```

---

(0 rows affected)

(1 row affected)

Completion time: 2023-05-10T14:01:38.4749242+03:00

# Vizualizari

1. Creati un view care sa contina destinatia zborurilor (nume tara, nume aeroport, nume locatie).

```
IF OBJECT_ID ('Destinatii', 'V') IS NOT NULL
DROP VIEW Destinatii
```

GO

```
create view Destinatii (Aeroport, Locatie, Tara)
as
select Aeroporturi.nume , Locatii.nume_locatie, Tari.nume_tara
from Aeroporturi
INNER JOIN Locatii ON Locatii.ID_locatie = Aeroporturi.ID_oras
INNER JOIN Tari ON Tari.ID_tara = Locatii.ID_tara
```

GO

```
select * from Destinatii
```

	Aeroport	Locatie	Tara
1	Marco Polo	Venetia	Italia
2	Henri Coanda	Bucuresti	Romania
3	JFK	New York	SUA
4	Fiumicino	Roma	Italia
5	El Prat	Barcelona	Spain
6	Heathrow	Londra	Marea Britanie
7	Tegel	Berlin	Germania
8	Narita	Tokyo	Japonia
9	Beijing Capital	Beijing	China
10	Sheremetyevo	Moscova	Rusia

2. Creati un view care contine calatorii impreuna cu zborurile fiecaruia si clasa.

```
IF OBJECT_ID ('Pasageri', 'V') IS NOT NULL
DROP VIEW Pasageri
```

GO

```
CREATE VIEW Pasageri
as
```

```
select Calatori.nume AS Prenume, Calatori.prenume AS Nume, Clase.NUME_CLASA AS
Clasa, Zboruri.durata AS 'Durata zbor', Zboruri.data_plecure AS Data, Zboruri.ora AS Ora
FROM Calatori
INNER JOIN Bilete ON Bilete.ID_calator = Calatori.ID_calator
INNER JOIN Clase ON Bilete.ID_clasa = Clase.ID_clasa
```



```
INNER JOIN BiletePerZbor ON BiletePerZbor.id_bilet = Bilete.ID_bilet
INNER JOIN Zboruri ON BiletePerZbor.id_zbor = Zboruri.ID_zbor
```

```
select*
from Pasageri
```

	Prenume	Nume	Clasa	Durata zbor	Data	Ora
1	Popescu	Ana	ECONOMIC	180	2023-12-11	12:30:00.0000000
2	Ionescu	Alex	ECONOMIC	180	2023-12-11	12:30:00.0000000
3	Popa	Maria	ECONOMIC	180	2023-12-11	12:30:00.0000000
4	Vasilescu	Alexandra	ECONOMIC	90	2023-12-12	14:45:00.0000000
5	Popescu	Mihai	ECONOMIC	90	2023-12-12	14:45:00.0000000
6	Dumitru	Diana	ECONOMIC	90	2023-12-12	14:45:00.0000000
7	Radulescu	Andrei	ECONOMIC	90	2023-12-12	14:45:00.0000000
8	Georgescu	Elena	ECONOMIC	240	2023-12-13	08:00:00.0000000
9	Petrescu	Marian	ECONOMIC	240	2023-12-13	08:00:00.0000000
10	Stefanescu	Alina	ECONOMIC	240	2023-12-13	08:00:00.0000000

3. Creati un view care contine toti angajatii cu functia si departamentul in care lucreaza.

```
IF OBJECT_ID ('Angajati', 'V') IS NOT NULL
DROP VIEW Angajati

GO
CREATE VIEW Angajati
as
select Staff.num AS Nume, Staff.prenume AS Prenume, Functii.ID_functie AS
Functie, Departamente.denumire AS Departament
from Staff
INNER JOIN Repartizare_functii ON Repartizare_functii.ID_staff = Staff.ID_staff
INNER JOIN Functii ON Functii.ID_functie = Repartizare_functii.ID_functie
INNER JOIN Departamente ON Departamente.ID_departament = Functii.ID_departament
```

	Nume	Prenume	Functie	Departament
1	Popescu	Alex	4	Planificare si dezvoltare
2	Petrescu	Maria	5	Tehnologia informatiei
3	Georgescu	Andrei	6	Financiar-contabil
4	Stanescu	Elena	7	Mentenanata
5	Dumitrescu	Ion	8	Operatiuni aeroportuare
6	Popa	Maria	9	Mentenanata
7	Mihai	Vlad	10	Financiar-contabil
8	Gheorghe	Andreea	11	Siguranta
9	Popescu	Mihai	13	Marketing
10	Radu	Ooana	12	Aprovizionare

4. Creati un view care sa contina companiile impreuna cu avioanele si zborurile fiecareia.

```
IF OBJECT_ID ('OrganizareZboruri', 'V') IS NOT NULL
DROP VIEW OrganizareZboruri
```

GO

```
CREATE VIEW OrganizareZboruri
as
SELECT Companii.num_e_companie AS Companie , Zboruri.data_plecure AS DataPlecure,
Zboruri.durata AS Durata, Zboruri.ora AS Ora, Avioane.model AS Avion
FROM Companii
INNER JOIN Avioane ON Avioane.ID_companie = Companii.ID_companie
INNER JOIN Zboruri ON Zboruri.ID_avion = Avioane.ID_avion

select*
from OrganizareZboruri
```

	Companie	DataPlecure	Durata	Ora	Avion
1	Blue Air	2023-12-10	120	09:00:00.0000000	Boeing 747
2	American Airlines	2023-12-11	180	12:30:00.0000000	Airbus A380
3	Air France	2023-12-12	90	14:45:00.0000000	Boeing 737
4	Alitalia	2023-12-13	240	08:00:00.0000000	Airbus A320
5	Iberia	2023-12-14	150	10:15:00.0000000	Boeing 777
6	British Airways	2023-12-15	180	15:30:00.0000000	Airbus A330
7	Lufthansa	2023-12-16	120	11:00:00.0000000	Embraer E190
8	ANA	2023-12-17	360	19:00:00.0000000	Bombardier CRJ700
9	Air China	2023-12-18	540	02:00:00.0000000	Airbus A380
10	Aeroflot	2023-12-19	480	05:30:00.0000000	Sukhoi Superjet 100

5. Creati un view care sa contina taskurile si numele si prenumele angajatilor.

```
IF OBJECT_ID ('Responsabiliti', 'V') IS NOT NULL
    DROP VIEW Responsabiliti

GO
CREATE VIEW Responsabiliti
as
    select Staff.num AS Nume, Staff.prenume AS Prenume, Tasks.num AS Task
    from Staff
    INNER JOIN StaffTasks ON Staff.ID_staff = StaffTasks.ID_staff
    INNER JOIN Tasks ON StaffTasks.ID_task = Tasks.ID_task

    select*
    from Responsabiliti
```

	Nume	Prenume	Task
1	Popescu	Alex	Actualizarea strategiei de marketing
2	Petrescu	Maria	Rezolvarea problemelor ?i întâmpinarea nevoilor ...
3	Georgescu	Andrei	Urmărirea pla?ilor ?i a încasarilor ?i întocmirea ra...
4	Stanescu	Elena	Verificarea tehnica a aeronavei înainte de decolare
5	Dumitrescu	Ion	Îmbarcarea pasagerilor
6	Dumitrescu	Ion	Verificarea biletelor ?i a documentelor pasagerilor
7	Popa	Maria	Verificarea tehnica a aeronavei înainte de decolare
8	Mihai	Vlad	Rezolvarea problemelor ?i întâmpinarea nevoilor ...
9	Gheorghe	Andreea	Asigurarea siguran?ei pasagerilor ?i a aeronavei
10	Gheorghe	Andreea	Încarcarea ?i descarcarea bagajelor
11	Radu	Oana	Aprovizionarea aeronavei cu combustibil ?i alte pr...
12	Popescu	Mihai	Actualizarea strategiei de marketing

6. Creati un view care sa contina denumirea contractelor, a impozitelor, a functiilor si numele si prenumele angajatilor.

```
IF OBJECT_ID ('ContractesiImpozite', 'V') IS NOT NULL
    DROP VIEW ContractesiImpozite

GO
CREATE VIEW ContractesiImpozite
as
    SELECT Staff.num, Staff.prenume, Contracte.descriere AS Contract, Impozite.num
AS Impozit
    FROM Functii
```

```

INNER JOIN Repartizare_funcatii ON Functii.ID_functie =
Repartizare_funcatii.ID_functie
INNER JOIN Staff ON Repartizare_funcatii.ID_staff = Staff.ID_staff
INNER JOIN Contracte ON Contracte.ID_angajat = Staff.ID_staff
INNER JOIN ContracteImpozite ON Contracte.ID_contract =
ContracteImpozite.ID_contract
INNER JOIN Impozite ON Impozite.ID_impozit = ContracteImpozite.ID_impozit

SELECT*
FROM ContractesiImpozite

```

	nume	prenume	Contract	Impozit
1	Popescu	Alex	Contract de munca pe perioada nedeterminata. 40 ...	Impozit pe venitul salarial
2	Popescu	Alex	Contract de munca pe perioada nedeterminata. 40 ...	Contributie la CASS
3	Popescu	Alex	Contract de munca pe perioada nedeterminata. 40 ...	Contributie la CAS
4	Petrescu	Maria	Contract de munca pe perioada determinata	Impozit pe venitul salarial
5	Petrescu	Maria	Contract de munca pe perioada determinata	Contributie la CASS
6	Petrescu	Maria	Contract de munca pe perioada determinata	Contributie la CAS
7	Georgescu	Andrei	Contract de colaborare	Impozit pe venitul salarial
8	Georgescu	Andrei	Contract de colaborare	Contributie la CASS
9	Stanescu	Elena	Contract de prestari servicii	Impozit pe venitul salarial
10	Dumitrescu	Ion	Contract de ucenicie	Impozit pe venitul salarial
11	Popa	Maria	Contract de munca pe perioada determinata	Impozit pe venitul salarial
12	Mihai	Vlad	Contract de munca in regim de munca flexibil	Impozit pe venitul salarial
13	Gheorghe	Andreea	Contract de munca pe perioada determinata	Contributie la CASS
14	Gheorghe	Andreea	Contract de munca pe perioada determinata	Contributia la Fondul de garantare pentru plata c...
15	Popescu	Mihai	Contract de munca pe perioada nedeterminata	Contributia la Fondul de risc si accidente de mu...
16	Popescu	Mihai	Contract de munca pe perioada nedeterminata	Contributie la CASS
17	Popescu	Mihai	Contract de munca pe perioada nedeterminata	Impozit pe venitul salarial

7. Creati un view care sa contina toate zborurile si portile alocate acestora.

```

IF OBJECT_ID ('Porti', 'V') IS NOT NULL
    DROP VIEW Porti

GO
CREATE VIEW Porti
as
    SELECT Zboruri.*, Gates.numar AS Gate, Gates.capacitate
    FROM Zboruri
    INNER JOIN ZboruriGates ON Zboruri.ID_zbor = ZboruriGates.ID_zbor
    INNER JOIN Gates ON Gates.ID_gate = ZboruriGates.ID_gate

SELECT*
FROM Porti

```

	ID_zbor	ID_avion	ora	data_plecare	durata	ID_aeroport	Gate	capacitate
1	63	1	09:00:00.0000000	2023-12-10	120	34	2	110
2	64	2	12:30:00.0000000	2023-12-11	180	35	3	120
3	65	3	14:45:00.0000000	2023-12-12	90	36	4	130
4	66	4	08:00:00.0000000	2023-12-13	240	37	5	140
5	67	5	10:15:00.0000000	2023-12-14	150	38	6	150
6	68	6	15:30:00.0000000	2023-12-15	180	39	7	160
7	69	7	11:00:00.0000000	2023-12-16	120	40	8	170
8	70	8	19:00:00.0000000	2023-12-17	360	41	9	180
9	71	9	02:00:00.0000000	2023-12-18	540	42	10	190
10	72	10	05:30:00.0000000	2023-12-19	480	43	1	100
11	72	10	05:30:00.0000000	2023-12-19	480	43	11	NULL

8. Creati un view care include numele si prenumele calatorilor si numele companiei la care au achizitionat bilet.

```

IF OBJECT_ID ('ViewCalatori', 'V') IS NOT NULL
    DROP VIEW ViewCalatori

GO
CREATE VIEW ViewCalatori
as
    SELECT Calatori.num, Calatori.prenume, Companii.num_companie AS Companie
    FROM Calatori
    INNER JOIN Bilete ON Bilete.ID_calator = Calatori.ID_calator
    INNER JOIN BiletePerZbor ON BiletePerZbor.id_bilet = Bilete.ID_bilet
    INNER JOIN Zboruri ON Zboruri.ID_zbor = BiletePerZbor.id_zbor
    INNER JOIN Avioane ON Avioane.ID_avion = Zboruri.ID_avion
    INNER JOIN Companii ON Companii.ID_companie = Avioane.ID_companie

SELECT*
FROM ViewCalatori

```

	nume	prenume	Companie
1	Popescu	Ana	American Airlines
2	Ionescu	Alex	American Airlines
3	Popa	Maria	American Airlines
4	Vasilescu	Alexandra	Air France
5	Popescu	Mihai	Air France
6	Dumitru	Diana	Air France
7	Radulescu	Andrei	Air France

# Triggere

1. Trigger care impiedica adaugarea salariului brut mai mic de 2500 lei.

```
IF OBJECT_ID ('tr_salariu_brut_minim', 'TR') IS NOT NULL
    DROP TRIGGER tr_salariu_brut_minim;

GO
CREATE TRIGGER tr_salariu_brut_minim
ON Functii
FOR INSERT
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE salariu_brut < 2500)
    BEGIN
        RAISERROR ('Salariul brut trebuie să fie mai mare de 2500!' , 16, 1);
        ROLLBACK TRANSACTION;
    END
END;

INSERT INTO Functii (ID_departament, nume, salariu_brut) VALUES (13, 'Controlor acte',
2000)

Msg 50000, Level 16, State 1, Procedure tr_salariu_brut_minim, Line 8 [Batch Start Line 297]
Salariul brut trebuie să fie mai mare de 2500!
Msg 3609, Level 16, State 1, Line 298
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2023-05-10T23:32:20.6062664+03:00
```

2. Trigger care impiedica introducerea aceleiasi nume de companie de mai multe ori.

```
IF OBJECT_ID ('tr_companii_duplicate_name', 'TR') IS NOT NULL DROP TRIGGER
tr_companii_duplicate_name;
go
CREATE TRIGGER tr_companii_duplicate_name
ON Companii
FOR INSERT, UPDATE
AS
BEGIN
    IF EXISTS(
        SELECT 1
        FROM inserted i
        JOIN Companii c ON i.nume_companie = c.nume_companie AND i.ID_companie !=
        c.ID_companie)
    BEGIN
```

```

        RAISERROR('Nu pot fi inserate doua companii cu acelasi nume!', 16, 1);
        ROLLBACK;
    END
END;

```

```

insert into Companii(ume_companie) values('FranceAirline')
insert into Companii(ume_companie) values('FranceAirline')

```

```

Msg 50000, Level 16, State 1, Procedure tr_companii_duplicate_name, Line 12 [Batch Start Line 568]
Nu pot fi inserate doua companii cu acelasi nume!
Msg 3609, Level 16, State 1, Line 570
The transaction ended in the trigger. The batch has been aborted.

```

### 3. Trigger pentru a introduce numarul de telefon in tabela "Staff"

```

go
CREATE TRIGGER trig_staff_telefon
ON Staff
FOR INSERT
AS
BEGIN
IF NOT EXISTS(SELECT * FROM inserted WHERE telefon IS NOT NULL)
BEGIN
RAISERROR('Nu s-a specificat telefonul pentru acest staff!', 16, 1)
ROLLBACK TRANSACTION
END
END

insert into Staff(ume,prenume, email) values('Mariuc','Andrei', 'AndreiMa@email.com')

```

```

-
Msg 50000, Level 16, State 1, Procedure trig_staff_telefon, Line 8 [Batch Start Line 513]
Nu s-a specificat telefonul pentru acest staff!
Msg 3609, Level 16, State 1, Line 514
The transaction ended in the trigger. The batch has been aborted.

```

### 4. Trigger adaugarea unui numar in tabelul "Gates"

```

go
CREATE TRIGGER trig_gates_numar
ON Gates
FOR INSERT
AS
BEGIN
IF NOT EXISTS(SELECT * FROM inserted WHERE numar IS NOT NULL)
BEGIN
RAISERROR('Nu s-a specificat numarul pentru aceasta poarta!', 16, 1)
ROLLBACK TRANSACTION
END

```

END

```
insert into Gates(capacitate) values(60)
```

5. Trigger pentru a preveni adaugarea unei locatii fara a specifica tara

```
IF OBJECT_ID ('trig_locatie_tara', 'TR') IS NOT NULL
DROP TRIGGER trig_locatie_tara;
```

```
go
CREATE TRIGGER trig_locatie_tara
ON Locatii
FOR INSERT
AS
BEGIN
IF NOT EXISTS(SELECT * FROM inserted WHERE ID_tara IS NOT NULL)
BEGIN
RAISERROR('Nu s-a specificat tara pentru aceasta locatie!', 16, 1)
ROLLBACK TRANSACTION
END
END
```

```
insert into Locatii(nume_locatie) values ('Cluj')
```

```
Msg 50000, Level 16, State 1, Procedure trig_locatie_tara, Line 8 [Batch Start Line 474]
Nu s-a specificat tara pentru aceasta locatie!
Msg 3609, Level 16, State 1, Line 475
The transaction ended in the trigger. The batch has been aborted.
```

6. Trigger pentru tabelul "Calatori" care să verifice dacă CNP-ul introdus sau actualizat este valid

```
Msg 50000, Level 16, State 1, Procedure trig_gates_numar, Line 8 [Batch Start Line 498]
Nu s-a specificat numarul pentru aceasta poarta!
Msg 3609, Level 16, State 1, Line 499
The transaction ended in the trigger. The batch has been aborted.
```

(format din 13 cifre).



```
IF OBJECT_ID ('tr_check_cnp', 'TR') IS NOT NULL
```

```
insert into Calatori (CNP, nume, prenume, sex, email, telefon, adresa) values ('123', 'Popa', 'Irina', 'Feminin', 'popa@gmail.com', '0770658253',
```

```
DROP TRIGGER tr_check_cnp;
```

```
Msg 50000, Level 16, State 1, Procedure tr_check_cnp, Line 8 [Batch Start Line 341]
```

```
CNP-ul introdus trebuie sa contina 13 cifre!
```

```
Msg 3609, Level 16, State 1, Line 342
```

```
The transaction ended in the trigger. The batch has been aborted.
```

```
go
CREATE TRIGGER tr_check_cnp
ON Calatori
FOR INSERT, UPDATE
AS
BEGIN
    IF EXISTS(SELECT * FROM inserted WHERE LEN(CNP) != 13 )
    BEGIN
        RAISERROR('CNP-ul introdus trebuie sa contina 13 cifre!', 16, 1)
        ROLLBACK
    END
END;
```

## 7. Trigger sa previna introducerea duplicatelor in tabela "Tari"

```
IF OBJECT_ID ('trg_Tari_duplicate', 'TR') IS NOT NULL
DROP TRIGGER trg_Tari_duplicate;
```

```
go
CREATE TRIGGER trg_Tari_duplicate
ON Tari
AFTER INSERT
AS
BEGIN
    IF EXISTS (SELECT 1 FROM Tari GROUP BY nume_tara HAVING COUNT(*) > 1)
    BEGIN
        RAISERROR('Cannot insert duplicate values in the nume_tara column of
the Tari table.', 16, 1)
        ROLLBACK TRANSACTION
    END
END
```

```
insert into Tari (nume_tara) values ('Italia')
insert into Tari (nume_tara) values ('Italia')
```

```
Msg 50000, Level 16, State 1, Procedure trg_Tari_duplicate, Line 8 [Batch Start Line 360]
Cannot insert duplicate values in the nume_tara column of the Tari table.
Msg 3609, Level 16, State 1, Line 362
The transaction ended in the trigger. The batch has been aborted.
```

## 8. Trigger pentru actualizarea pretului biletelor

```
IF OBJECT_ID ('trig_update_pret_bilete', 'TR') IS NOT NULL
DROP TRIGGER trig_update_pret_bilete;
go
CREATE TRIGGER trig_update_pret_bilete
ON Bilete
AFTER UPDATE
AS
BEGIN
    UPDATE Bilete
    SET pret_bilet = i.pret_bilet
    FROM Bilete b
    INNER JOIN inserted i ON b.ID_bilet = i.ID_bilet
    WHERE b.pret_bilet < i.pret_bilet
END;

insert Bilete (pret_bilet) values (1000);

update Bilete
SET pret_bilet = 10
where pret_bilet = 1000
```

---

```
Msg 208, Level 16, State 1, Procedure trig_update_pret_bilete, Line 6 [Batch Start Line 379]
```

## 9. Trigger care să nu permită inserarea unui număr de telefon cu mai mult de 10 cifre în tabela "Calatori"

```
IF OBJECT_ID ('tr_check_telefon', 'TR') IS NOT NULL

Msg 50000, Level 16, State 1, Procedure tr_check_telefon, Line 8 [Batch Start Line 414]
Numarul de telefon introdus trebuie sa contina 10 cifre!
Msg 3609, Level 16, State 1, Line 415
The transaction ended in the trigger. The batch has been aborted.

DROP TRIGGER tr_check_telefon;
go
CREATE TRIGGER tr_check_telefon
ON Calatori
FOR INSERT, UPDATE
AS
BEGIN
    IF EXISTS (SELECT * FROM inserted WHERE LEN(CNP) != 10 )
```

```

BEGIN
    RAISERROR('Numarul de telefon introdus trebuie sa contina 10
cifre!', 16, 1)
    ROLLBACK
END
END;

insert Calatori (CNP, nume, prenume, telefon, email, adresa)
values ('1234567800101','Radu', 'Marin', '123' , 'marin@gmail.com', 'Constanta')

```

10. Trigger care să prevină inserarea unei date de plecare mai mici decât data actuală în tabelul "Zboruri"

```

IF OBJECT_ID ('trigger_data_plecure', 'TR') IS NOT NULL
DROP TRIGGER trigger_data_plecure;
go
CREATE TRIGGER trigger_data_plecure ON Zboruri
FOR INSERT, UPDATE
AS
BEGIN
IF EXISTS (SELECT * FROM inserted WHERE data_plecure < GETDATE())
BEGIN
RAISERROR ('Data plecarii nu poate fi mai mica decat data actuala.', 16, 1)
ROLLBACK TRANSACTION
END
END;

insert into Zboruri (data_plecure) values ('1993-06-23')

```

---

```

Msg 50000, Level 16, State 1, Procedure trigger_data_plecure, Line 7 [Batch Start Line 439]
Data plecarii nu poate fi mai mica decat data actuala.
Msg 3609, Level 16, State 1, Line 440
The transaction ended in the trigger. The batch has been aborted.

```

# Proceduri stocate

## 1. Procedura pentru inserarea unei noi companii în baza de date

```
IF OBJECT_ID ('dbo.InsertCompanie', 'P') IS NOT NULL
DROP PROC dbo.InsertCompanie;
go
CREATE PROCEDURE dbo.InsertCompanie
@nume_companie varchar(20)
AS
BEGIN
INSERT INTO Companii (nume_companie)
VALUES (@nume_companie)
END
```

```
EXEC dbo.InsertCompanie 'AirBoom';
```

```
select*
from Companii
```

(1 row affected)

Completion time: 2023-04-23T20:47:47.8676651+03:00

	ID_companie	nume_companie
1	6	FranceAirline
2	8	AirBoom

## 2. Procedura stocată pentru a adăuga un aeroport

```
IF OBJECT_ID ('dbo.AddAeroport', 'P') IS NOT NULL
DROP PROC dbo.AddAeroport;
go
CREATE PROCEDURE dbo.AddAeroport
@nume varchar(50),
@ID_oras int
AS
BEGIN
INSERT INTO Aeroporturi(nume, ID_oras)
VALUES (@nume, @ID_oras);
END
```

```
exec dbo.AddAeroport
@nume = 'Charles du Gaulle',
@ID_oras = 3
```

```
select*
from Aeroporturi
```

(1 row affected)

	ID_aeroport	nume	ID_oras
1	1	Charles du Gaulle	3

Completion time: 2023-04-23T21:11:56.8119861+03:00

### 3. Procedura stocată pentru a adăuga o locație

```
IF OBJECT_ID ('dbo.AddLocatie', 'P') IS NOT NULL
DROP PROC dbo.AddLocatie;
```

go

```
CREATE PROCEDURE dbo.AddLocatie
@ID_tara int,
@nume_locatie varchar(20)
AS
BEGIN
INSERT INTO Locatii(ID_tara, nume_locatie)
VALUES (@ID_tara, @nume_locatie);
END
```

```
select*
from Tari
```

```
exec dbo.AddLocatie
@ID_tara = 1,
@nume_locatie = 'Venezia'
```

```
select*
from Locatii
```

	ID_locatie	ID_tara	nume_locatie
1	3	3	Paris
2	5	1	Venezia

### 4. Procedura cre se ocupa cu tratarea erorilor IF OBJECT\_ID('CatchError', 'P') IS NOT NULL DROP PROC CatchError

```
IF OBJECT_ID('CatchError', 'P') IS NOT NULL
DROP PROC CatchError;
GO
CREATE PROC CatchError
@errorCode AS INT
AS
BEGIN
IF @errorCode <> 0
BEGIN
IF @@TRANCOUNT > 0 ROLLBACK TRAN;
PRINT 'Transaction failed with error code ' + CAST(@errorCode AS VARCHAR);
```

```
RETURN;
END;
END;
```

##### 5. Procedura stocata petru calcularea numarului de angajati care lucreaza la acelasi task

```
IF OBJECT_ID ('dbo.CountTaskEmployees', 'P') IS NOT NULL
DROP PROC dbo.CountTaskEmployees;
go
CREATE PROCEDURE dbo.CountTaskEmployees
    @TaskName VARCHAR(50)
AS
BEGIN
    SELECT COUNT(*) AS NumEmployees
    FROM StaffTasks st
    INNER JOIN Tasks t ON t.ID_task = st.ID_task
    WHERE t.nume = @TaskName;
END;

insert into StaffTasks(ID_staff, ID_task) values(4, 1)
insert into StaffTasks(ID_staff, ID_task) values(10, 1)

select*
from Tasks
select*
from Staff

EXEC dbo.CountTaskEmployees @TaskName = 'Verificare zbor'
```

	ID_task	nume	ID_zbor
1	1	Verificare zbor	4

	ID_staff	nume	prenume	email	telefon	departament
1	4	Coca	Andrei	Andrei@email.com	NULL	Bagaje
2	7	Duciuc	Cornelia	duciucc@gmail.com	07363767245	Verificare acte
3	9	Dan	Cornel	dancornel@gmail.com	07363767249	Verificare
4	10	Ciocan	Dorin	ciocan@gmail.com	0723345676	Bagaje

	NumEmployees
1	2

##### 6. Procedura stocata pentru calcularea numarului de bilete vandute

```
IF OBJECT_ID ('NumarBileteVandute', 'P') IS NOT NULL
```

```
DROP PROC NumarBileteVandute;
```

```
go
CREATE PROCEDURE NumarBileteVandute
AS
BEGIN
    SELECT Z.ID_zbor, COUNT(*) AS NumarBileteVandute
    FROM Zboruri Z
    INNER JOIN BiletePerZbor B
    ON Z.ID_zbor = B.id_zbor
    GROUP BY Z.ID_zbor
END
```

#### 7. Procedura pentru calcularea numarului de porti disponibile

```
IF OBJECT_ID('sp_NumarGatesDisponibile', 'P') IS NOT NULL
DROP PROC sp_NumarGatesDisponibile;

go
CREATE PROCEDURE sp_NumarGatesDisponibile @ID_Zbor int
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @NrGates INT;

    SELECT @NrGates = COUNT(*) FROM Gates WHERE ID_gate NOT IN
        (SELECT ID_gate FROM ZboruriGates WHERE ID_zbor = @ID_Zbor);

    SELECT @NrGates AS NumarGatesDisponibile;
END

EXEC sp_NumarGatesDisponibile @ID_Zbor = 3;
```

	NumarGatesDisponibile
1	11

#### 8. Procedura stocata pentru calcularea numarului de angajati

```
IF OBJECT_ID('CalculNumarAngajati', 'P') IS NOT NULL
DROP PROC CalculNumarAngajati;
```

```
go
CREATE PROCEDURE dbo.CalculNumarAngajati
AS
BEGIN
    SELECT COUNT(*) AS 'NumarAngajati'
    FROM Staff
END

select*
from Staff

EXEC dbo.CalculNumarAngajati
```

	NumarAngajati
1	9

9. Procedura care calculeaza suma totala a biletelor vandute per zbor.

```
IF OBJECT_ID('CalculeazaSumaVanzariBilete', 'P') IS NOT NULL
DROP PROC CalculeazaSumaVanzariBilete;

go
CREATE PROCEDURE CalculeazaSumaVanzariBilete
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @suma int;
    SELECT @suma = SUM(b.pret_bilet)
    FROM BiletePerZbor bpz
    INNER JOIN Bilete b ON bpz.id_bilet = b.ID_bilet;
    PRINT 'Suma totala din urma vanzarii biletelor este ' + CAST(@suma AS VARCHAR(20));
END
```

	ID_staff	nume	prenume	email	telefon
1	4	Coca	Andrei	Andrei@email.com	NULL
2	7	Duciuc	Cornelia	duciucc@gmail.com	07363767245
3	9	Dan	Cornel	dancornel@gmail.com	07363767249
4	10	Ciocan	Dorin	ciocan@gmail.com	0723345676

	NumarAngajati
1	4

10. Procedura stocata pentru calcularea numarului total de zboruri din aeroport

```
IF OBJECT_ID('CountFlights', 'P') IS NOT NULL
DROP PROC CountFlights;
```

```
go
CREATE PROCEDURE CountFlights
AS
BEGIN
    SELECT COUNT(*) AS TotalFlights
    FROM Zboruri;
END
```

```
select*
from Zboruri
```



EXEC CountFlights;

	ID_zbor	ID_companie	ID_avion	ora	data_plecare	durata	ID_aeroport
1	4	NULL	NULL	NULL	2024-04-07	NULL	NULL

TotalFlights	
1	1

11. Procedura care se ocupa cu comiterea tranzactiilor.

```
IF OBJECT_ID('CommitTransction', 'P') IS NOT NULL
DROP PROC CommitTransction;
GO
CREATE PROC CommitTransction
AS
BEGIN
IF @@TRANCOUNT > 0
BEGIN
COMMIT TRAN;
PRINT 'Transaction successfully committed';
END;
END;
```

# Tranzacții

## Explicite

1. Adaugarea unui calator si biletului asociat, impreuna cu clasa la care calatoreste.

```
BEGIN TRY  
BEGIN TRANSACTION;
```

```
DECLARE @ID_calator int;  
INSERT INTO Calatori (CNP, nume, prenume, sex, email, telefon, adresa)  
VALUES ('1234567890123', 'Godina', 'Catalin', 'MASCULIN', 'catalin@email.com',  
'0712345678', 'Bucuresti Blvd. Gerge Cosbuc');
```

```
SET @ID_calator = SCOPE_IDENTITY();
```

```
DECLARE @ID_clasa int;  
INSERT INTO Clase (NUME_CLASA)  
VALUES ('ECONOMIC');
```

```
SET @ID_clasa = SCOPE_IDENTITY();
```

```
DECLARE @ID_bilet int;  
INSERT INTO Bilete (ID_calator, ID_clasa, pret_bilet)  
VALUES (@ID_calator, @ID_clasa, 100);
```

```
SET @ID_bilet = SCOPE_IDENTITY();
```

```
DECLARE @ID_zbor int;  
SET @ID_zbor = 71; -- ID-ul zborului la care doriți să asociați biletul
```

```
INSERT INTO BiletePerZbor (id_bilet, id_zbor)  
VALUES (@ID_bilet, @ID_zbor);
```

```
END TRY  
begin catch  
if ERROR_NUMBER() = 2627 --error for duplicate key violation  
begin  
print 'Primary key violation';  
end  
else if ERROR_NUMBER() = 547 --constraint violations  
begin  
print 'Constraint violation';  
end
```

```

else
begin
print 'Unhandled error';
end
if @@TRANCOUNT > 0
ROLLBACK TRANSACTION
end catch

```

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-05-20T15:13:06.5013057+03:00

2. Adaugarea unui avion impreuna cu compania aeriana, unui aeroport impreuna cu locatia si tara acestuia pentru a adauga un zbor.

```
BEGIN TRY
```

```
BEGIN TRANSACTION;
```

```
INSERT INTO Companii (nume_companie) VALUES ('Air Canada');
```

```
DECLARE @ID_companie INT;
SET @ID_companie = SCOPE_IDENTITY();
```

```
INSERT INTO Avioane (ID_avion, model, capacitate, ID_companie)
VALUES (20, 'Boeing 333', 200, @ID_companie);
```

```
INSERT INTO Tari (nume_tara) VALUES ('Canada');
```

```
DECLARE @ID_tara INT;
SET @ID_tara = SCOPE_IDENTITY();
```

```
INSERT INTO Locatii (ID_tara, nume_locatie) VALUES (@ID_tara, 'Ottawa');
```

```
DECLARE @ID_locatie INT;
SET @ID_locatie = SCOPE_IDENTITY();
```

```
INSERT INTO Aeroporturi (nume, ID_oras) VALUES ('Ottawa Airport', @ID_locatie);
```

```

DECLARE @ID_aeroport INT;
SET @ID_aeroport = SCOPE_IDENTITY();

INSERT INTO Zboruri (ID_avion, ora, data_plecare, durata, ID_aeroport)
VALUES (20, '12:00', '2023-05-21', 120, @ID_aeroport);

ROLLBACK TRANSACTION;

END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    THROW;
END CATCH;

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-05-20T15:31:48.7637693+03:00

```

### 3. Modificarea unei companii aeriene:

```

BEGIN TRY
BEGIN TRAN
UPDATE Companii
SET nume_comanie = 'Air Aeroflot'
WHERE ID_comanie = 21;
COMMIT TRAN
END TRY
begin catch
if ERROR_NUMBER() = 2627 --error for duplicate key violation
begin
print 'Primary key violation';
end
else if ERROR_NUMBER() = 547 --constraint violations
begin
print 'Constraint violation';
end
else
begin
print 'Unhandled error';
end
if @@TRANCOUNT > 0

```

```
rollback tran  
end catch
```

```
(1 row affected)
```

```
Completion time: 2023-05-10T23:47:42.3121927+03:00
```

4. Actualizarea unei locatii, inserarea unei companii si a unui avion si stergerea unui aeroport .

```
BEGIN TRANSACTION;
```

```
BEGIN TRY
```

```
    UPDATE Locatii  
    SET nume_locatie = 'Cluj-Napoca'  
    WHERE ID_locatie = 2;  
    SAVE TRANSACTION Tr4;
```

```
BEGIN TRY
```

```
    INSERT INTO Companii (nume_companie) VALUES ('Blue Air');
```

```
    DECLARE @ID_c INT;  
    SET @ID_c = SCOPE_IDENTITY();
```

```
    INSERT INTO Avioane (ID_avion, model, capacitate, ID_companie)  
    VALUES (14, 'Airbus A320', 180, @ID_c);  
    COMMIT TRANSACTION Tr4;
```

```
END TRY
```

```
BEGIN CATCH
```

```
    ROLLBACK TRANSACTION;
```

```
    THROW;
```

```
END CATCH;
```

```
DELETE FROM Aeroporturi  
WHERE ID_aeroport = 3;
```

```
COMMIT TRANSACTION;
```

```
END TRY
```

```
BEGIN CATCH
```

```
BEGIN TRANSACTION
```

```
    ROLLBACK TRANSACTION;
```

```
    THROW;
```

```
END CATCH;
```

5. Adaugarea unei destinatii (tara, locatie, aeroport).

```
BEGIN TRANSACTION;

BEGIN TRY
    SAVE TRANSACTION Tr3;

    INSERT INTO Tari (nume_tara) VALUES ('Grecia');

    DECLARE @ID_t INT;
    SET @ID_t = SCOPE_IDENTITY();

    INSERT INTO Locatii (ID_tara, nume_locatie) VALUES (@ID_t, 'Atena');

    DECLARE @ID_l INT;
    SET @ID_l = SCOPE_IDENTITY();

    INSERT INTO Aeroporturi (nume, ID_oras) VALUES ('Aeroportul Eleftherios', @ID_l);

    COMMIT TRANSACTION Tr3;
END TRY
BEGIN CATCH

    ROLLBACK TRANSACTION;
    THROW;
END CATCH;

COMMIT TRANSACTION;

(1 row affected)

(1 row affected)|

(1 row affected)
```

5. Tranzactie pentru adaugarea unei locatii in tabela Locatii, cu referinta la o tara existenta:

```
BEGIN TRY

BEGIN tran
DECLARE @ID_tara INT
SELECT @ID_tara = ID_tara FROM Tari WHERE nume_tara = 'Romania'
INSERT INTO Locatii (ID_tara, nume_locatie) VALUES (@ID_tara, 'Bucuresti')
COMMIT

END TRY
begin catch
```

```

if ERROR_NUMBER() = 2627 --error for duplicate key violation
begin
print 'Primary key violation';
end
else if ERROR_NUMBER() = 547 --constraint violations
begin
print 'Constraint violation';
end
else
begin
print 'Unhandled error';
end
if @@TRANCOUNT > 0
rollback tran
end catch

```

	ID_locatie	ID_tara	nume_locatie
1	5	1	Venezia
2	7	5	Bucuresti

6. Tranzactie pentru adaugarea unui aeroport in tabela Aeroporturi, cu referinta la o locatie existenta:

```
BEGIN TRY
```

```

BEGIN tran
DECLARE @ID_locatie INT
SELECT @ID_locatie = ID_locatie FROM Locatii WHERE nume_locatie = 'Bucuresti'
INSERT INTO Aeroporturi (nume, ID_oras) VALUES ('Aeroportul Henri Coanda', @ID_locatie)
COMMIT tran

```

```

END TRY
begin catch
if ERROR_NUMBER() = 2627 --error for duplicate key violation
begin
print 'Primary key violation';
end
else if ERROR_NUMBER() = 547 --constraint violations
begin
print 'Constraint violation';
end
else
begin
print 'Unhandled error';
end
if @@TRANCOUNT > 0
rollback tran
end catch

```

	ID_aeroport	nume	ID_oras
1	2	Aeroportul Henri Coanda	7

7. Actualizarea salariului brut pentru functia de Manager din tabela Functii:

```
BEGIN TRY

BEGIN tran
UPDATE Functii SET salariu_brut = 12000 WHERE nume = 'Manager';
COMMIT

END TRY
if @@TRANCOUNT > 0
rollback tran
end catch
```

(1 row affected)

Completion time: 2023-05-10T19:53:38.5556240+03:00

8. Adaugarea unei functii noi in tabela Functii:

```
BEGIN TRY

BEGIN tran
INSERT INTO Functii (nume, salariu_brut) VALUES ('Manager', 10000);
COMMIT;

END TRY
begin catch
if ERROR_NUMBER() = 2627 --error for duplicate key violation
begin
print 'Primary key violation';
end
else if ERROR_NUMBER() = 547 --constraint violations
begin
print 'Constraint violation';
end
else
begin
print 'Unhandled error';
end
if @@TRANCOUNT > 0
rollback tran
end catch
```



(1 row affected)

Completion time: 2023-05-10T19:54:12.6354141+03:00

9. Adăugarea unui nou staff la baza de date:

```
BEGIN TRY
BEGIN tran

INSERT INTO Staff(nume, prenume, email, telefon)
VALUES ('Popescu', 'Ana', 'ana.popescu@gmail.com', '0721123456');

DECLARE @id_staff INT
SET @id_staff = @@IDENTITY

INSERT INTO Repartizare_functii(ID_staff, ID_functie)
VALUES (@id_staff, 2);

COMMIT tran
END TRY
begin catch
if ERROR_NUMBER() = 2627 --error for duplicate key violation
begin
print 'Primary key violation';
end
else if ERROR_NUMBER() = 547 --constraint violations
begin
print 'Constraint violation';
end
else
begin
print 'Unhandled error';
end
if @@TRANCOUNT > 0
rollback tran
end catch
```

	ID_staff	nume	prenume	email	telefon
1	4	Coca	Andrei	Andrei@email.com	NULL
2	7	Duciuc	Cornelia	duciucc@gmail.com	07363767245
3	9	Dan	Cornel	dancornel@gmail.com	07363767249
4	10	Ciocan	Dorin	ciocan@gmail.com	0723345676
5	12	Popescu	Ana	ana.popescu@gmail.com	0721123456

10. Achiziționarea unui bilet de avion:

```
BEGIN TRY
```

```
BEGIN tran
```

```
DECLARE @pret_bilet INT
```

```
SET @pret_bilet = 250
```

```
INSERT INTO Bilete(ID_calator, ID_clasa, pret_bilet)
```

```
VALUES (3, 1, @pret_bilet);
```

```
DECLARE @id_bilet INT
```

```
SET @id_bilet = @@IDENTITY
```

```
INSERT INTO BiletePerZbor(id_bilet, id_zbor)
```

```
VALUES (@id_bilet, 4);
```

```
COMMIT TRAN
```

```
END TRY
```

```
begin catch
```

```
if ERROR_NUMBER() = 2627 --error for duplicate key violation
```

```
begin
```

```
print 'Primary key violation';
```

```
end
```

```
else if ERROR_NUMBER() = 547 --constraint violations
```

```
begin
```

```
print 'Constraint violation';
```

```
end
```

```
else
```

```
begin
```

```
print 'Unhandled error';
```

```
end
```

```
if @@TRANCOUNT > 0
```

```
rollback tran
```

```
end catch
```

# Implicite

SET IMPLICIT\_TRANSACTIONS ON

1. Adaugarea unei tari:

```
INSERT INTO Tari (nume_tara) VALUES ('Albania');
IF @@ERROR <> 0
BEGIN
    ROLLBACK TRANSACTION;
END
ELSE
BEGIN
    COMMIT TRANSACTION;
END
```

2. Achizitionare bilet:

```
INSERT INTO Zboruri (ID_companie, ID_avion, ora, data_plecare, durata, ID_aeroport)
VALUES (6, 1, '14:30:00', '2023-05-01', 120, 1);
IF @@ERROR <> 0
BEGIN
    ROLLBACK TRANSACTION;
END
ELSE
BEGIN
    COMMIT TRANSACTION;
END
```

3. Adaugare zbor:

```
INSERT INTO Zboruri (ID_avion, ora, data_plecare, durata, ID_aeroport) VALUES ( 1,
'14:30:00', '2023-05-01', 120, 1);
IF @@ERROR <> 0
BEGIN
    ROLLBACK TRANSACTION;
END
ELSE
BEGIN
    COMMIT TRANSACTION;
END
```

#### 4. Adaugare locatie:

```
INSERT INTO Locatii (ID_tara, nume_locatie) VALUES (1, 'Bucuresti');
IF @@ERROR <> 0
BEGIN
    ROLLBACK TRANSACTION;
END
ELSE
BEGIN
    COMMIT TRANSACTION;
END
```

#### 5. Adaugare companie:

```
INSERT INTO Companii (nume_companie) VALUES ('Blue Air');
IF @@ERROR <> 0
BEGIN
    ROLLBACK TRANSACTION;
END
ELSE
BEGIN
    COMMIT TRANSACTION;
END
```

#### 6. Adaugarea unei clase:

```
INSERT INTO Clase(NUME_CLASA) VALUES ('ECONOMIC');
IF @@ERROR <> 0
BEGIN
    ROLLBACK TRANSACTION;
END
ELSE
BEGIN
    COMMIT TRANSACTION;
END
```

#### 7. Ștergerea unei funcții din baza de date:

```
DELETE FROM Functii WHERE ID_functie = 2
ROLLBACK
```

#### 8. Actualizarea unui zbor:

```
UPDATE Zboruri SET ID_companie = 2, ID_avion = 3 WHERE ID_zbor = 1
```

ROLLBACK

9.Actualizarea datelor unui calator:

```
UPDATE Calatori SET nume = 'Popescu', prenume = 'Andrei', sex = 'MASCULIN', email =  
'popescu.andrei@gmail.com' WHERE ID_calator = 1
```

ROLLBACK

10.Adaugare calatori:

```
INSERT INTO Calatori (CNP, nume, prenume, sex, email, telefon, adresa) VALUES  
( '1234567890123', 'Popescu', 'Ion', 'MASCULIN', 'popescu.ion@example.com', '0722333444',  
'Str. Victoriei, Nr. 1');  
INSERT INTO Calatori (CNP, nume, prenume, sex, email, telefon, adresa) VALUES  
( '2345678901234', 'Ionescu', 'Maria', 'FEMININ', 'ionescu.maria@example.com',  
'0722444555', 'Str. Libertatii, Nr. 2');  
COMMIT;
```

## Back-up

```
BACKUP DATABASE [GestiuneAeroport3]
TO DISK = N'C:\DataBase3\GestiuneAeroport3bak' WITH NOFORMAT,
NOINIT,
NAME = N'GestiuneAeroport3-Full Database Backup',
SKIP,
NOREWIND,
NOUNLOAD,
STATS = 10
GO

19 percent processed.
29 percent processed.
39 percent processed.
49 percent processed.
59 percent processed.
68 percent processed.
78 percent processed.
88 percent processed.
97 percent processed.
100 percent processed.
Processed 1288 pages for database 'GestiuneAeroport3', file 'PrimaryData' on file 1.
Processed 16 pages for database 'GestiuneAeroport3', file 'DataA' on file 1.
Processed 16 pages for database 'GestiuneAeroport3', file 'DataB' on file 1.
Processed 16 pages for database 'GestiuneAeroport3', file 'ArchiveData' on file 1.
Processed 2 pages for database 'GestiuneAeroport3', file 'DataLog' on file 1.
BACKUP DATABASE successfully processed 1338 pages in 0.048 seconds (217.692 MB/sec).

Completion time: 2023-05-15T19:45:27.3163149+03:00
```

## Evidenta numarului de query-uri

TIP QUERY	NUMAR
Inserare date în tabele	22
Selectare date din tabele	42
Update date din tabele	15
Ștergere date din tabele	15
Lucru cu CTE	7
Creare views	8
Creare proceduri stocate	11
Creare triggers	10
Lucru cu tranzacții	20
<b>TOTAL</b>	<b>150</b>