

Academia Tehnică Militară
„Ferdinand I”
România
Mai 2021

Aplicație pentru administrarea unui club sportiv

Baze de date

Col. prof. univ. dr. ing. Țigănescu Viorel Tudor
Dr.ing. Julian Călin
CSIII Vîlcu Dana

Sd.Cap.Neagu Alexandra-Codruța
C112B

Cuprins

Cuprins	2
Scopul Bazei de date	3
Tabele.....	4
Diagrame tabelelor	7
Obiectele bazei de date	8
TABELE	8
VIEW	15
PROCEDURI STOCATE	19
TRIGGERE.....	30
Comenzi date în baza de date.....	35
SELECT	35
INSERT	57
UPDATE.....	63
DELETE	69
TRANZACȚII.....	75
CTE.....	80

Scopul Bazei de date

Baza de date este construită pentru a manageria un club sportiv ce cuprinde mai multe sporturi individuale, mai multe persoane ce pot fii sportivi sau însoțitori, evenimentele la care au participat, dar și veniturile și facturile acestuia.

Vor exista tabele pentru Sporturi(va exista și un sport numit „toate”), Sportivi și Angajați. Deoarece atât sportivii pot participa la mai multe sporturi cât și angajații pot preda la mai multe sporturi, dar și un anumit sport poate să aiba mai mulți angajați sau mai mulți sportivi se vor crea două tabele pentru a suplini relațiilor many-to-many : InregistrariSportivi și InregistrariAngajati.

Acești sportivi pe parcursul anilor participă la diverse evenimente, astfel se va crea un tabel Evenimente, fiecare eveniment va putea fi de mai multe tipuri(competiții, excursii, cantonamente) dar și pentru un anumit sport(inclusiv pentru „toate”). Fiindcă atât un sportiv cât și un angajat poate participa la mai multe evenimente dar și un eveniment poate avea mai multe persoane se vor crea tabele noi pentru a manageria aceste relații: Între Evenimente și Sportivi va exista tabela Premii ce va ține cont de ce loc(se va trece participare dacă nu a primit loc pe podium) a luat la o anumită competiție, categoria, dar și observații legate de acestea. Între evenimente și angajați va fi tabela Însoțitori.

La fiecare eveniment se vor plăti taxe, astfel se va crea tabela TaxeEvenimente ce va conține date referitoare la valoarea sau data plăților. Pentru un eveniment se pot plăti mai multe taxe, în diferite zile. Sportivi vor trebui să plătească niște taxe lunare, informații ce se vor stoca în tabela TaxeSportivi. În privința angajaților va exista tabela Salarii ce va urmări salariul brut și valoarea impozitului între două plăți.

Clubul cheltuiește bani și pe întreținerea centrului de activitate(considerând că există un singur centru), se va crea pentru acestea tabela Facturi dar și TipFacturi ce vor urmări plățile lunare.

Veniturile se vor accesa prin tabela Venituri, acestea pot fi de mai multe tipuri dar și de la diverse surse, pentru acestea se vor crea tabelele Venituri, SursaVenituri și TipVenituri.

Diagrama este împărțită în 3 părți, partea ce ține de întreținerea veniturilor, cea pentru facturi și ultima și cea mai complexă ce se ocupă de sportivi, evenimente, angajați și restul atașamentelor față de acestea.

Tabele

Angajat:

- IDAngajat - primary key int not null
- Nume - nvarchar(50) not null
- Prenume - nvarchar(50) not null
- CNP - char(13) not null – constrângere de lungime fixă, 13 caractere
- DataIncepereContract - date
- DataFinalizareContract - date
- IdTip – int not null foreign key -> TipAngajat

Evenimente

- IDeveniment – primary key int not null
- Nume – nvarchar(50) not null
- Nume Club Organizator nvarchar(50) not null
- Locatie – nvarchar(50) not null
- DataIncepere – date not null
- IDTip – int not null foreign key -> TipEveniment
- DataFinalizare – date
- IDTipSport – int not null foreign key -> Sporturi

Facturi

- IDFactura – primary key int not null
- Data – date not null
- Valoare – float not null
- IDTip – int not null foreign key -> TipFacturi

InregistrariAngajati

- IDInregistrare – primary key int not null
- IDSport – int not null foreign key -> Sporturi
- IDAngajat – int not null foreign key -> Angajati

InregistrariSportiv

- IDInregistrare – primary key int not null
- IDSport – int not null foreign key -> Sporturi
- IDSportiv – int not null foreign key -> Sportivi
- DataIncepere – date
- DataFinalizare – date

Insotitori

- IDInregistrare – primary key int not null
- IDAngajat – int not null foreign key -> Angajati
- IDEveniment – int not null foreign key -> Evenimente

Premii

- IDPremiu – primary key int not null
- IDSportiv – int not null foreign key -> Sportivi
- IDEveniment – int not null foreign key -> Evenimente
- Premiu - nchar(20)

Tabele

- Categorie – nvarchar(100)
- Observatii – nvarchar(500)

Salarii

- IDSalariu – primary key int not null
- Valoare Bruta – float not null
- Valoare Impozit – float not null
- IDAngajat – int not null foreign key -> Angajat
- DeLaData – date
- LaData – date

Sportivi

- IDSportivi – primary key int not null
- Nume – nvarchar(50) not null
- Prenume – nvarchar(50) not null
- CNP – char(13) not null - constrângere de lungime fixă, 13 caractere
- DataInscriere – date
- DataPlecare – date
- Observatii – nvarchar(500)

Sporturi

- IDSport – primary key int not null
- Denumire – nvarchar(50) not null

SursaVenituri

- IDSursa – primary key int not null
- Nume – nvarchar(100) not null

TaxeEvenimente

- IDTaxa – primary key int not null
- IDEveniment – int not null foreign key -> Evenimente
- Valoare – float not null
- Data – date not null
- Observatii – nvarchar(100)

TaxeSportivi

- IDTaxa – primary key int not null
- IDSportiv – int not null foreign key -> Sportivi
- Valoare – float not null
- Data – date not null

TipAngajat

- IDTip – primary key int not null
- Denumire – nvarchar(50) not null

TipEveniment

- IDTip – primary key not null
- Denumire – nvarchar(50) not null

Tabele

TipFactura

- IDTip – primary key int not null
- Denumire – nvarchar(100) not null

TipVenituri

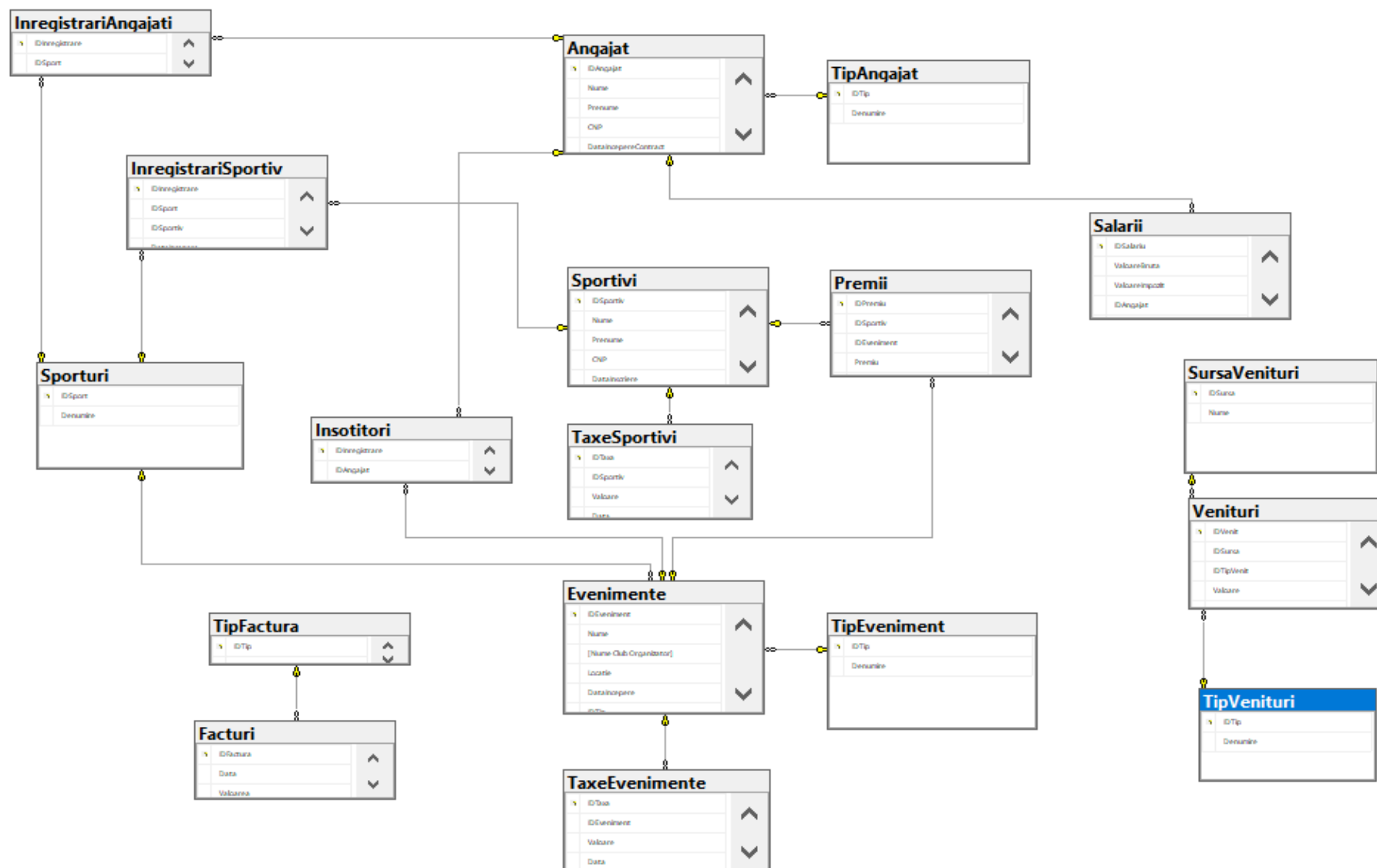
- IDTip – primary key int not null
- Denumire – nvarchar(100) not null

Venituri

- IDVenit – primary key int not null
- IDSursa – int not null foreign key -> SursaVenituri
- IDTipVenit – int not null foreign key -> TipVenituri
- Valoare – float not null
- Data – date not null

Diagrame tabelor

Figură 1



Obiectele bazei de date

TABELE

1. Să se creeze baza de date având un fișier de date master și unul de log.

```
CREATE DATABASE CLUB_SPORTIV
ON PRIMARY
(
    NAME = N'CLUB_SPORTIV',
    FILENAME =
N'D:\sql\MSSQL15.MSSQLSERVER\MSSQL\DATA\CLUB_SPORTIV.mdf' ,
    SIZE = 10MB ,
    MAXSIZE = unlimited,
    FILEGROWTH = 1MB
),
(
    NAME = N'CLUB_SPORTIV_n',
    FILENAME =
N'D:\sql\MSSQL15.MSSQLSERVER\MSSQL\DATA\CLUB_SPORTIV_n.ndf' ,
    SIZE = 10MB ,
    MAXSIZE = unlimited,
    FILEGROWTH = 1MB
),
(
    NAME = N'CLUB_SPORTIV_r',
    FILENAME =
N'D:\sql\MSSQL15.MSSQLSERVER\MSSQL\DATA\CLUB_SPORTIV_r.mdf' ,
    SIZE = 10MB ,
    MAXSIZE = unlimited,
    FILEGROWTH = 1MB
)
LOG ON
(
    NAME = N'CLUB_SPORTIV_log',
    FILENAME =
N'D:\sql\MSSQL15.MSSQLSERVER\MSSQL\DATA\CLUB_SPORTIV_log.ldf' ,
    SIZE = 10MB ,
    MAXSIZE = unlimited ,
    FILEGROWTH = 1GB
),
(
    NAME = N'CLUB_SPORTIV_log2',
    FILENAME =
N'D:\sql\MSSQL15.MSSQLSERVER\MSSQL\DATA\CLUB_SPORTIV_R_log.ldf' ,
    SIZE = 10MB ,
    MAXSIZE = unlimited ,
    FILEGROWTH = 1GB
)
```


2. Să se creeze toate tabelele pentru baza de date după diagrama prezentată([Figura 1](#)).

```

if OBJECT_ID('Angajat','u') is not null
    drop table Angajat
go
CREATE TABLE Angajat
(
    IDAngajat int IDENTITY(1,1) NOT NULL,
    Nume nvarchar(50) NOT NULL,
    Prenume nvarchar(50) NOT NULL,
    CNP char(13) NOT NULL,
    DataIncepereContract date NULL,
    DataFinalizareContract date NULL,
    IdTip int NOT NULL,
    CONSTRAINT [PK_Angajat] PRIMARY KEY(IDAngajat)
)

if OBJECT_ID('Evenimente','u') is not null
    drop table Evenimente
go
CREATE TABLE Evenimente(
    IDEveniment int IDENTITY(1,1) NOT NULL,
    Nume nvarchar(50) NOT NULL,
    [Nume Club Organizator] nvarchar(50) NOT NULL,
    Locatie nvarchar(50) NOT NULL,
    DataIncepere date NOT NULL,
    IDTip int NOT NULL,
    DataFinalizare date NULL,
    IDTipSport int NOT NULL,
    CONSTRAINT PK_Evenimente PRIMARY KEY(IDEveniment)
)

if OBJECT_ID('Facturi','u') is not null
    drop table Facturi
go
CREATE TABLE Facturi(
    IDFactura [int] IDENTITY(1,1) NOT NULL,
    Data date NOT NULL,
    Valoarea float NOT NULL,
    IDTip int NOT NULL,
    CONSTRAINT PK_MiscareFinanciara PRIMARY KEY(IDFactura)
)

if OBJECT_ID('InregistrariAngajati','u') is not null
    drop table InregistrariAngajati
go
CREATE TABLE InregistrariAngajati(
    IDInregistrare int IDENTITY(1,1) NOT NULL,
    IDSport int NOT NULL,

```

```

        IDAngajat int NOT NULL,
        CONSTRAINT PK_InregistrariAngajati PRIMARY KEY(IDInregistrare)
    )

if OBJECT_ID('InregistrariSportiv','u') is not null
    drop table InregistrariSportiv
go
CREATE TABLE InregistrariSportiv(
    IDInregistrare int IDENTITY(1,1) NOT NULL,
    IDSport int NOT NULL,
    IDSportiv int NOT NULL,
    DataIncepere date NULL,
    DataFinalizare date NULL,
    CONSTRAINT PK_InregistrariSportiv PRIMARY KEY(IDInregistrare)
)

if OBJECT_ID('Insotitori','u') is not null
    drop table Insotitori
go
CREATE TABLE Insotitori(
    IDInregistrare int IDENTITY(1,1) NOT NULL,
    IDAngajat int NOT NULL,
    IDEveniment int NOT NULL,
    CONSTRAINT PK_Insotitori PRIMARY KEY (IDInregistrare)
)

if OBJECT_ID('Premii','u') is not null
    drop table Premii
go
CREATE TABLE Premii(
    IDPremiu int IDENTITY(1,1) NOT NULL,
    IDSportiv int NOT NULL,
    IDEveniment int NOT NULL,
    Premiu nchar(20) NULL,
    Categorie nvarchar(100) NOT NULL,
    Observatii nvarchar(500) NULL,
    CONSTRAINT PK_Premii PRIMARY KEY(IDPremiu)
)

if OBJECT_ID('Salarii','u') is not null
    drop table Salarii
go
CREATE TABLE Salarii(
    IDSalariu int IDENTITY(1,1) NOT NULL,
    ValoareBruta float NOT NULL,
    ValoareImpozit float NOT NULL,
    IDAngajat int NOT NULL,
    DeLaData date NULL,

```

```

        LaData date NULL,
        CONSTRAINT PK_Salarii PRIMARY KEY(ID)
    )

if OBJECT_ID('Sporturi','u') is not null
    drop table Sporturi
go
CREATE TABLE Sporturi(
    IDSport int IDENTITY(1,1) NOT NULL,
    Denumire nvarchar(50) NOT NULL,
    CONSTRAINT PK_Sporturi PRIMARY KEY(IDSport)
)

if OBJECT_ID('Sportivi','u') is not null
    drop table Sportivi
go
CREATE TABLE Sportivi(
    IDSportiv int IDENTITY(1,1) NOT NULL,
    Nume nvarchar(50) NOT NULL,
    Prenume nvarchar(50) NOT NULL,
    CNP char(13) NOT NULL,
    DataInscriere date NOT NULL,
    DataPlecare date NULL,
    Observatii nvarchar(500) NULL,
    CONSTRAINT PK_Sportiv PRIMARY KEY (IDSportiv)
)

if OBJECT_ID('SursaVenituri','u') is not null
    drop table SursaVenituri
go
CREATE TABLE SursaVenituri(
    IDSursa int IDENTITY(1,1) NOT NULL,
    Nume nvarchar(100) NOT NULL,
    CONSTRAINT PK_SursaVenitur PRIMARY KEY(IDSursa)
)

if OBJECT_ID('TaxeEvenimente','u') is not null
    drop table TaxeEvenimente
go
CREATE TABLE TaxeEvenimente(
    IDTaxa int IDENTITY(1,1) NOT NULL,
    IDEveniment int NOT NULL,
    Valoare float NOT NULL,
    Data date NOT NULL,
    Observatii nvarchar(100) NULL,
    CONSTRAINT PK_TaxeEvenimente PRIMARY KEY(IDTaxa)
)

```

```

if OBJECT_ID('TaxeSportivi','u') is not null
    drop table TaxeSportivi
go
CREATE TABLE TaxeSportivi(
    IDTaxa int IDENTITY(1,1) NOT NULL,
    IDSportiv int NOT NULL,
    Valoare float NOT NULL,
    Data date NOT NULL,
    CONSTRAINT PK_TaxeSportivi PRIMARY KEY(IDTaxa)
)

if OBJECT_ID('TipAngajat','u') is not null
    drop table TipAngajat
go
CREATE TABLE TipAngajat(
    IDTip int IDENTITY(1,1) NOT NULL,
    Denumire nvarchar(50) NOT NULL,
    CONSTRAINT PK_TipAngajat PRIMARY KEY(IDTip)
)

if OBJECT_ID('TipEveniment','u') is not null
    drop table TipEveniment
go
CREATE TABLE TipEveniment(
    IDTip int IDENTITY(1,1) NOT NULL,
    Denumire nvarchar(50) NOT NULL,
    CONSTRAINT PK_TipEveniment PRIMARY KEY(IDTip)
)

if OBJECT_ID('TipFactura','u') is not null
    drop table TipFactura
go
CREATE TABLE TipFactura(
    IDTip int IDENTITY(1,1) NOT NULL,
    Denumire nvarchar(100) NOT NULL,
    CONSTRAINT PK_TipFactura PRIMARY KEY(IDTip)
)

if OBJECT_ID('TipVenituri','u') is not null
    drop table TipVenituri
go
CREATE TABLE TipVenituri(
    IDTip int IDENTITY(1,1) NOT NULL,
    Denumire nvarchar(100) NOT NULL,
    CONSTRAINT PK_TipSursaVenituri PRIMARY KEY(IDTip)
)

```

```

if OBJECT_ID('Venituri','u') is not null
    drop table Venituri
go
CREATE TABLE Venituri(
    IDVenit int IDENTITY(1,1) NOT NULL,
    IDSursa int NOT NULL,
    IDTipVenit int NOT NULL,
    Valoare float NOT NULL,
    Data date NOT NULL,
    CONSTRAINT PK_Venitur PRIMARY KEY(IDVenit)
)
3. Să se adauge cheile externe la toate tabele.
4.
ALTER TABLE Angajat
ADD CONSTRAINT FK_Angajat_TipAngajat FOREIGN KEY (IdTip)
REFERENCES TipAngajat(IDTip)
GO
ALTER TABLE [dbo].[Evenimente] ADD DEFAULT ((5)) FOR [IDTipSport]
GO
ALTER TABLE [dbo].[Evenimente]
ADD CONSTRAINT [FK_Evenimente_Evenimente] FOREIGN KEY([IDTipSport])
REFERENCES [dbo].[Sporturi] ([IDSport])
GO
ALTER TABLE [dbo].[Evenimente]
ADD CONSTRAINT [FK_Evenimente_TipEveniment] FOREIGN KEY([IDTip])
REFERENCES [dbo].[TipEveniment] ([IDTip])
GO
ALTER TABLE [dbo].[Facturi]
ADD CONSTRAINT [FK_Facturi_TipFactura] FOREIGN KEY([IDTip])
REFERENCES [dbo].[TipFactura] ([IDTip])
GO
ALTER TABLE [dbo].[InregistrariAngajati]
ADD CONSTRAINT [FK_InregistrariAngajati_Angajat] FOREIGN KEY([IDAngajat])
REFERENCES [dbo].[Angajat] ([IDAngajat])
GO
ALTER TABLE [dbo].[InregistrariAngajati]
ADD CONSTRAINT [FK_InregistrariAngajati_Sporturi] FOREIGN KEY([IDSport])
REFERENCES [dbo].[Sporturi] ([IDSport])
GO
ALTER TABLE [dbo].[InregistrariSportiv]
ADD CONSTRAINT [FK_InregistrariSportiv_Sportivi] FOREIGN KEY([IDSportiv])
REFERENCES [dbo].[Sportivi] ([IDSportiv])
GO
ALTER TABLE [dbo].[InregistrariSportiv]
ADD CONSTRAINT [FK_InregistrariSportiv_Sporturi] FOREIGN KEY([IDSport])
REFERENCES [dbo].[Sporturi] ([IDSport])
GO
ALTER TABLE [dbo].[Premii]

```

```

ADD CONSTRAINT [FK_Premii_Evenimente] FOREIGN KEY([IDEveniment])
REFERENCES [dbo].[Evenimente] ([IDEveniment])
GO
ALTER TABLE [dbo].[Premii]
ADD CONSTRAINT [FK_Premii_Sportivi] FOREIGN KEY([IDSportiv])
REFERENCES [dbo].[Sportivi] ([IDSportiv])
GO
ALTER TABLE [dbo].[Salarii]
ADD CONSTRAINT [FK_Salarii_Angajat] FOREIGN KEY([IDAngajat])
REFERENCES [dbo].[Angajat] ([IDAngajat])
GO
ALTER TABLE [dbo].[TaxeEvenimente]
ADD CONSTRAINT [FK_TaxeEvenimente_Evenimente] FOREIGN KEY([IDEveniment])
REFERENCES [dbo].[Evenimente] ([IDEveniment])
GO
ALTER TABLE [dbo].[TaxeSportivi]
ADD CONSTRAINT [FK_TaxeSportivi_Sportivi] FOREIGN KEY([IDSportiv])
REFERENCES [dbo].[Sportivi] ([IDSportiv])
GO
ALTER TABLE [dbo].[Venituri]
ADD CONSTRAINT [FK_Venituri_SursaVenituri] FOREIGN KEY([IDSursa])
REFERENCES [dbo].[SursaVenituri] ([IDSursa])
GO
ALTER TABLE [dbo].[Venituri]
ADD CONSTRAINT [FK_Venituri_TipSursaVenituri] FOREIGN KEY([IDTipVenit])
REFERENCES [dbo].[TipVenituri] ([IDTip])
GO
ALTER TABLE [dbo].[Insotitori]
ADD CONSTRAINT [FK_Insotitori_Angajat] FOREIGN KEY([IDAngajat])
REFERENCES [dbo].[Angajat] ([IDAngajat])
GO
ALTER TABLE [dbo].[Insotitori]
ADD CONSTRAINT [FK_Insotitori_Evenimente] FOREIGN KEY([IDEveniment])
REFERENCES [dbo].[Evenimente] ([IDEveniment])
GO
5. Să se adauge constrângerile de CNP la Sportivi și Angajați
ALTER TABLE [dbo].[Sportivi]
ADD CONSTRAINT CNPCheck CHECK(len(CNP) = 13)
go
ALTER TABLE [dbo].[Angajat]
ADD CONSTRAINT CNPCheckA CHECK(len(CNP) = 13)
go

```

VIEW

6. După diverse ștergeri din Tabelul Sporturi sau chiar update-uri nerealizate există posibilitatea ca unii sportivi care nu mai sunt înscriși la nici un sport să apară ca și cum ar fi.

Să se facă un view care afișează pentru fiecare astfel de sportiv id-ul și ultima dată la care a fost activ la un sport.

Acest view va fi folosit la triggerul de ștergere a unui sport, sau la plecarea unui sportiv de la un anumit sport pentru a ști ce înregistrări necesită modificări după anumite acțiuni.

```
if OBJECT_ID('AdaugareDataPlecareSportivi','v') is not null
    drop view AdaugareDataPlecareSportivi
go
create view AdaugareDataPlecareSportivi as
with c as
(
    select S.IDSportiv as ID ,count(*) as NrSporturiTotale,
           count (I.DataFinalizare) as NrSporturiNeparticipante,
           MAX(I.DataFinalizare) as UltimaData
    from InregistrariSportiv as I
    inner join Sportivi as S
    on S.IDSportiv=I.IDSportiv
    group by S.IDSportiv
)
select c.ID, c.UltimaData from c
inner join Sportivi
on Sportivi.IDSportiv = c.ID
where c.NrSporturiTotale=c.NrSporturiNeparticipante and
Sportivi.DataPlecare is null
```

În cazul de față există un singur sportiv ce îndeplinește condițiile.

	ID	UltimaData
1	42	2017-05-15

7. Afișați numele, prenumele, sexul, numărul de sporturi al sportivului care a participat la cele mai multe sporturi de la înscriere.

```
if OBJECT_ID('CelMaiSportivDintreSportivi','v') is not null
    drop view CelMaiSportivDintreSportivi
go
create view CelMaiSportivDintreSportivi as
with c as
(
    select IDSportiv, count(*) as nrsporturi from InregistrariSportiv
    group by IDSportiv
),
b as
(
    select max(c.nrsporturi) as maxim from c
)
```

```
select s.Nume+' '+s.Prenume as Nume,
case when s.CNP like '2%' then 'Feminin'
      when s.CNP like '1%' then 'Masculin'
      when s.CNP like '5%' then 'Masculin'
      when s.CNP like '6%' then 'Feminin'
      else 'Nedefinit'
end as Gen,
c.nrsporturi as NrSporturi
from c
inner join Sportivi as s
on s.IDSportiv=c.IDSportiv
inner join b
on c.nrsporturi=b.maxim
where c.nrsporturi=b.maxim
```

Rezultatul în momentul actual este

	Nume	Gen	NrSporturi
1	Negoiu Cristian	Masculin	3

8. Afișați id-ul, numele, prenumele și numărul de premii 1,2,3 respectiv participare al fiecarui sportiv primite la competiții de la toate sporturile.

```
if OBJECT_ID('LocuriCompetitiiPerSportiv','V') is not null
drop view LocuriCompetitiiPerSportiv
go
create view LocuriCompetitiiPerSportiv as
(
    SELECT S.IDSportiv as ID ,s.Nume as Nume, S.Prenume as Prenume,
    COUNT(case when P.Premiu like '%1%' then 5 end) as locul1,
    COUNT(case when P.Premiu like '%2%' then 3 end) as locul2,
    COUNT(case when P.Premiu like '%3%' then 2 end) as locul3,
    COUNT(case when P.Premiu like '%participare%' then 1 end) as participare
    from Sportivi AS S
    INNER JOIN Premii AS P
    ON P.IDSportiv=S.IDSportiv
    INNER JOIN Evenimente AS E
    ON E.IDEveniment=P.IDEveniment
    INNER JOIN TipEveniment AS T
    ON T.IDTip =E.IDTip
    WHERE T.Denumire LIKE 'Competitie'
    group by s.Nume, s.Prenume, s.IDSportiv
)
```


Obiectele bazei de date VIEW

	ID	Nume	Prenume	locul1	locul2	locul3	participare
1	1	Popescu	Stefan	7	3	0	0
2	2	Blaga	Ana-Maria	4	1	0	0
3	3	Stoica	Andrei	3	5	1	0
4	4	Stoica	Andreea	1	3	3	1
5	5	Margarit	Norbert	0	0	0	1
6	6	Arsene	Vlad	3	0	0	0
7	7	Arsene	Nicolae	0	0	1	0
8	11	Ghenea	Diana	0	0	1	0
9	12	Negoiu	Cristian	0	0	0	2
10	16	Negoiu	Raluca	4	3	1	0
11	18	Manea	Robert	0	1	2	1
12	19	Ionescu	Roberta	1	0	0	1
13	22	Anolo	Mihai	0	0	1	1
14	24	Coroie	Narcisa	1	0	0	0
15	26	Manole	Alexandru	0	0	1	0
16	27	Amz	Catalin	0	0	0	1
17	31	Lipan	Oana	0	1	0	0
18	34	Mihalache	Cristina	1	1	0	1

9. Afișați numele, prenumele, sexul, numărul de competiții al celui mai activ angajat din club.

```

if OBJECT_ID('CelMaiActivAngajatiCompetitii','v') is not null
    drop view CelMaiActivAngajatiCompetitii
go
create view CelMaiActivAngajatiCompetitii as

with c as
(
    select A.Nume, A.Prenume,
    case when A.CNP like '2%' then 'Feminin'
         when A.CNP like '1%' then 'Masculin'
         when A.CNP like '5%' then 'Masculin'
         when A.CNP like '6%' then 'Feminin'
         else 'Nedefinit'
    end as Gen, count(*) as NrCompetitii from Insotitori as I
    inner join Angajat as A
    on A.IDAngajat=I.IDAngajat
    inner join Evenimente as E
    on E.IDEveniment=I.IDEveniment
    inner join TipEveniment as T
    on T.IDTip=E.IDTip
    where T.Denumire like 'Competitie'
    group by A.Nume, A.Prenume, A.CNP
)
select c.Nume, c.Prenume, c.Gen, c.NrCompetitii from c
where c.NrCompetitii = (select max(c.NrCompetitii) from c)

```

	Nume	Prenume	Gen	NrCompetitii
1	Copu	Marian	Masculin	9

10. Să se creze un view ce conține sportivul cu cele mai multe puncte de la fiecare sport.

```
if OBJECT_ID('CelMaiBunSportivSport','v') is not null
    drop view CelMaiBunSportivSport
go
create view CelMaiBunSportivSport as
with P as(
    select S.IDSportiv as id,s.Nume as Nume, S.Prenume as Prenume,
    SP.Denumire as Denumire,
        COUNT(case when P.Premiu like '%1%' then 5 end) as locul1,
        COUNT(case when P.Premiu like '%2%' then 3 end) as locul2,
        COUNT(case when P.Premiu like '%3%' then 2 end) as locul3,
        COUNT(case when P.Premiu like '%participare%' then 1 end) as participare
    from Premii as P
    inner join sportivi as S
    on P.IDSportiv=S.IDSportiv
    inner join InregistrariSportiv as I
    on I.IDSportiv=S.IDSportiv
    inner join Sporturi as SP
    on I.IDSport =SP.IDSport
    inner join Evenimente as E
    on E.IDEveniment=P.IDEveniment and E.IDTipSport=SP.IDSport
    group by SP.Denumire,S.IDSportiv, S.Nume, S.Prenume
),a as
(
    select id,Nume, Prenume,Denumire,
    (locul1*5+locul2*3+locul3*2+participare) as punctaj from P
),c as
(
    select Denumire, max(punctaj) as PunctajMaxim from a
    group by Denumire
)
select a.Denumire, a.Nume+' '+ A.Prenume as Nume, c.PunctajMaxim from a inner join c
on a.Denumire=c.Denumire
where a.punctaj=c.PunctajMaxim
```

	Denumire	Nume	PunctajMaxim
1	Ju.Jitsu	Popescu Stefan	44
2	Box	Arsene Vlad	5
3	Karate	Arsene Vlad	5
4	Box	Negoiu Raluca	5
5	BJJ	Manea Robert	3
6	MMA	Coroie Narcisa	5

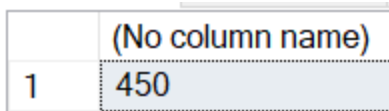
PROCEDURI STOCATE

11. Creați o procedură stocată ce va returna valoarea taxelor dintre două date, datastart și datasfarsit.

```
if OBJECT_ID('TaxeIntreDate','p') is not null
    drop proc TaxeIntreDate
go
create proc TaxeIntreDate(
    @dateinceput date,
    @datesfarsit date,
    @suma int=0 output)
as begin
(
    select @suma=SUM(Valoare) from TaxeSportivi
    where Data>@dateinceput and Data <@datesfarsit
)
return;
end
```

Testarea va fi:

```
declare @suma int=0;
exec TaxeIntreDate
    @dateinceput ='20120101',
    @datesfarsit ='20120201',
    @suma=@suma output;
set @suma=@suma
select @suma
```



(No column name)	
1	450

12. Afișați fluxul monetar între data x și data y.

```
if OBJECT_ID('ProfitIntreDate','p') is not null
    drop proc ProfitIntreDate
go
create proc ProfitIntreDate(
    @datestart date,
    @dateend date,
    @counter int=0,
    @maxid int=0,
    @total int =0 output)
as begin
    select @total=0, @counter=min(IDAngajat), @maxid=max(IDAngajat) from Angajat
    while(@counter is not null and @counter<=@maxid)
    begin
        declare @suma int =0;
        exec ValoareSalariiTotal
            @idangajat=@counter,
```

```
        @startdata=@datestart,  
        @enddata=@dateend,  
        @valsalar=@suma output;  
    if(@suma is null)  
        set @suma=0;  
    set @total=@total+@suma  
    set @counter=@counter+1;  
end;  
with c(valoare) as  
(  
    select Valoarea from Facturi  
    where Data>@datestart and Data<@dateend  
    union all  
    select Valoare from TaxeEvenimente  
    where Data>@datestart and Data<@dateend  
)  
select @suma=SUM(valoare) from c  
set @total=@total+@suma  
set @suma=0;  
with c (Valoare) as  
(  
    select Valoare from TaxeSportivi  
    union all  
    select Valoare from Venituri  
)  
select @suma=SUM(Valoare) from c  
set @total=(@total-@suma)*(-1)  
return;  
end;
```

Testare:

```
declare @tot int =0  
exec ProfitIntreDate  
    @datestart ='2016-01-01',  
    @dateend ='2016-12-31',  
    @total =@tot output  
select @tot
```

(No column name)	
1	26920

13. Creați o procedură stocată ce returnează anul în care s-au luat cele mai multe premii de către un anumit sportiv. În cazul în care există mai mulți ani cu același număr de premii să se returneze anul cel mai recent.

```
if OBJECT_ID('AnNumarPremiiMaximSportiv','p') is not null
    drop proc AnNumarPremiiMaximSportiv
go
create proc AnNumarPremiiMaximSportiv(
    @nume nvarchar(30),
    @prenume nvarchar(30),
    @an nvarchar(10) = '2010' output)
as begin
    with c as
    (
        select p.IDSportiv, count(*) as nrpremii, YEAR(e.DataIncepere) as an from
Premii as p
        inner join Sportivi as S
        on S.IDSportiv=P.IDSportiv
        inner join Evenimente as E
        on E.IDEveniment=P.IDEveniment
        where s.Nume like @nume and s.Prenume like @prenume
        group by p.IDSportiv, e.DataIncepere
    ),maxan as
    (
        select MAX(nrpremii) as man from c
    )
    select @an= MAX(c.an) from c
    inner join maxan
    on c.nrpremii=maxan.man
    return;
end
declare @anmax nvarchar(10)='';
exec AnNumarPremiiMaximSportiv
    @nume = 'Popescu',
    @prenume = 'Stefan',
    @an=@anmax output;
select @anmax;
```

	(No column name)
1	2017

14. Creați o procedură stocată ce va returna locația ultimului eveniment la care a participat sportivul cu numele și prenumele trimis ca parametru.

```
if OBJECT_ID('LocatiaEvenimentSportiv','p') is not null
    drop proc LocatiaEvenimentSportiv
go
create proc LocatiaEvenimentSportiv(
    @nume nvarchar(30),
    @prenume nvarchar(30),
    @locatia nvarchar(30) output)
as begin
    select top 1 @locatia=e.Locatie from Sportivi as S
    inner join Premii as P
    on S.IDSportiv=p.IDSportiv
    inner join Evenimente as E
    on E.IDEveniment=p.IDEveniment
    where S.Nume=@nume and S.Prenume=@prenume
    order by e.DataFinalizare desc
return;
end
```

Testare:

```
declare @loc nvarchar(20)='';
exec LocatiaEvenimentSportiv
    @nume='Popescu',
    @prenume='Stefan',
    @locatia=@loc output;
select @loc
```

	(No column name)
1	Cluj

15. Creați o procedură stocată ce returnează totalul banilor dați pe o competiție, ca parametrii se vor trimite numele, locația și data.

```
if OBJECT_ID('BaniPerCompetitie','P') is not null
    drop proc BaniPerCompetitie
go
create proc BaniPerCompetitie(
    @nume nvarchar(30),
    @locatie nvarchar(30),
    @data date,
    @suma int =0 output)
as begin
    select @suma= SUM(T.Valoare) from TaxeEvenimente as T
    inner join Evenimente as E
    on T.IDEveniment=E.IDEveniment
    where E.Nume like @nume and E.Locatie like @locatie
    and @data >=E.DataIncepere and @data<=E.DataFinalizare
return;
end
```

Testare:

```
declare @valoare int =0;
exec BaniPerCompetitie
    @nume='Cupa Junior',
    @locatie='Constanta',
    @data='2017-02-28',
    @suma=@valoare output;
select @valoare
```

(No column name)	
1	500

16. Creați o procedură stocată ce returnează suma totală de bani dată pe un anumit tip de factură(de ex Gaz, Curent).

```
if OBJECT_ID('BaniFactura', 'p') is not null
    drop proc BaniFactura
go
create proc BaniFactura(
    @denumire nvarchar(30),
    @suma int =0 output)
as begin
    select @suma=SUM(F.Valoarea) from Facturi as F
    inner join TipFactura as T
    on F.IDTip=T.IDTip
    where T.Denumire like @denumire
return;
end
```

Testare:

```
declare @sum int =0;
exec BaniFactura
    @denumire='Gaz',
    @suma=@sum output
select @sum
```

(No column name)	
1	13340

17. Creați o procedură stocată care returnează numărul de sportivi și numărul de însoțitori participanți la un anumit eveniment.

```
if OBJECT_ID ('NrParticipantiCompetitie','P') is not null
    drop proc NrParticipantiCompetitie
go
create proc NrParticipantiCompetitie(
    @nume nvarchar(30),
    @locatie nvarchar(30),
    @data date,
    @nrangajati int =0 output,
    @nrsportivi int =0 output)
as begin
with c as(
    select count(*) as nr from Premii as P
    inner join Evenimente as E
    on P.IDEveniment=E.IDEveniment
    where E.Nume like @nume and E.Locatie like @locatie
        and @data >=E.DataIncepere and @data<=E.DataFinalizare
    group by P.IDSportiv
    )
    select @nrsportivi=count(*) from c

    select @nrangajati= count(*) from Insotitori as I
    inner join Evenimente as E
    on E.IDEveniment=I.IDEveniment
    where E.Nume like @nume and E.Locatie like @locatie
        and @data >=E.DataIncepere and @data<=E.DataFinalizare

return;
end
```

Testare:

```
declare @sportivi int=0;
declare @angajati int=0;
exec NrParticipantiCompetitie
@nume='Cupa Junior',
@locatie='Constanta',
@data='2017-02-28',
@nrSportivi=@sportivi output,
@nrAngajati=@angajati output
select @sportivi as 'NrSportivi', @angajati as 'NrInsotitori'
```

	NrSportivi	NrInsotitori
1	3	1

18. Să se introducă taxele de participare la o competiție pentru toți sportivii înscriși. Se vor trimite ca parametrii numele competiției și taxa per probă. Se specifică ca taxele se vor calcula ca și numărul de probe la care a participat sportivul (numărul de premii) înmulțit cu prețul taxei. Se va insera această taxă în tabela aferentă cu data competiției (având în vedere că baza de date are majoritatea datelor până în anul 2017 se va folosi data competiției, în cazul în care aceasta aplicație ar fi folosită în viața reală s-ar folosi data curentă). Această sumă se va returna pentru a putea fi observată. Se va avea grijă la erorile ce pot apărea în cazul în care competiția nu există, se va afișa numele competiției și un mesaj aferent, cât și ultima competiție introdusă deoarece este mai probabil să fie vorba de aceasta.

```
if OBJECT_ID('IntroducereTaxeEveniment', 'p') is not null
    drop proc IntroducereTaxeEveniment
go
create proc IntroducereTaxeEveniment(
    @nume nvarchar(30),
    @locatie nvarchar(30),
    @data date,
    @taxa int,
    @taxetotale int output)
as begin
begin try
    if exists( select count(*) from Evenimente as E
                where E.Nume like @nume and E.Locatie like @locatie
                and @data >=E.DataIncepere and @data<=E.DataFinalizare
                having count(*) <1)
    begin
        throw 50000,'Nu exista competitia',0;
    end
    insert into TaxeEvenimente(IDEveniment,Valoare,Data,Observatii)
    select E.IDEveniment, count(*)*@taxa, E.DataIncepere,'Taxe' from Evenimente as E
    inner join Premii as P
    on E.IDEveniment=P.IDEveniment
    where E.Nume like @nume and E.Locatie like @locatie
    and @data >=E.DataIncepere and @data<=E.DataFinalizare
    group by E.IDEveniment, E.DataIncepere;

    select top 1 @taxetotale=Valoare from TaxeEvenimente
    order by IDTaxa desc
end try
begin catch
    IF ERROR_NUMBER() = 547 -- Constraint violations
    BEGIN
        PRINT 'Constraint violation';
    END
    ELSE IF ERROR_NUMBER()=50000
```

```
BEGIN
    PRINT 'Nu exista competitia data cu numele '+@nume+' ultima competitie este ' ;
    select top 1 * from Evenimente as E
    inner join TipEveniment as T
    on T.IDTip=E.IDTip
    where T.Denumire like 'Competitie'
    order by IDEveniment desc
END

BEGIN
    PRINT 'Unhandled error';
END;

end catch
end
```

Testare bună:

```
declare @total int =0;
exec IntroducereTaxeEveniment
    @nume ='Cupa Junior',
    @locatie='Constanta',
    @data ='2017-02-28',
    @taxa =40,
    @taxetotale=@total output
select @total
```

Testare proasta:

```
declare @total int =0;
exec IntroducereTaxeEveniment
    @nume ='Cupa Junior2',
    @locatie='Constanta',
    @data ='2017-02-28',
    @taxa =40,
    @taxetotale=@total output
select @total
```

```
Nu exista competitia data cu numele Cupa Junior2 ultima competitie este
```

```
(1 row affected)
```

```
Unhandled error
```

```
(1 row affected)
```

```
Completion time: 2021-05-05T18:37:50.6681369+03:00
```

	IDEveniment	Nume	Nume Club Organizator	Locatie	DataIncepere	IDTip	DataFinalizare	IDTipSport	IDTip	Denumire
1	21	Cupa Dino	Victoria	Cluj	2017-02-04	1	2017-02-06	1	1	Competitie

	(No column name)
1	0

19. Să se creeze o procedură stocată pentru inserarea taxelor sportivilor, se va primi ca parametru numele și prenumele sportivului, data(implicit data curentă) și taxa (implicit 100lei)

```
if OBJECT_ID('IntroducereTaxaSportiv','p') is not null
    DROP proc IntroducereTaxaSportiv
go
create proc IntroducereTaxaSportiv(
    @nume nvarchar(30),
    @prenume nvarchar(30),
    @taxa int=100,
    @data date)
as begin
    if @data is null
        set @data=GETDATE()

    insert into TaxeSportivi(IDSportiv,Valoare,Data)
    select IDSportiv,@taxa,@data from Sportivi
    where Nume like @nume and Prenume like @prenume
end
```

Testare:

```
exec IntroducereTaxaSportiv
    @nume='Popescu',
    @prenume='Stefan',
    @taxa=100,
    @data='2017-01-15'
```

20. Să se creeze o procedură stocată care să returneze numărul de ani de când este înscris un anumit sportiv(pentru acesta se va trimite ca parametru numele și prenumele).

```
if OBJECT_ID('AniInscriere','p') is not null
    DROP proc AniInscriere
go
create proc AniInscriere(
    @nume nvarchar(30),
    @prenume nvarchar(30),
    @ani int =0 output)
as begin
    select @ani=YEAR(GETDATE())-YEAR( DataInscriere) from Sportivi
    where Nume like @nume and Prenume like @prenume
    return;
end
```

```
--testare
declare @ani int =0
exec AniInscriere
    @nume='Popescu',
    @prenume='Stefan',
    @ani=@ani output
select @ani
```

(No column name)	
1	11

21. Faceți o procedură stocată ce va calcula toți banii primiți de un anumit angajat.
Implicit va calcula valoarea banilor de la data angajării până la data curentă.

```
if OBJECT_ID('ValoareSalariiTotal','P') is not null
    drop proc ValoareSalariiTotal
go
create proc ValoareSalariiTotal(
    @idangajat as int,
    @startdata as date,
    @enddata as date,
    @valSalar as int=0 output)
as begin
if NULLIF(@startdata,'') is null and NULLIF(@enddata,'') is null
begin
    select @enddata=
    case when LaData is null then GETDATE()
    when LaData is not null then LaData
    end
    from Salarii where IDAngajat=@idangajat
    select @startdata=DATEADD(day,1,MIN(s.DeLaData))
        from Salarii as s where ValoareBruta>0 and IDAngajat=@idangajat;

end ;
with c as
(
    select @idangajat as idA, @startdata as data_inceput from Salarii as S
    where (S.DeLaData < @startdata and (S.LaData > @startdata or S.LaData is null))
and IDAngajat=@idangajat
union all
select idA,DATEADD(MONTH,1, data_inceput )
from c
where DATEADD(MONTH,1,data_inceput)<= @enddata
),a as
(
    select S.IDAngajat, (S.ValoareBruta*(1-(S.ValoareImpozit)/100)) as Salariu,
S.IDSalariu as ids,
        count (c.data_inceput) as nrluni from c
inner join Salarii as S
on c.idA=S.IDAngajat
```

```
        where ((c.data_inceput>S.DeLaData and c.data_inceput<=s.LaData) or  
(c.data_inceput>S.DeLaData and s.LaData is null))  
              and ValoareBruta>0  
        group by S.ValoareBruta, S.ValoareImpozit, S.IDAngajat, S.IDSalariu  
    ),b as  
    (  
        select a.IDAngajat,(a.Salariu*a.nrluni) as salariu from a  
        inner join Salarii as S  
        on S.IDSalariu=a.ids  
    )  
    select @valSalar=SUM(b.salariu) from b  
    return;  
end
```

TRIGGERE

22. Să se facă un trigger care să verifice ca la introducerea unui nou sportiv acesta să nu mai fie inserat.

```
if OBJECT_ID('InainteDeSportiv','tr') is not null
    drop trigger InainteDeSportiv
go
create trigger InainteDeSportiv
on Sportivi
after insert
as
begin
    if @@ROWCOUNT=0 return;
    set nocount on;
    if exists (select count(*) from inserted as i
               join Sportivi as A
               on A.CNP=i.CNP
               having COUNT(*)>1)
    begin
        throw 50000, 'acest sportiv exista deja',0;
    end
end
```

23. Să se facă un trigger care să verifice ca la introducerea unui nou angajat acesta să nu mai fie inserat.

```
if OBJECT_ID('InainteDeAngajat','tr') is not null
    drop trigger InainteDeAngajat
go
create trigger InainteDeAngajat
on Angajat
after insert
as
begin
    if @@ROWCOUNT=0 return;
    set nocount on;
    if exists (select count(*) from inserted as i
               join Angajat as A
               on A.CNP=i.CNP
               having COUNT(*)>1)
    begin
        throw 50000, 'acest angajat exista deja',0;
    end
end
```

24. Să se facă un trigger care în loc de ștergerea unui sport să adauge la numele acestuia „expirat” modifice data de plecare a tuturor sportivilor de la acel sport cu data curentă.

```
if OBJECT_ID('StergereSport','tr') is not null
    drop trigger StergereSport
go
create trigger StergereSport
on Sporturi
instead of delete
as
begin
    if @@ROWCOUNT=0 return;
    set nocount on;
    update Sporturi
    set Denumire=(select Denumire from deleted) + '_expirat'
    where IDSport=(select IDSport from deleted)

    update InregistrariSportiv
    set DataFinalizare=GETDATE()
    where IDSport=(select IDSport from deleted) and DataFinalizare is null
end
```

Pt ștergerea sportului MMA s-au adăugat la sportivii corespunzători data curentă 2021-04-30

	IDSport	Denumire		IDInregistrare	IDSport	IDSportiv	DataIncepere	DataFinalizare
1	1	Ju.Jitsu	1	5	4	4	2014-03-15	2021-04-30
2	2	BJJ	2	16	4	17	2014-01-01	2015-01-01
3	3	Karate	3	22	4	24	2015-04-23	2021-04-30
4	4	MMA_expirat	4	24	4	25	2015-07-30	2016-12-23
5	5	Toate	5	25	4	26	2015-06-30	2021-04-30
6	6	Box	6	29	4	31	2015-10-30	2021-04-30
7	7	Wushu_expirat	7	35	4	41	2016-05-15	2021-04-30

25. Să se facă un trigger care la ștergerea unui angajat să îi șteargă toate înregistrările aferente din Însotitori și Salarii.

```
if OBJECT_ID('stergere_angajat_dupaNume','TR') is not null
drop trigger stergere_angajat_dupaNume;
go
create trigger stergere_angajat_dupaNume
on Angajat
instead of delete
as
begin
    if @@ROWCOUNT=0 return;
    set nocount on;
    delete from Insoțitori
    where IDAngajat = (select IDAngajat from deleted)
    delete from Salarii
    where IDAngajat=(select IDAngajat from deleted)
    delete from InregistrariAngajati
```

```
        where IDAngajat=(select IDAngajat from deleted)
        delete from Angajat
        where IDAngajat=(select IDAngajat from deleted)
end
```

26. Să se facă un trigger care la ștergerea unui sportiv să șteargă toate înregistrările aferente din TaxeSportivi și Premii.

```
if OBJECT_ID('stergere_sportiv_dupaNume','TR') is not null
drop trigger stergere_sportiv_dupaNume;
go
create trigger stergere_sportiv_dupaNume
on Sportivi
instead of delete
as
begin
if @@ROWCOUNT=0 return;
set nocount on;
    delete from Premii
    where IDSportiv = (select IDSportiv from deleted)
    delete from TaxeSportivi
    where IDSportiv=(select IDSportiv from deleted)
    delete from InregistrariSportiv
    where IDSportiv=(select IDSportiv from deleted)
    delete from Sportivi
    where IDSportiv=(select IDSportiv from deleted)
end
```

27. Să se creeze un trigger pe SursaVenituri care la inserția unei surse noi să verifice sa nu mai existe. În caz că există afișează informațiile deja existente și un mesaj aferent, altfel se va afișa noua linie inserată.

```
if OBJECT_ID('inserare_sursa_venit','TR') is not null
drop trigger inserare_sursa_venit
go
create trigger inserare_sursa_venit
on SursaVenituri
after insert
as begin
    if @@ROWCOUNT=0 return;
    set nocount on;
    if exists (select count(*) from inserted as i
              join SursaVenituri as S
              on S.Nume=i.Nume
              having count(*)>1)
    begin
        select top 1 * from SursaVenituri where Nume=
        (select Nume from inserted);
        throw 50001, 'aceasta sursa exista deja', 0;
    end
else
```



```
begin
    select * from inserted
end
end
28. La inserarea unui premiu să se verifice dacă sportivul este înscris la tipul de sport pe
    care îl are competiția. În cazul în care nu este se va afișa un mesaj corespunzător.
if OBJECT_ID('verificareInscriereSport','TR') is not null
    drop trigger verificareInscriereSport
go
create trigger verificareInscriereSport
on Premii
after insert as
begin
    if @@ROWCOUNT=0 return;
    set nocount on;
    if exists (select count(*) from InregistrariSportiv as s
                join inserted as i
                on i.IDSportiv=s.IDSportiv
                join Evenimente as e
                on e.IDEveniment=i.IDEveniment
                where e.IDTipSport=s.IDSport
                having count(*)<1)
        begin
            if exists(select count(*) from Evenimente as e
                    inner join inserted as i
                    on e.IDEveniment=i.IDEveniment
                    where e.IDTipSport=5
                    having count(*) <1 )
                begin
                    throw 50002, 'Sportivul nu are cum sa participe la aceasta
competitie',0;
                end
            end
        end
end
29. La un update în Salarii, în cazul în care acesta se face pe coloana PânăLaData, să se
    facă comanda doar dacă noua dată adăugată este după cea din coloana DeLaData. În
    cazul în care se dorește să se modifice o altă coloană, utilizatorul va fi oprit.
if OBJECT_ID('modifica_salariu','TR') is not null
    drop trigger modifica_salariu
go
create trigger modifica_salariu
on Salarii
after insert,update as
begin
    if @@ROWCOUNT=0 return;
    set nocount on;
```

```
if exists ( select count(*) from Salarii as S
join inserted as i
on i.IDSalariu=S.IDSalariu
where i.DeLaData>i.LaData
having count(*) >=1)
begin
    throw 50003, 'Nu se poate adauga acest salariu, datele nu sunt corecte',0;
end
```

end

30. La adăugarea unui nou salariu, să se pună pe coloana PanaLaData ultimului salariu valoarea coloanei DeLaData de la noua înregistrare.

```
if OBJECT_ID('inserare_salariu','tr') is not null
    drop trigger inserare_salariu
go
create trigger inserare_salariu
on Salarii
after insert as
begin
    if @@ROWCOUNT=0 return;
    set nocount on;
    update Salarii
    set LaData=(select DeLaData from inserted )
    where IDAngajat=(select IDAngajat from inserted)
    and LaData is null;
```

end

31. Să se creeze un trigger care va modifica tabela de salarii, coloana PanaLaData cu data aferentă în cazul în care un angajat nu mai aparține clubului.

```
if OBJECT_ID('plecare_angajat','tr') is not null
    drop trigger plecare_angajat
go
create trigger plecare_angajat
on Angajat
after update as
begin
    if @@ROWCOUNT=0 return;
    set nocount on;
    update Salarii
    set LaData=(select DataFinalizareContract from inserted)
    where IDAngajat=(select IDAngajat from inserted)
    and LaData is null
```

end

Comenzi date în baza de date

SELECT

1. Să se afișeze numele și prenumele tuturor sportivilor care au trecut observații în tabelul Sportivi.

SELECT Nume, Prenume , Observatii **FROM** Sportivi

where Observatii is not NULL

	Nume	Prenume	Observatii
1	Arsene	Nicolae	Probleme Medicale
2	Mihaiu	Alexandru	Operat
3	Bondar	Lavinia	Probleme picior stang, Operata

2. Afișați id-ul, CNP-ul, Numele și prenumele tuturor sportivilor ce au făcut antrenamente la data de 2017-03-17 la sportul Ju-Jitsu sau la BJJ.

SELECT S.IDSportiv, S.Nume+' '+ S.Prenume **as** Nume, S.CNP **from** Sportivi **AS** S

INNER JOIN InregistrariSportiv **AS** I

ON S.IDSportiv =I.IDSportiv

INNER JOIN Sporturi **AS** Sp

ON SP.IDSport=S.IDSport

WHERE (Sp.Denumire LIKE 'JuJitsu' and

(I.DataFinalizare >=**CONVERT**(date,'2017-03-17') or I.DataFinalizare is

NULL))

or(Sp.Denumire like 'BJJ' and

(I.DataFinalizare >=**CONVERT**(date,'2017-03-17') or I.DataFinalizare is

NULL))

Group by S.Nume,S.Prenume, S.CNP,S.IDSportiv, I.DataFinalizare

	IDSportiv	Nume	CNP
1	3	Stoica Andrei	5001212145678
2	16	Negoiu Raluca	6010809124321
3	18	Manea Robert	5040315142323
4	19	Ionescu Roberta	6031321322132
5	22	Anolo Mihai	5031412322112
6	34	Mihalache Cristina	6000213141412
7	37	Novi Luna	6031014123213
8	1	Popescu Stefan	5011124145432

3. Afișați numele și prenumele, tipul angajaților care au participat la excursia din 2016-05-02 în ordine alfabetică

select A.Nume, A.Prenume, T.Denumire **from** Angajat **as** A

inner join TipAngajat **as** T

on T.IDTip=A.IdTip

inner join Insotitori **as** I

on I.IDAngajat=A.IDAngajat

inner join Evenimente **as** E

on E.IDEveniment=I.IDEveniment

inner join TipEveniment **as** TE

on TE.IDTip=E.IDTip

where TE.Denumire='Excursie' and E.DataIncepere like **CONVERT**(date,'2016-05-02')

order by A.Nume, A.Prenume

	Nume	Prenume	Denumire
1	Anton	Cristian	Presedinte
2	Copu	Marian	Antrenor
3	Grumi	Andrei	Sofer
4	Stoica	Ionut	Instructor

4. Afișați numele și numărul de sportivi de la fiecare sport care se antrenează în momentul actual, în ordine descrescătoare a numărului de persoane.

```
select S.Denumire, count(I.IDSportiv) as NrSportivi from Sporturi as S
inner join InregistrariSportiv as I
on S.IDSport=I.IDSport
where I.DataFinalizare is null
group by S.Denumire
order by NrSportivi desc
```

	Denumire	NrSportivi
1	Ju.Jitsu	7
2	MMA	5
3	Karate	3
4	Box	3
5	BJJ	2

5. Care este ultima competiție la care a participat sportivul Popescu Stefan.

```
select top 1 Sportivi.IDSportiv,Evenimente.IDEveniment, Evenimente.Nume,
Evenimente.Locatie from Evenimente
inner join TipEveniment
on TipEveniment.IDTip=Evenimente.IDTip
inner join Premii
on Premii.IDEveniment =Evenimente.IDEveniment
inner join Sportivi
on Sportivi.IDSportiv = Premii.IDSportiv
where TipEveniment.Denumire like 'Competitie' and Sportivi.Nume like 'Popescu' and
Sportivi.Prenume like 'Stefan'
group by Sportivi.IDSportiv, Evenimente.IDEveniment, Evenimente.Nume,
Evenimente.Locatie
order by Evenimente.IDEveniment desc
```

	IDSportiv	IDEveniment	Nume	Locatie
1	1	18	Campionatul National	Cluj

6. Câte locuri s-au câștigat la competiția Cupa Aquila din anul 2016.

```
select count(*) from Premii as P
inner join Evenimente as E
on p.IDEveniment=e.IDEveniment
inner join TipEveniment as t
on t.IDTip=e.IDTip
where t.Denumire like 'Competitie' and
e.Nume like 'Cupa Aquila' and
YEAR(e.DataIncepere) like '2016'
```

	(No column name)
1	6

7. Afișați pentru fiecare sport valid toți sportivi înscriși până acum, numărul de sportivi total și numărul de sportivi actuali.

```
select S.Denumire, count(*) as NrTotal ,  
       count(*) - count(I.DataFinalizare) as NrActual  
from InregistrariSportiv as I  
inner join Sporturi as S  
on S.IDSport=I.IDSport  
where S.Denumire not like '%expirat%'  
group by S.Denumire
```

	Denumire	NrTotal	NrActual
1	BJJ	3	2
2	Box	6	3
3	JuJitsu	11	7
4	Karate	7	3

8. Care sunt angajații care se ocupă de sportul cu cei mai mulți sportivi participanți activ.

```
with c as  
(  
select I.IDSport, count(*) - count(I.DataFinalizare) as NrActual  
from InregistrariSportiv as I  
group by I.IDSport  
) , b as  
(  
select max(c.NrActual) as maxim from c  
)  
select a.Nume+' '+a.Prenume from InregistrariAngajati as i  
inner join c  
on c.IDSport = i.IDSport  
inner join b on  
c.NrActual=b.maxim  
inner join Angajat as a  
on a.IDAngajat=i.IDAngajat
```

	(No column name)
1	Anton Cristian
2	Copu Marian
3	Princa Cristina
4	Grumi Andrei

9. Care sunt instructorii participanți la Gratarul din Valcele din anul 2016. Să se afișeze numele și prenumele acestora.

```
select A.Nume, A.Prenume from Angajat as A  
inner join TipAngajat as T  
on A.IdTip=T.IDTip  
inner join Insotitori as I  
on i.IDAngajat=A.IDAngajat  
inner join Evenimente as E
```

```
on E.IDEveniment=I.IDEveniment
where E.Nume = 'Gratar' and
      E.Locatie like 'Valcele' and
      Year(E.DataIncepere) like '2016' and
      T.Denumire like 'Instructor'
```

	Nume	Prenume
1	Stoica	Ionut

10. Afișați numele, prenumele, sexul, numărul de sporturi la care predă și numărul de evenimente la care a participat cel mai activ angajat din club.

```
with c as(
select I.IDAngajat, count(*) as NrEvenimente from Insotitori as I
group by I.IDAngajat
), b as
(
select I.IDAngajat, count(*) as NrSporturi from InregistrariAngajati as I
group by I.IDAngajat
)
select top 1 A.Nume, A.Prenume,
case when A.CNP like '2%' then 'Feminin'
      when A.CNP like '1%' then 'Masculin'
      when A.CNP like '5%' then 'Masculin'
      when A.CNP like '6%' then 'Feminin'
      else 'Nedefinit'
end as Gen,
b.NrSporturi,c.NrEvenimente
from c
inner join Angajat as A
on c.IDAngajat=A.IDAngajat
inner join b
on c.IDAngajat=b.IDAngajat
order by NrEvenimente desc
```

	Nume	Prenume	Gen	NrSporturi	NrEvenimente
1	Copu	Marian	Masculin	2	13

11. Pentru cel mai activ angajat afișați toate cantonamentele la care a paricipat.

```
with c as
(
select I.IDAngajat, count(*) as NrEvenimente from Insotitori as I
group by I.IDAngajat
),b as
(
select max(c.NrEvenimente) as maxim from c
)
select E.Nume, E.Locatie, E.DataIncepere, E.DataFinalizare from c
inner join b
on c.NrEvenimente=b.maxim
```

SELECT

```

inner join Insotitori as I
on i.IDAngajat=c.IDAngajat
inner join Evenimente as E
on E.IDEveniment = I.IDEveniment
inner join TipEveniment as T
on T.IDTip=e.IDTip
inner join Angajat as A
on A.IDAngajat=c.IDAngajat
where T.Denumire like 'Cantonament'
group by E.Nume, E.Locatie, E.DataIncepere, E.DataFinalizare

```

	Nume	Locatie	DataIncepere	DataFinalizare
1	Cantonament Bran	Bran	2012-11-05	2012-10-12
2	Cantonament Comandau	Comandau	2012-10-03	2012-10-10

12. Numele sportivilor și numărul de taxe plătite în anul 2014 de către aceștia.

```

select S.Nume, S.Prenume, count(*) from TaxeSportivi as TS
inner join Sportivi as S
on S.IDSportiv=TS.IDSportiv
where year(TS.Data) =2014
group by S.Nume, S.Prenume

```

	Nume	Prenume	(No column name)
1	Mihailu	Alexandru	12
2	Blaga	Ana-Maria	12
3	Stoica	Andreea	12
4	Stoica	Andrei	12
5	Negoiu	Cristian	12
6	Ghenea	Diana	12
7	Arsene	Nicolae	12
8	Margarit	Norbert	8
9	Negoiu	Raluca	12
10	Manea	Robert	10
11	Ionescu	Robert	4
12	Popescu	Stefan	13
13	Arsene	Vlad	12

13. Cheltuielile totale din anul 2017 făcute pe comenzi.

```

select sum(F.Valoarea) as ValoareComenzi from Facturi as F
inner join TipFactura as T
on T.IDTip=F.IDTip
where T.Denumire like 'Comanda' and
year(F.Data)= 2017

```

	ValoareComenzi
1	1200

14. Suma totală de bani primiți de la Bertis ca și Sponsorizare.

```
select V.Valoare, V.Data from Venituri as V
inner join SursaVenituri as S
on S.IDSursa=V.IDSursa
inner join TipVenituri as T
on V.IDTipVenit=T.IDTip
where S.Nume like 'Bertis' and T.Denumire like 'Sponsorizare'
```

	Valoare	Data
1	1000	2016-04-18

15. Care sunt datele sportivelor de la box ce participă activ la antrenamente.

```
select S.Nume, S.Prenume, S.DataInscriere, S.Observatii
from Sportivi as S
inner join InregistrariSportiv as I
on I.IDSportiv=S.IDSportiv
inner join Sporturi as Sp
on Sp.IDSport=I.IDSport
where (S.CNP like '6%' or S.CNP like '2%')
and Sp.Denumire like 'Box'
```

	Nume	Prenume	DataInscriere	Observatii
1	Ghenea	Diana	2010-10-01	NULL
2	Negoiu	Raluca	2013-09-09	NULL

16. Care sunt sportivii care au participat la cel puțin 2 competiții și anii la care au participat.

```
select P.IDSportiv, S.Nume+' '+S.Prenume as Nume,
       year(E.DataFinalizare) as An,
       count (*) as NrEvenimente from Premii as P
inner join Evenimente as E
on E.IDEveniment=P.IDEveniment
inner join Sportivi as S
on S.IDSportiv=P.IDSportiv
group by P.IDSportiv, S.Nume, S.Prenume, E.DataFinalizare
having count (*) >=2
order by An, P.IDSportiv
```


	IDSportiv	Nume	An	NrEvenimente
1	1	Popescu Stefan	2014	2
2	1	Popescu Stefan	2017	2
3	3	Stoica Andrei	2017	2
4	16	Negoiu Raluca	2017	2
5	18	Manea Robert	2017	2
6	18	Manea Robert	2017	2
7	19	Ionescu Roberta	2017	2
8	22	Anolo Mihai	2017	2
9	34	Mihalache Cristina	2017	2
10	37	Novi Luna	2017	2

17. Afișați toți sportivii și numărul de competiții la care au participat în funcție de sport.

```
select s.Nume+' '+S.Prenume ,e.IDTipSport,
       YEAR(e.DataIncepere) ,count (*) from Premii as p
inner join Evenimente as e
on p.IDEveniment=e.IDEveniment
inner join Sportivi as s
on s.IDSportiv=p.IDSportiv
group by e.IDTipSport, S.Nume+' '+S.Prenume ,
       YEAR(e.DataIncepere)
order by YEAR(e.DataIncepere), e.IDTipSport
```

21	Blaga Ana-Maria	1	2013	1
22	Popescu Stefan	1	2013	2
23	Stoica Andreea	1	2013	2
24	Stoica Andrei	1	2013	2
25	Ghenea Diana	6	2013	1
26	Blaga Ana-Maria	1	2014	2
27	Negoiu Raluca	1	2014	2
28	Popescu Stefan	1	2014	3
29	Stoica Andreea	1	2014	2
30	Stoica Andrei	1	2014	2
31	Anolo Mihai	1	2014	1

18. Afișați numele sportului, anul de incepere, numele si prenumele sportivilor care participă sau au participat la un sport de mai mult de 5 ani. Specificati cine mai e activ si cine nu.

```
select i.IDSportiv,i.IDSport,
       year(GETDATE())-year(i.DataIncepere) as AniActivitate,
       'activi' as Activitate
from InregistrariSportiv as i
where i.DataFinalizare is null
union
select i.IDSportiv,i.IDSport,
       year(i.DataFinalizare)-year(i.DataIncepere) as AniActivitate,
       'plecati' as Activitate
from InregistrariSportiv as i
where i.DataFinalizare is not null
order by IDSportiv
```

	IDSportiv	IDSport	AniActivitate	Activitate
1	1	1	10	plecati
2	2	1	5	plecati
3	3	1	11	activi
4	4	1	5	plecati
5	4	4	7	plecati
6	5	3	4	plecati
7	6	3	11	activi
8	6	6	11	activi
9	7	3	5	plecati
10	11	6	2	plecati
11	12	1	2	plecati
12	12	2	6	plecati

19. Numărul de fete și de baieti care au participat la evenimentul x.

select

count (case when SUBSTRING(S.CNP,1,1)=1 or SUBSTRING(S.CNP,1,1)=5 then 1 end) as

Baieti,

count (case when SUBSTRING(S.CNP,1,1)=2 or SUBSTRING(S.CNP,1,1)=6 then 1 end) as

Fete

from Evenimente as E

inner join Premii as p

on p.IDEveniment=e.IDEveniment

inner join TipEveniment as t

on t.IDTip=e.IDTip

inner join Sportivi as s

on s.IDSportiv=p.IDSportiv

where t.Denumire like 'Cantonament'

and e.Nume like 'Cantonament Comandau'

and YEAR(e.DataIncepere) =2012

	Baieti	Fete
1	6	3

20. Să se afișeze toți sportivii care pot participa la competiția Cupa Dino din data de 2017-02-04, ordonați alfabetic.

select sp.Nume+' '+sp.Prenume **from** InregistrariSportiv as i

inner join Sporturi as s

on i.IDSport=s.IDSport

inner join Evenimente as e

on i.IDSport=e.IDTipSport

inner join Sportivi as sp

on sp.IDSportiv=i.IDSportiv

where e.Nume like 'Cupa Dino' and e.DataIncepere like '2017-02-04'

and sp.DataPlecare is null

order by sp.Nume

Comenzi date în baza de date SELECT

	(No column name)
1	Anolo Mihai
2	Ionescu Roberta
3	Manea Robert
4	Mihalache Cristina
5	Negoiu Raluca
6	Novi Luna
7	Stoica Andrei
8	Stoica Andreea

21. Să se calculeze toți banii ce au ieșit din club în anul x.

```

declare @total int, @counter int, @maxid int
select @total=0, @counter=min(IDAngajat), @maxid=max(IDAngajat) from Angajat
while(@counter is not null and @counter<=@maxid)
begin
    declare @suma int =0;
    exec ValoareSalariiTotal
        @idangajat=@counter,
        @valsalar=@suma output;
    set @total=@total+@suma
    set @counter=@counter+1;
end;
select @total
with c(valoare) as
(
    select Valoarea from Facturi
    union all
    select Valoare from TaxeEvenimente
)
select @suma=SUM(valoare) from c
set @total=@total+@suma
select @total

```

	(No column name)
1	714090

22. Să se calculeze toți banii intrați în club.

```

with c (Valoare) as
(
    select Valoare from TaxeSportivi
    union all
    select Valoare from Venituri
)
select SUM(Valoare) from c

```

	(No column name)
1	53150

SELECT

23. Să se afișeze profitul din anul 2016.

```

declare @total int, @counter int, @maxid int
select @total=0, @counter=min(IDAngajat), @maxid=max(IDAngajat) from Angajat
while(@counter is not null and @counter<=@maxid)
begin
    declare @suma int =0;
    exec ValoareSalariiTotal
        @idangajat=@counter,
        @startdata='2016-01-01',
        @enddata='2016-12-31',
        @valsalar=@suma output;
    if(@suma is null)
        set @suma=0;
    set @total=@total+@suma
    set @counter=@counter+1;
end;
with c(valoare) as
(
    select Valoarea from Facturi
    where Data>'2016-01-01' and Data<'2016-12-31'
    union all
    select Valoare from TaxeEvenimente
    where Data>'2016-01-01' and Data<'2016-12-31'
)
select @suma=SUM(valoare) from c
set @total=@total+@suma
set @suma=0;
with c (Valoare) as
(
    select Valoare from TaxeSportivi
    union all
    select Valoare from Venituri
)
select @suma=SUM(Valoare) from c
set @total=(@total-@suma)*(-1)
select @total

```

(No column name)	
1	-13520

24. Să se afișeze numele, locația, tipul, suma taxelor tuturor evenimentelor la care s-a plătit mai mult de 300lei.

```

with c as
(select E.Nume,E.[Nume Club Organizator], E.Locatie, S.Denumire,
    SUM(T.Valoare) as Costuri from Evenimente as E
inner join TaxeEvenimente as T
on E.IDEveniment=T.IDEveniment
inner join Sporturi as S
on S.IDSport=E.IDTipSport

```

```
group by E.Nume, E.[Nume Club Organizator], E.Locatie, S.Denumire
)
select * from c
where Costuri >300
```

	Nume	Nume Club Organizator	Locatie	Denumire	Costuri
1	Campionatul National	FRAM	Bucuresti	Ju.Jitsu	1500
2	Campionatul National	FRAM	Cluj	Ju.Jitsu	550
3	Cantonament Bran	CS	Bran	Karate	800
4	Cantonament Comandau	CS	Comandau	Toate	1000
5	Cupa Aquila	Aquila	Giurgiu	Ju.Jitsu	950
6	Cupa Junior	Primaria Constan?a	Constanta	B.JJ	500
7	Cupa Valcea	CSM Valcea	Valcea	MMA_expirat	500
8	Stagiu De Vara FRAM	FRAM	Constanta	Ju.Jitsu	1500

25. Afişaţi numele şi prenumele, data de înscriere în club ,numărului de sporturi la care a fost înscris, numărul de sporturi la care nu mai participă în prezent şi data ultimului sport la care a renunţat, ordonaţi după data de înscriere în club al tuturor sportivilor.

```
select S.Nume+' '+S.Prenume,S.DataInscriere, count(*) as NrSporturiTotale,
        count (I.DataFinalizare) as NrSporturiNeparticipante,
        MAX(I.DataFinalizare) as UltimaData
from InregistrariSportiv as I
inner join Sportivi as S
on S.IDSportiv=I.IDSportiv
group by S.Nume, S.Prenume, S.DataInscriere
order by S.DataInscriere
```

	(No column name)	DataInscriere	NrSporturiTotale	NrSporturiNeparticipante	UltimaData
1	Blaga Ana-Maria	2010-09-11	1	1	2015-03-13
2	Popescu Stefan	2010-09-11	1	1	2020-06-15
3	Stoica Andreea	2010-09-13	2	1	2015-10-13
4	Stoica Andrei	2010-09-13	1	0	NULL
5	Margarit Norbert	2010-09-15	1	1	2014-08-14
6	Arsene Nicolae	2010-09-23	1	1	2015-10-01
7	Arsene Vlad	2010-09-23	2	0	NULL
8	Ghenea Diana	2010-10-01	1	1	2012-10-01
9	Negoiu Cristian	2010-10-02	3	3	2016-10-01
10	Negoiu Raluca	2013-09-09	2	0	NULL
11	Mihaiu Alexandru	2014-01-01	1	1	2015-01-01
12	Manea Robert	2014-03-23	2	0	NULL
13	Ionescu Roberta	2014-06-04	1	0	NULL
14	Anolo Mihai	2014-10-13	1	0	NULL
15	Jogan Calin	2015-02-15	1	1	2016-10-24
16	Coroie Narcisa	2015-04-23	1	0	NULL
17	Balea Teodor	2015-05-15	2	2	2016-12-23
18	Manole Alexandru	2015-06-30	1	0	NULL
19	Amz Catalin	2015-09-23	1	0	NULL
20	Lipan Oana	2015-10-30	1	0	NULL
21	Mihalache Cristina	2016-01-31	1	0	NULL
22	Novi Luna	2016-02-15	2	0	NULL

26. Care sunt numele, prenumele și ziua atât a sportivilor cât și a angajaților ce sunt născuți în luna mai ordonați după zi

```
with persoane(Nume, Prenume, CNP) as
(
    select Nume,Prenume, CNP from Angajat
    union
    select Nume, Prenume,CNP from Sportivi
)
select Nume,Prenume, SUBSTRING(CNP,6,2) as zi from persoane
where SUBSTRING(CNP,4,2)='05'
order by zi
```

	Nume	Prenume	zi
1	Branza	Ionut	09
2	Margarit	Norbert	13
3	Negoiu	Cristian	17
4	Coroie	Narcisa	30

27. Care sunt evenimentele organizate de CS, ordonate după numărul de sportivi participanți.

```
select E.Nume,count(*) as nr from Evenimente as E
inner join Premii as P
on P.IDEveniment=E.IDEveniment
where E.[Nume Club Organizator] like 'CS'
group by E.Nume
order by nr asc
```

	Nume	nr
1	Cantonament Bran	4
2	Gratar	6
3	Biciclete Sugas Bai	7
4	Cantonament Comandau	9

28. Câți bani a plătit fiecare sportiv, afișați suma, numele, prenumele și data de înscriere ca și lună și an.

```
select S.Nume, S.Prenume, MONTH (S.DataInscriere) as luna, YEAR(S.DataInscriere) as an,
SUM(T.Valoare) as ValoareTotala from Sportivi as S
inner join TaxeSportivi as T
on T.IDSportiv=S.IDSportiv
group by S.Nume, S.Prenume, S.DataInscriere
order by an, luna
```

Nume	Prenume	luna	an	ValoareTotala	Jogan	Calin	2	2015	1050
Stoica	Andreea	9	2010	3600	Coroie	Narcisa	4	2015	1050
Stoica	Andrei	9	2010	3600	Balea	Teodor	5	2015	1000
Negoiu	Cristian	10	2010	3600	Manole	Alexandru	6	2015	950
Ghenea	Diana	10	2010	3450	Amz	Catalin	9	2015	750
Negoiu	Raluca	9	2013	1950	Lipan	Oana	10	2015	700
Mihaiu	Alexandru	1	2014	600	Mihala	Cristina	1	2016	950
Manea	Robert	3	2014	2050	Branza	Ionut	4	2016	400
Ionescu	Roberta	6	2014	1450	Bondar	Cristian	5	2016	300
Anolo	Mihai	10	2014	1100	Bondar	Lavinia	5	2016	300

29. Afișați plățile de la toate competițiile la care a participat sportivul Popescu Stefan.

```
select SUM(T.Valoare) from Premii as P
inner join Sportivi as S
on S.IDSportiv=P.IDSportiv
inner join Evenimente as E
on P.IDEveniment=E.IDEveniment
inner join TaxeEvenimente as T
on T.IDEveniment=E.IDEveniment
inner join TipEveniment as TE
on TE.IDTip=E.IDTip
where S.Nume like 'Popescu' and S.Prenume like 'Stefan' and TE.Denumire like 'Competitie'
```

	(No column name)
1	4550

30. Afișați numărul de competiții la de la fiecare tip de sport și numărul total de participanți.(daca x a participat la 3 competitii at sunt 3 participanti).

```
with c as
(
    select P.IDEveniment,E.IDTipSport, S.IDSportiv, count(*) as NrProbe from Premii as P
    inner join Sportivi as S
    on S.IDSportiv=P.IDSportiv
    inner join Evenimente as E
    on P.IDEveniment=E.IDEveniment
    group by P.IDEveniment, S.IDSportiv, E.IDTipSport
),a as
(
    select IDTipSport,IDEveniment ,count(*) as NrParticipanti from c
    group by IDEveniment, IDTipSport
)
select s.Denumire,SUM(a.NrParticipanti) from a
inner join Sporturi as S
on a.IDTipSport=S.IDSport
group by s.Denumire
```

	Denumire	(No column name)
1	BJJ	3
2	Box	4
3	JuJitsu	40
4	Karate	7
5	MMA_expirat	8
6	Toate	22

31. Copii care au participat la ultima excursie.

```
with c as
(select top 1 E.IDEveniment from Evenimente as E
inner join TipEveniment as T
on T.IDTip=E.IDTip
where T.Denumire like 'Excursie')
```

SELECT

```

order by E.DataIncepere desc
)
select S.Nume, S.Prenume from c
inner join Premii as P
on c.IDEveniment = p.IDEveniment
inner join Sportivi as S
on S.IDSportiv = P.IDSportiv

```

32. Afișați pentru fiecare competiție datele acesteia și media per sportiv a taxelor.

```

select E.Nume, E.Locatie, E.DataIncepere,
       SUM(T.Valoare)/count(*) as PretPerSportiv from Evenimente as E
inner join TaxeEvenimente as T
on E.IDEveniment = T.IDEveniment
inner join Premii as P
on P.IDEveniment = E.IDEveniment
group by E.Nume, E.Locatie, E.DataIncepere

```

	Nume	Locatie	DataIncepere	PretPerSpo
1	Biciclete Sugas Bai	Sugas Bai	2014-05-23	50
2	Campionatul National de Box	Bucuresti	2016-02-18	200
3	Campionatul National	Bucuresti	2012-04-15	500
4	Campionatul National	Bucuresti	2014-05-30	500
5	Campionatul National	Bucuresti	2016-06-10	300
6	Campionatul National	Cluj	2017-03-14	275
7	Campionatul National	Sibiu	2014-07-12	300
8	Cantonament Bran	Bran	2012-11-05	800
9	Cantonament Comandau	Comandau	2012-10-03	1000
10	Cupa Aquila	Giurgiu	2013-12-05	250
11	Cupa Aquila	Giurgiu	2016-12-05	225
12	Cupa Brasov	Brasov	2011-09-12	200
13	Cupa Junior	Constanta	2017-02-28	165
14	Cupa Kokoro	Brasov	2014-09-12	150
15	Cupa Romana	Cluj-Napoca	2013-11-12	150
16	Cupa Valcea	Valcea	2016-12-18	250
17	Gratar	Valcele	2016-05-02	0
18	Stagiu De Vara FRAM	Constanta	2013-09-10	1500

33. Care este suma totală de bani dați pe cantonamente.

```

select sum(T.Valoare) from Evenimente as E
inner join TaxeEvenimente as T
on E.IDEveniment = T.IDEveniment
inner join TipEveniment as Te
on TE.IDTip = E.IDTip
where Te.Denumire like 'Cantonament'

```

	(No column name)
1	1800

34. Care este suma totală de bani primită din donații.

```
select SUM(V.Valoare) from Venituri as V
inner join TipVenituri as T
on V.IDTipVenit=T.IDTip
where T.Denumire like 'Donatii'
```

	(No column name)
1	2400

35. Care este vârsta ce o îndeplinește fiecare sportiv în anul curent.

```
select Nume+' '+Prenume,
       case
       when SUBSTRING(CNP,2,2) > FORMAT(getdate(), 'yy') then CAST(FORMAT(getdate(), 'yy')
       as int)-CAST(SUBSTRING(CNP,2,2) as int) +100
       when SUBSTRING(CNP,2,2) < FORMAT(getdate(), 'yy') then CAST(FORMAT(getdate(), 'yy')
       as int)-CAST(SUBSTRING(CNP,2,2) as int)
       else '00'
       end as Ani
from Sportivi
```

	(No column name)	Ani
1	Popescu Stefan	20
2	Blaga Ana-Maria	20
3	Stoica Andrei	21
4	Stoica Andreea	21
5	Margarit Norbert	19
6	Arsene Vlad	17
7	Arsene Nicolae	21
8	Ghenea Diana	21
9	Negoiu Cristian	18
10	Negoiu Raluca	20
11	Mihaiu Alexandru	19
12	Manea Robert	17
13	Ionescu Roberta	18
14	Anolo Mihai	18
15	Jogan Calin	18
16	Coroie Narcisa	19

36. Afișați numele, prenumele și data nașterii(xx-Month) a fiecărui sportiv ce participă la sportul JuJitsu.

```
select s.Nume+' '+s.Prenume as Nume,
       SUBSTRING(s.CNP,6,2) as zi,
       datename ( month, dateadd (month ,CAST(SUBSTRING(s.CNP,4,2) as int)-1,0)) as luna
from Sportivi as S
inner join InregistrariSportiv as I
on I.IDSportiv=S.IDSportiv
inner join Sporturi as Sp
on I.IDSport=sp.IDSport
where sp.Denumire like 'JuJitsu'
```

Comenzi date în baza de date SELECT

	Nume	zi	luna
1	Popescu Stefan	24	November
2	Blaga Ana-Maria	14	September
3	Stoica Andrei	12	December
4	Stoica Andreea	12	December
5	Negoiu Cristian	17	May
6	Negoiu Raluca	09	August
7	Manea Robert	15	March
8	Ionescu Roberta	21	January
9	Anolo Mihai	12	February
10	Mihalache Cristina	13	February
11	Novi Luna	14	October

37. Aflați ultima categorie la care a participat fiecare sportiv. În cazul în care la ultima competiție a participat la două categorii se vor afișa amandouă.

```
with c as(
select S.IDSportiv,max(E.DataIncepere) as datamax from Premii as P
inner join Sportivi as S
on P.IDSportiv=S.IDSportiv
inner join Evenimente as E
on E.IDEveniment=P.IDEveniment
inner join TipEveniment as T
on T.IDTip=E.IDTip
where T.Denumire like 'Competitie'
group by S.IDSportiv
)
select S.Nume +' '+S.Prenume,P.Categorie from c
inner join Evenimente as e
on e.DataIncepere=c.datamax
inner join TipEveniment as T
on T.IDTip=E.IDTip
inner join Premii as P
on P.IDEveniment=E.IDEveniment
inner join Sportivi as S
on S.IDSportiv=c.IDSportiv
where T.Denumire like 'Competitie' and P.IDSportiv=c.IDSportiv
order by c.IDSportiv
```

	(No column name)	Categorie			
1	Popescu Stefan	U18 -65kg	12	Negoiu Raluca	U18 -50gk
2	Popescu Stefan	U18 -70kg	13	Negoiu Raluca	U18 -55kg
3	Blaga Ana-Maria	U18 -60kg	14	Manea Robert	U16 -45kg
4	Stoica Andrei	U18 -60kg	15	Manea Robert	U16 -50kg
5	Stoica Andrei	U18 -65kg	16	Ionescu Roberta	U16 -48kg
6	Stoica Andreea	U18 -57kg	17	Ionescu Roberta	U16 -52kg
7	Margarit Norbert	U18 -55kg	18	Anolo Mihai	U16 -50kg
8	Arsene Vlad	U18 -60kg	19	Anolo Mihai	U16 -55kg
9	Arsene Nicolae	U18 -55kg	20	Coroie Narcisa	U16 -60kg
10	Ghenea Diana	U15 -50kg	21	Manole Alexandru	U18 -70kg
11	Negoiu Cristian	U18 -55kg	22	Amz Catalin	U16 -60kg

38. Pentru fiecare sportiv calculați o proporție a evenimentelor la care a participat în funcție de sport.

```
with ps as(
select P.IDSportiv, S.Denumire, count(*) as NrEvSportiv from Premii as P
inner join Evenimente as E
on P.IDEveniment=E.IDEveniment
inner join Sporturi as S
on E.IDTipSport=S.IDSport
group by P.IDSportiv, S.Denumire
), pe as(
select S.Denumire, count(*) as NrEvTotal from Evenimente as E
inner join Sporturi as S
on S.IDSport=E.IDTipSport
group by S.Denumire
)
select S.Nume+ ' '+S.Prenume, pe.Denumire, cast(ps.NrEvSportiv as float)/cast(pe.NrEvTotal as float) from pe
inner join ps
on pe.Denumire=ps.Denumire
inner join Sportivi as S
on S.IDSportiv=ps.IDSportiv
order by ps.IDSportiv, pe.Denumire
```

	(No column name)	Denumire	(No column name)
1	Popescu Stefan	Ju.Jitsu	1
2	Popescu Stefan	Karate	0.5
3	Popescu Stefan	Toate	1
4	Blaga Ana-Maria	Ju.Jitsu	0.5
5	Blaga Ana-Maria	Toate	0.666666666666667
6	Stoica Andrei	Ju.Jitsu	1
7	Stoica Andrei	Toate	1
8	Stoica Andreea	Ju.Jitsu	0.8
9	Stoica Andreea	MMA_expirat	1
10	Stoica Andreea	Toate	1
11	Margarit Norbert	Karate	1
12	Margarit Norbert	Toate	0.333333333333333
13	Arsene Vlad	BJJ	1
14	Arsene Vlad	Box	0.5
15	Arsene Vlad	Karate	1
16	Arsene Vlad	Toate	0.666666666666667
17	Arsene Nicolae	Karate	1
18	Arsene Nicolae	Toate	0.666666666666667
19	Ghenea Diana	Box	0.5
20	Ghenea Diana	Toate	0.333333333333333
21	Negoiu Cristian	Box	0.5
22	Negoiu Cristian	Ju.Jitsu	0.1
23	Negoiu Cristian	Toate	1

39. Afișați numele, data și numărul de zile cât au durat fiecare competiție din București.

```
select Nume,DataIncepere,  
       DATEDIFF(day,DataIncepere,DataFinalizare) as NrZile from Evenimente as E  
inner join TipEveniment as T  
on E.IDTip=T.IDTip  
where Locatie like 'Bucuresti' and T.Denumire like 'Competitie'
```

	Nume	DataIncepere	NrZile
1	Campionatul National	2012-04-15	2
2	Campionatul National	2014-05-30	3
3	Campionatul National de Box	2016-02-18	3
4	Campionatul National	2016-06-10	3

40. Afișați pentru fiecare sportiv numele, prenumele și numărul de localități în care a fost, nu contează tipul evenimentului sau al sportului.

```
with c as  
(  
    select P.IDSportiv, E.Locatie from Premii as P  
    inner join Evenimente as E  
    on P.IDEveniment = E.IDEveniment  
    group by p.IDSportiv, E.Locatie  
)  
select S.Nume, S.Prenume, count(*) from c  
inner join Sportivi as S  
on C.IDSportiv=S.IDSportiv  
group by S.Nume, S.Prenume
```

	Nume	Prenume	(No column name)
1	Manole	Alexandru	1
2	Blaga	Ana-Maria	5
3	Stoica	Andreea	8
4	Stoica	Andrei	8
5	Amz	Catalin	1
6	Negoiu	Cristian	4
7	Mihalache	Cristina	2
8	Ghenea	Diana	2
9	Branza	Ionut	1
10	Novi	Luna	2
11	Anolo	Mihai	1
12	Coroie	Narcisa	1
13	Arsene	Nicolae	4
14	Margarit	Norbert	3
15	Lipan	Oana	1
16	Negoiu	Raluca	7
17	Manea	Robert	2
18	Ionescu	Roberta	1
19	Popescu	Stefan	8
20	Arsene	Vlad	6

41. Care este competiția cu cei mai mulți participanți(se va lua în considerare ca și număr de participanți numărul de probe) și cate locuri pe podium au fost la aceasta.

```
with c as  
(  
    select P.IDEveniment, count(*) as nrProbe from Premii as P  
    inner join Evenimente as E  
    on P.IDEveniment=E.IDEveniment  
    inner join TipEveniment as T  
    on T.IDTip=E.IDTip  
    where T.Denumire like 'Competitie'
```

SELECT

```

        group by P.IDEveniment
    ),a as
    (
        select max(nrProbe) as maximnumber from c
    )
    select E.Nume, E.Locatie,E.DataIncepere, E.DataFinalizare ,count(
        case when P.Premiu like 'locul%' then 1
        else 0
        end) as podium
    from a
    inner join c
    on c.nrProbe =a.maximnumber
    inner join Premii as P
    on P.IDEveniment=c.IDEveniment
    inner join Evenimente as E
    on E.IDEveniment=c.IDEveniment
    group by E.Nume, E.Locatie,E.DataIncepere, E.DataFinalizare

```

	Nume	Locatie	DataIncepere	DataFinalizare	podium
1	Campionatul National	Cluj	2017-03-14	2017-03-16	16

42. Care este valoarea impozitului dat la stat pentru angajatul Anton Cristian de cand s-a angajat.

```

declare @idangajat int;
select @idangajat=IDAngajat from Angajat where Nume like 'Anton' and Prenume like 'Cristian'
declare @startdata date;
declare @enddata date;
select @enddata=
    case when LaData is null then GETDATE()
    when LaData is not null then LaData
    end
    from Salarii where IDAngajat=@idangajat
select @startdata=DATEADD (day,1,MIN(s.DeLaData))
    from Salarii as s where ValoareBruta>0 and IDAngajat=@idangajat;

with c as
(
    select @idangajat as idA, @startdata as data_inceput from Salarii as S
    where (S.DeLaData <@startdata and( S.LaData>@startdata or S.LaData is null))
    and IDAngajat=@idangajat
    union all
    select idA,DATEADD(MONTH,1, data_inceput )
    from c
    where DATEADD(MONTH,1,data_inceput)<= @enddata
),a as
(

```

```

select S.IDAngajat, (S.ValoareBruta*(S.ValoareImpozit)/100) as Impozit,
S.IDSalariu as ids,
count (c.data_inceput) as nrluni from c
inner join Salarii as S
on c.idA=S.IDAngajat
where ((c.data_inceput>S.DeLaData and c.data_inceput<=S.LaData) or
(c.data_inceput>S.DeLaData and S.LaData is null))
and ValoareBruta>0
group by S.ValoareBruta, S.ValoareImpozit, S.IDAngajat, S.IDSalariu
),b as
(
select a.IDAngajat,(a.Impozit*a.nrluni) as impozit from a
inner join Salarii as S
on S.IDSalariu=a.ids
)
select SUM(impozit) from b

```

(No column name)	
1	38800

43. Afișați pentru angajatul cu CNP-ul „1681214109887,, salariul Net din fiecare lună de la data când valoare acestuia este mai mare decât 0.

```

declare @idangajat int;
select @idangajat=IDAngajat from Angajat where CNP ='1681214109887'
declare @start date;
select @start=DATEADD(DAY,2, MIN(DeLaData)) from Salarii where ValoareBruta>0 and
IDAngajat=@idangajat
declare @end date ;
select @end=
case when LaData is null then GETDATE()
when LaData is not null then LaData
end
from Salarii where IDAngajat=@idangajat;

with c as
(
select S.IDAngajat as idA, @start as data_inceput from Salarii as S
where (S.DeLaData <@start and( S.LaData>@start or S.LaData is null)) and
S.IDAngajat=@idangajat
union all
select idA,DATEADD(MONTH,1, data_inceput )
from c
where DATEADD(MONTH,1,data_inceput)<= @end
)
select DATENAME(MONTH,c.data_inceput)as luna , YEAR(c.data_inceput) as an,
(S.ValoareBruta*(1-(S.ValoareImpozit)/100)) as valoareSalariu from c
inner join Salarii as S

```

SELECT

```

on c.idA=S.IDAngajat
where ((c.data_inceput>S.DeLaData and c.data_inceput<s.LaData) or
(c.data_inceput>S.DeLaData and s.LaData is null))
and ValoareBruta>0

```

	luna	an	valoareSalariu				
1	January	2013	900	7	July	2013	900
2	February	2013	900	8	August	2013	900
3	March	2013	900	9	September	2013	900
4	April	2013	900	10	October	2013	900
5	May	2013	900	11	November	2013	900
6	June	2013	900	12	December	2013	900
				13	January	2014	900

44. Afișați pentru angajatul Anton Cristian salariul și numărul de luni pentru care a primit fiecare sumă.

```

declare @idangajat int;
select @idangajat=IDAngajat from Angajat where Nume like 'Anton' and Prenume like 'Cristian'
declare @startdata date;
declare @enddata date;
select @enddata=
    case when LaData is null then GETDATE()
    when LaData is not null then LaData
    end
from Salarii where IDAngajat=@idangajat
select @startdata=DATEADD (day,1,MIN(s.DeLaData))
from Salarii as s where ValoareBruta>0 and IDAngajat=@idangajat;

with c as
(
    select @idangajat as idA, @startdata as data_inceput from Salarii as S
    where (S.DeLaData <@startdata and ( S.LaData>@startdata or S.LaData is null))
and IDAngajat=@idangajat
union all
select idA,DATEADD(MONTH,1, data_inceput )
from c
where DATEADD(MONTH,1,data_inceput)<= @enddata
)
select (S.ValoareBruta*(1-(S.ValoareImpozit)/100)) as Salariu, count (c.data_inceput) as
nrluni from c
inner join Salarii as S
on c.idA=S.IDAngajat
where ((c.data_inceput>S.DeLaData and c.data_inceput<s.LaData)
or (c.data_inceput>S.DeLaData and s.LaData is null))
and ValoareBruta>0
group by S.ValoareBruta, S.ValoareImpozit

```

	Salariu	nrluni
1	900	36
2	1125	36
3	1500	29

45. Care este valoarea totală a salariilor date către angajați de la deschiderea clubului.

Această valoare se va calcula cu ajutorul unui while.

```
declare @total int, @counter int, @maxid int
select @total=0, @counter=min(IDAngajat), @maxid=max(IDAngajat) from Angajat
while(@counter is not null and @counter<=@maxid)
begin
    declare @suma int =0;
    exec ValoareSalariiTotal
        @idangajat=@counter,
        @valsalar=@suma output;
    set @total=@total+@suma
    set @counter=@counter+1;
end
select @total
```

(No column name)
656500

INSERT

46. Să se insereze în baza de date doi sportivi noi pe nume Bondar Cristian și Bondar Lavinia cu CNP urile 5050117141329 și 6081124143532. Data de înscriere este 15 mai 2016, iar fata are probleme la piciorul stang deoarece abia a fost operată.

```
insert into Sportivi(Nume, Prenume, CNP,DataInscriere)
values('Bondar', 'Cristian', '5050117141329', '2016-05-15'),
('Bondar', 'Lavinia', '6081124143532', '2016-04-15')
```

```
update Sportivi
set Observatii='Probleme picior stang, Operata'
where Nume like 'Bondar' and Prenume like 'Lavinia'
```

47. Pentru copii adăugați mai devreme selectați-le ca și sport, Karate și MMA pentru băiat și Karate pentru fată, toate de la data înscrierii. Pentru ușurință am verificat care sunt id-urile sportivilor și al sporturilor la care vom face referire.

```
select * from Sportivi
where Nume like 'Bondar' and Prenume like 'Cristian'
select * from Sportivi
where Nume like 'Bondar' and Prenume like 'Lavinia'
select * from Sporturi
delete from Sportivi where IDSportiv>42
insert into InregistrariSportiv(IDSportiv, IDSport, DataIncepere)
values
(41,3, '2016-05-15'),
(41,4, '2016-05-15'),
(42,3, '2016-05-15')
```

48. Introduceți o nouă competiție „Cupa Junior” în data de 2017-02-28, care ține o zi pentru sportul cu id-ul 2, organizată de Primăria Constanța.

```
select * from TipEveniment
where Denumire like 'Competitie'
insert into Evenimente(IDTip, IDTipSport,Nume, Locatie,
[Nume Club Organizator],DataIncepere,DataFinalizare)
values (1,2, 'Cupa Junior', 'Constanța', 'Primăria Constanța', '2017-02-28', '2017-03-01')
```

49. La competiția introdusă anterior să se adauge ca și participanți sportivii cu id-urile, premiile și categoriile: 6->1(U18 -60kg); 18->3(U16 -40kg); 16->2(U18 -70kg);18->participare(U18 -45kg)

```
select top 1 * from Evenimente
order by IDEveniment desc

insert into Premii(IDSportiv, IDEveniment, Premiu, Categorie) values
(6,19, '1', 'U18 -60kg'),
(18,19, '3', 'U18 -40kg'),
(16,19, '2', 'U18 -70kg'),
(18,19, 'participare', 'U18 -45kg')
```

50. Plățile pentru această competiție s-au efectuat în două tranșe, una cu o lună mai repede în valoare de 200lei pentru cazarea sportivilor și una în ziua concursului în valoare de 100 lei ce a constat în taxa de concurs.

```
insert into TaxeEvenimente(IDEveniment, Valoare, Data, Observatii)
values (19, 200, convert(date, '2017-01-28'), 'Cazare'),
(19, 100, convert(date, '2017-01-28'), 'Taxe')
```

51. Tot pentru acest eveniment, sportivii au avut nevoie de un însoțitor, cel care a participat are id-ul 5

```
insert into Insoțitori(IDEveniment, IDAngajat)
values (19, 5)
```

52. Introduceți toate taxele din 2011 ale sportivilor ce participă la macar un sport.

Au fost introduse pentru anii 2011-2016. Pentru introducerea totală a taxelor s-au inserat pe rând taxele pe an în primii 2 ani și apoi pe lună cu aceeași data la care se modifica anul/luna pentru ultimele inserări pentru a putea ține cont de toți sportivii înscriși la momentul acela.

```
insert into TaxeSportivi(IDSportiv, Data, Valoare) values
(1, convert(date, '2014-01-16'), 50),
(2, convert(date, '2014-01-16'), 50),
(3, convert(date, '2014-01-16'), 50),
(4, convert(date, '2014-01-16'), 50),
(5, convert(date, '2014-01-16'), 50),
(6, convert(date, '2014-01-16'), 50),
(7, convert(date, '2014-01-16'), 50),
(11, convert(date, '2014-01-16'), 50),
(12, convert(date, '2014-01-16'), 50),
(16, convert(date, '2014-01-16'), 50),
(18, convert(date, '2014-01-16'), 50),
(19, convert(date, '2014-01-16'), 50),
(22, convert(date, '2014-01-16'), 50),
(24, convert(date, '2014-01-16'), 50),
(25, convert(date, '2014-01-16'), 50),
(26, convert(date, '2014-01-16'), 50),
(27, convert(date, '2014-01-16'), 50),
(31, convert(date, '2014-01-16'), 50),
(34, convert(date, '2014-01-16'), 50),
(34, convert(date, '2014-01-16'), 50),
(39, convert(date, '2014-01-16'), 50),
(41, convert(date, '2014-01-16'), 50),
(42, convert(date, '2014-01-16'), 50)
```

53. Să se adauge un noile tipuri de venit Sponsorizare și Donație. Iar ca surse să se introducă Primăria Covasna, Berits, Fram. Să se introducă pentru acestea și înregistrări în tabela Venituri.

```
insert into TipVenituri (Denumire)
values ('Donatii'), ('Sponsorizare')
```

```
insert into SursaVenituri(Nume)
values ('Bertis'), ('Primaria Covasna'), ('FRAM')
```

INSERT

```
insert into Venituri(IDSursa, IDTipVenit, Valoare, Data) values
(2,4,1200, '2014-04-12'),
(3,5,3000, '2015-06-23'),
(4,5,800, '2016-02-02'),
(2,5,1000, '2016-04-18'),
(3,4,1200, '2016-08-14')
```

	IDVenit	IDSursa	IDTipVenit	Valoare	Data
1	1	2	4	1200	2014-04-12
2	2	3	5	3000	2015-06-23
3	3	4	5	800	2016-02-02
4	4	2	5	1000	2016-04-18
5	5	3	4	1200	2016-08-14

54. Să se angajeze o nouă persoană, Crivat Raluca, cu CNP-ul 2781215187662, cu un contract de muncă începând din 2017-01-05, care va lucra ca și fotograf. Pentru acest tip de angajat se va crea o nouă înregistrare.

```
insert into TipAngajat(Denumire) values ('Fotograf')
select *from TipAngajat where Denumire like 'Fotograf'
insert into Angajat(Nume, Prenume, CNP, DataIncepereContract, IdTip)
values ('Crivat', 'Raluca', '2781215187662', '2017-01-05', 7)
```

55. Să se adauge facturile dintre anii 2011-2016. S-a recurs la aceeași metodă ca la introducerea de taxeSportiv.

```
insert into Facturi(IDTip, Valoarea, Data) values
(1,200, '2011-01-13'),
(2,150, '2011-01-13'),
(3,150, '2011-01-13'),
(1,200, '2011-02-13'),
(2,190, '2011-02-13'),
(3,120, '2011-02-13'),
(1,330, '2011-03-13'),
(2,180, '2011-03-13'),
(3,190, '2011-03-13'),
(1,210, '2011-04-13'),
(2,120, '2011-04-13'),
(3,140, '2011-04-13'),
(1,310, '2011-05-13'),
(2,120, '2011-05-13'),
(3,270, '2011-05-13'),
(1,200, '2011-06-13'),
(2,190, '2011-06-13'),
(3,200, '2011-06-13'),
(1,190, '2011-07-13'),
(2,170, '2011-07-13'),
(3,180, '2011-07-13'),
(1,230, '2011-08-13'),
(2,210, '2011-08-13'),
(3,150, '2011-08-13'),
(1,250, '2011-09-13'),
(2,210, '2011-09-13'),
```

```
(3,210,'2011-09-13'),  
(1,310,'2011-10-13'),  
(2,220,'2011-10-13'),  
(3,270,'2011-10-13'),  
(1,320,'2011-11-13'),  
(2,250,'2011-11-13'),  
(3,230,'2011-11-13'),  
(1,250,'2011-12-13'),  
(2,270,'2011-12-13'),  
(3,230,'2011-12-13')
```

56. S-a realizat o comandă în valoare de 500lei, la data de 2017-03-07.

```
declare @id as int  
select @id=IDTip from TipFactura where Denumire like 'Comanda'  
insert into Facturi(IDTip,Valoarea,Data) values  
(@id, 700,'2017-03-10')
```

57. Să se insereze un angajat nou, ce va ajuta la testarea triggerului de ștergere.

Adăugați-l atât la însoțitori cât și la salarii.

```
insert into Angajat (Nume,Prenume,CNP,IdTip,DataIncepereContract)  
values ('Angajat', 'Gresit', '1234567891234',2,'2013-09-12')
```

```
select * from Angajat where CNP='1234567891234'  
insert into Insoțitori(IDAngajat,IDEveniment) values (10,18)  
insert into Salarii(IDAngajat,ValoareImpozit,ValoareBruta,DeLaData)  
values (10,23,1232,'2013-04-12')
```

58. Inserați tipurile de venituri Donații, Sponsorizări și sursele Primăria Covasna, Bertis, FRAM.Să se introducă venituri aferente pentru anii 2011-2016.

```
insert into TipVenituri (Denumire)  
values ('Donatii'),('Sponsorizare')
```

```
insert into SursaVenituri(Nume)  
values ('Bertis'),('Primaria Covasna'),('FRAM')
```

```
insert into Venituri(IDSursa,IDTipVenit,Valoare,Data) values  
(2,4,1200,'2014-04-12'),  
(3,5,3000,'2015-06-23'),  
(4,5,800,'2016-02-02'),  
(2,5,1000,'2016-04-18'),  
(3,4,1200,'2016-08-14')
```

59. Adăugați un salariu în valoare de 2000 lei cu impozit de 25% pentru angajatul x începând cu 1 ian 2020.

```
insert into Salarii ( IDAngajat,ValoareBruta,ValoareImpozit,DeLaData)  
values (4,2000,25,'2020-01-01')
```

60. Să se insereze în taxeEvenimente costurile aferente fiecărui eveniment.

61.

```
insert into TaxeEvenimente(IDEveniment,Data,Valoare,Observatii) values
(1, CONVERT(Date, '2011-09-12'), 200, 'Taxe Inscriere'),
(2, CONVERT(Date, '2012-04-15'), 400, 'Taxe Inscriere'),
(2, CONVERT(Date, '2012-04-01'), 600, 'Cazare si masa'),
(3, CONVERT(Date, '2012-09-01'), 1000, 'Cazare si materiale'),
(4, CONVERT(Date, '2012-10-03'), 800, 'Cazare si mancare'),
(5, CONVERT(Date, '2013-09-20'), 1500, 'Cazare, transport, masa'),
(6, CONVERT(Date, '2013-11-12'), 200, 'Taxe inscriere'),
(6, CONVERT(Date, '2013-11-12'), 100, 'Transport'),
(7, CONVERT(Date, '2013-12-05'), 200, 'Transport'),
(7, CONVERT(Date, '2013-12-01'), 300, 'Cazare, masa'),
(8, CONVERT(Date, '2013-12-05'), 50, 'Gustari'),
(9, CONVERT(Date, '2014-05-25'), 500, 'Taxe'),
(10, CONVERT(Date, '2014-07-12'), 300, 'Cazare'),
(11, CONVERT(Date, '2014-09-13'), 200, 'Taxe'),
(11, CONVERT(Date, '2014-09-10'), 100, 'Gustari'),
(12, CONVERT(Date, '2016-02-18'), 200, 'Transport'),
(13, CONVERT(Date, '2016-05-02'), 0, 'Taxa Verde'),
(15, CONVERT(Date, '2016-06-03'), 300, 'Cazare'),
(16, CONVERT(Date, '2016-12-05'), 200, 'Masa'),
(16, CONVERT(Date, '2016-12-05'), 250, 'Cazare'),
(17, CONVERT(Date, '2016-12-15'), 300, 'Taxe'),
(17, CONVERT(Date, '2016-12-19'), 200, 'Transport'),
(18, CONVERT(Date, '2017-03-10'), 300, 'Cazare'),
(18, CONVERT(Date, '2017-03-14'), 250, 'Taxe'),
(19, CONVERT(Date, '2017-02-25'), 200, 'Taxe'),
(21, CONVERT(Date, '2017-02-03'), 180, 'Masa')
```

62. Adăugați un nou eveniment ce se organizează în Sugaș Bai, în data de 01-05-2017 și ține o zi. Evenimentul este organizat de către CS și este o excursie deschisă pentru sportivi și angajații de la toate sporturile.

```
insert into Evenimente(Nume,[Nume Club
Organizator],Locatie,IDTip,IDTipSport,DataIncepere,DataFinalizare)
select 'Excursie Sugas Bai', 'CS', 'Sugas Bai', T.IDTip,S.IDSport, '2017-
05-01', '2017-05-02'
from TipEveniment as T, Sporturi as S
where T.Denumire like 'Excursie' and S.Denumire like 'Toate'
```

63. Pentru acest eveniment introduceți o taxă de 100 lei, cu observația „Dulciuri”.

```
insert into TaxeEvenimente(IDEveniment,Valoare,Data,Observatii)
select IDEveniment, 100,DataIncepere,'Dulciuri' from Evenimente
where Evenimente.Nume like 'Excursie Sugas Bai' and Evenimente.Locatie
like 'Sugas Bai'
```

64. Pentru același eveniment se adaugă ca și participanți angajatul Stoica Ionut și ca sportivi cei cu id-urile: 26,27,31,34,37,39,41

```
insert into Insotitori(IDAngajat, IDEveniment)
select a.IDAngajat, e.IDEveniment from Angajat as a, Evenimente as e
where E.Nume like 'Excursie Sugas Bai' and E.Locatie like 'Sugas Bai'
and
      A.Nume like 'Stoica' and A.Prenume like 'Ionut'
```

```
insert into Premii(IDEveniment, IDSportiv, Categorie)
values (25,26, '-'),
(25,27, '-'),
(25,31, '-'),
(25,34, '-'),
(25,37, '-'),
(25,39, '-'),
(25,41, '-')
```

UPDATE

65. Modificați tabelul Evenimente astfel încât coloana IDEveniment să permită valoarea NULL.

```
alter table Premii  
alter column IDEveniment int null
```

66. Modifică astfel încât toate evenimentele organizate de "Aquila" să aibă IDTipSport =

```
update Evenimente  
set IDTipSport =1  
where [Nume Club Organizator] like 'Aquila'
```

67. În tabela ce manageriază taxele evenimentelor se va adăuga o coloană „Observatii” de tip text care să permită null.

```
alter table TaxeEvenimente  
add Observatii nvarchar(100)
```

68. Perioada contractului angajatei Crivat Raluca a fost provizoriu, actualizați baza de date cu ziua finalizării contractului, 2017-07-05.

```
update Angajat  
set DataFinalizareContract='2017-07-05'  
where Nume like 'Crivat' and Prenume like 'Raluca'
```

	IDAngajat	Nume	Prenume	CNP	DataIncepereContract	DataFinalizareContract	IdTip
1	9	Crivat	Raluca	2781215187662	2017-01-05	2017-07-05	7

69. Tuturor comenzilor din data de martie 2017 le-au fost aplicate un comision, astfel pretul total al acestora a fost 75% din valoare inițială.

Initial:

	IDFactura	Data	Valoarea	IDTip	IDTip	Denumire
1	253	2017-03-07	500	4	4	Comanda
2	254	2017-03-10	700	4	4	Comanda

```
select * from Facturi  
inner join TipFactura  
on Facturi.IDTip=TipFactura.IDTip  
where TipFactura.Denumire like 'Comanda' and Facturi.Data like '2017-03%'
```

```
update F  
set F.Valoarea=Valoarea*75/100  
from Facturi as F  
inner join TipFactura as T  
on T.IDTip=F.IDTip  
where T.Denumire like 'Comanda' and F.Data like '2017-03%'
```

	IDFactura	Data	Valoarea	IDTip	IDTip	Denumire
1	253	2017-03-07	375	4	4	Comanda
2	254	2017-03-10	525	4	4	Comanda

70. La introducerea salariilor a fost inclusă o eroare, începând cu noul an(2018) salariile au fost modificate pentru toți instructorii.(adaugați 2017-12-31 unde există null și angajatul este instructor). Adăugați un salariu nou de 1700lei, impozit de 20% începând cu 2018-01-01 pentru acești angajați.

```
update S
set S.LaData = '2017-12-31'
from Salarii as S
inner join Angajat as A
on S.IDAngajat=A.IDAngajat
inner join TipAngajat as T
on T.IDTip=A.IdTip
where T.Denumire like 'Instructor' and S.LaData is null
```

```
select S.IDAngajat from Salarii as S
inner join Angajat as A
on S.IDAngajat=A.IDAngajat
inner join TipAngajat as T
on T.IDTip=A.IdTip
where T.Denumire like 'Instructor' and S.LaData like '2017-12-31'
```

```
insert into Salarii(IDAngajat,ValoareBruta, ValoareImpozit,DeLaData)
values (5,1700,20, '2018-01-01'),
(6,1700,20, '2018-01-01')
```

11	13	1600	25	5	2017-01-01	2017-12-31
12	14	1400	20	6	2017-01-01	2017-12-31
13	15	1000	20	8	2020-01-01	NULL
14	16	1700	20	5	2018-01-01	NULL
15	17	1700	20	6	2018-01-01	NULL

71. Sportivul Bondar Lavinia pleacă din clubul sportiv în 2017-04-15. Updateați plecarea ei de la toate sporturile participante.

```
update S
set S.DataPlecure='2017-05-15'
from Sportivi as S
where S.Nume like 'Bondar' and S.Prenume like 'Lavinia'
```

```
update I
set I.DataFinalizare ='2017-05-15'
from Sportivi as S
inner join InregistrariSportiv as I
on I.IDSportiv=S.IDSportiv
where S.Nume like 'Bondar' and S.Prenume like 'Lavinia'
```

```
select *from Sportivi as S
where S.Nume like 'Bondar' and S.Prenume like 'Lavinia'
select *
from Sportivi as S
inner join InregistrariSportiv as I
on I.IDSportiv=S.IDSportiv
where S.Nume like 'Bondar' and S.Prenume like 'Lavinia'
```


72. Modificați toate tipurile de premii, „1 devine locul 1”, 2 devine „locul 2” și 3 devine „locul 3” din tabela Premii.

```
update Premii
set Premiu =
(
    case
        when Premiu like '%1%' then 'locul 1'
        when Premiu like '%2%' then 'locul 2'
        when Premiu like '%3%' then 'locul 3'
        else Premiu
    end
)
```

	IDPremiu	IDSportiv	IDEveniment	Premiu	Categorie	Observatii
1	1	1	1	locul 1	U12 -34kg	NULL
2	2	2	1	locul 1	U12 -40kg	NULL
3	3	3	1	locul 2	U12 -34kg	NULL
4	4	4	1	locul 3	U12 -36kg	NULL
5	5	1	2	locul 1	U12 -34kg	NULL
6	6	2	2	locul 1	U12 -40kg	NULL
7	7	3	2	locul 1	U12 -45kg	NULL
8	8	4	2	locul 2	U12 -36kg	NULL

73. În urma a două evenimente, s-au încurcat id-urile. Schimbați id-ul ultimei competiții cu id-ul ultimului cantonament în tabelele Premii și Însoțitori.

Înainte(Premii):

	IDPremiu	IDSportiv	IDEveniment	Premiu	Categorie	Observatii
1	19	5	18	-	-	NULL
2	20	6	18	-	-	NULL
3	21	7	18	-	-	NULL
4	22	1	18	-	-	NULL
5	87	1	4	locul 1	U18 -65kg	Premiu special FRAM
6	88	1	4	locul 1	U18 -70kg	Premiu special FRAM
7	89	3	4	locul 1	U18 -60kg	NULL
8	90	3	4	locul 2	U18 -65kg	NULL
9	91	16	4	locul 1	U18 -50kg	NULL

```
update P
set P.IDEveniment=
    case P.IDEveniment
        when (select top 1 E.IDEveniment from Evenimente as E
              inner join TipEveniment as T
              on T.IDTip=E.IDTip
              where T.Denumire like 'Cantonament'
              order by E.DataIncepere desc)
        then (select top 1 E.IDEveniment from Evenimente as E
              inner join TipEveniment as T
              on T.IDTip=E.IDTip
              where T.Denumire like 'Competitie'
              order by E.DataIncepere desc)
        when (select top 1 E.IDEveniment from Evenimente as E
              inner join TipEveniment as T
```

```

        on T.IDTip=E.IDTip
        where T.Denumire like 'Competitie'
        order by E.DataIncepere desc)
    then (select top 1 E.IDEveniment from Evenimente as E
        inner join TipEveniment as T
        on T.IDTip=E.IDTip
        where T.Denumire like 'Cantonament'
        order by E.DataIncepere desc)
    else P.IDEveniment
end
from Premii as P

```

	IDPremiu	IDSportiv	IDEveniment	Premiu	Categorie	Observatii
1	19	5	4	-	-	NULL
2	20	6	4	-	-	NULL
3	21	7	4	-	-	NULL
4	22	1	4	-	-	NULL
5	87	1	18	locul 1	U18 -65kg	Premiu special FRAM
6	88	1	18	locul 1	U18 -70kg	Premiu special FRAM
7	89	3	18	locul 1	U18 -60kg	NULL
8	90	3	18	locul 2	U18 -65kg	NULL

Analog Insoțitori:

```

update P
set P.IDEveniment=
    case P.IDEveniment
        when (select top 1 E.IDEveniment from Evenimente as E
            inner join TipEveniment as T
            on T.IDTip=E.IDTip
            where T.Denumire like 'Cantonament'
            order by E.DataIncepere desc)
        then (select top 1 E.IDEveniment from Evenimente as E
            inner join TipEveniment as T
            on T.IDTip=E.IDTip
            where T.Denumire like 'Competitie'
            order by E.DataIncepere desc)
        when (select top 1 E.IDEveniment from Evenimente as E
            inner join TipEveniment as T
            on T.IDTip=E.IDTip
            where T.Denumire like 'Competitie'
            order by E.DataIncepere desc)
        then (select top 1 E.IDEveniment from Evenimente as E
            inner join TipEveniment as T
            on T.IDTip=E.IDTip
            where T.Denumire like 'Cantonament'
            order by E.DataIncepere desc)
        else P.IDEveniment
    end
from Insotitori as P

```

74. Mutați toți sportivii de la BJJ la Ju-Jitsu.

```
update InregistrariSportiv
set IDSport=(select IDSport from Sporturi where Denumire like
'JuJitsu')
where IDSport=(select IDSport from Sporturi where Denumire like 'BJJ')
```

75. Modificați Locația Cupei Junior din data de 2017-02-28 din Constanța în 1 Mai.

```
update Evenimente
set Locatie='1 Mai'
where Locatie like 'Constanta' and Nume like 'Cupa Junior' and
DataIncepere like '2017-02-28'
```

76. Modificați data ultimei competiții, aceasta s-a întâmplat cu o lună înainte de data notată.

17	18	Campionatul National	FRAM	Cluj	2017-01-14	1	2017-01-16
18	19	Cupa Junior	Primaria Constanța	1 Mai	2017-02-28	1	2017-03-01
19	21	Cupa Dino	Victoria	Cluj	2017-02-04	1	2017-02-06

```
update Evenimente
set DataIncepere=DATEADD(month, 1 ,DataIncepere),
DataFinalizare=DATEADD(month, -1, DataFinalizare)
where IDEveniment=(select top 1 IDEveniment from Evenimente where
IDTip=1
order by DataIncepere desc)
```

17	18	Campionatul National	FRAM	Cluj	2017-02-14	1	2017-02-16
18	19	Cupa Junior	Primaria Constanța	1 Mai	2017-01-28	1	2017-02-01
19	21	Cupa Dino	Victoria	Cluj	2017-02-04	1	2017-02-06

Modificarea s-a făcut la înregistrarea 18.(id=19)

77. Să se modifice anul din 2014 în 2016 pentru toate taxele cu id-ul mai mare de 1054.Să se modifice și luna, din ianuarie să devină decembrie pentru aceleași taxe.

```
update TaxeSportivi
set Data=DATEADD(yy,2,Data)
where IDTaxa>1054

update TaxeSportivi
set Data=DATEADD(MONTH,11,Data)
where IDTaxa>1054
```

```
select * from TaxeSportivi
where IDTaxa>1054
```

	IDTaxa	IDSportiv	Valoare	Data
1	1055	1	50	2016-12-16
2	1056	2	50	2016-12-16
3	1057	3	50	2016-12-16
4	1058	4	50	2016-12-16
5	1059	5	50	2016-12-16
6	1060	6	50	2016-12-16

7	1061	7	50	2016-12-16
8	1062	11	50	2016-12-16
9	1063	12	50	2016-12-16
10	1064	16	50	2016-12-16
11	1065	18	50	2016-12-16
12	1066	19	50	2016-12-16
13	1067	22	50	2016-12-16
14	1068	24	50	2016-12-16
15	1069	25	50	2016-12-16
16	1070	26	50	2016-12-16
17	1071	27	50	2016-12-16
18	1072	31	50	2016-12-16
19	1073	34	50	2016-12-16
20	1074	34	50	2016-12-16
21	1075	39	50	2016-12-16
22	1076	41	50	2016-12-16
23	1077	42	50	2016-12-16

78. Modificați tipul de sport de la Cantonamentul din Bran din anul 2012 să fie pentru toate tipurile de sport existente.

```
update Evenimente
set IDTipSport=(select IDSport from Sporturi where Denumire like
'Toate')
where Nume like 'Cantonament Bran' and Locatie like 'Bran' and
YEAR(DataIncepere) ='2012'
```

79. Modificați prețul taxei ce are ca observații TaxaVerde (introducere taxa la gratar) și este pentru grătarul din data x la nr de sportivi participanți *10 lei.

```
with c as
(
    select count(*)as NRparticipanti , Evenimente.IDEveniment from
Premii
    inner join Evenimente
    on Evenimente.IDEveniment=Premii.IDEveniment
    inner join TaxeEvenimente
    on TaxeEvenimente.IDEveniment=Evenimente.IDEveniment
    where TaxeEvenimente.Observatii like 'Taxa Verde'
    group by Evenimente.IDEveniment
)
update TaxeEvenimente
set Valoare=(select NRparticipanti from c)*10
where IDEveniment=(select IDEveniment from c)
80. Pentru toate taxele sportivilor începând din 2012 dublați valoarea acestora.
update TaxeSportivi
set Valoare=Valoare*2
where YEAR (Data) >='2012'
```

DELETE

81. Să se șteargă Angajatul Crivat Robert și să se verifice că a fost șters din toate tabele care îl conțin cu ajutorul triggerului specific.

```
delete from Angajat  
where Nume like 'Crivat' and Prenume like 'Robert'
```

82. Să se șteargă Excursia Sugas Bai cu locația în Șugaș Bai din tabelul Evenimente și toate înregistrările ce fac referire la aceasta din celelalte tabele.

```
delete TaxeEvenimente from TaxeEvenimente  
inner join Evenimente  
on TaxeEvenimente.IDEveniment=Evenimente.IDEveniment  
where Evenimente.Nume like 'Excursie Sugas Bai' and Evenimente.Locatie  
like 'Sugas Bai'
```

```
delete Insotitori from Insotitori  
inner join Evenimente  
on Insotitori.IDEveniment=Evenimente.IDEveniment  
where Evenimente.Nume like 'Excursie Sugas Bai' and Evenimente.Locatie  
like 'Sugas Bai'
```

```
delete Premii from Premii  
inner join Evenimente  
on Premii.IDEveniment=Evenimente.IDEveniment  
where Evenimente.Nume like 'Excursie Sugas Bai' and Evenimente.Locatie  
like 'Sugas Bai'
```

```
delete from Evenimente  
where Nume like 'Excursie Sugas Bai' and Locatie like 'Sugas Bai'
```

83. Ștergere un tip de factură și toate facturile de acel tip.

```
delete Facturi from Facturi  
inner join TipFactura  
on TipFactura.IDTip=Facturi.IDTip  
where TipFactura.Denumire like 'Apa2'
```

```
delete from TipFactura  
where Denumire like 'Apa2'
```

84. Ștergeți sportul la care nu există nici un sportiv la momentul actual.

```
with nul as  
(  
    select IDSport, count(*) as nrplecati from InregistrariSportiv as  
i  
    where DataFinalizare is null  
    group by IDSport  
)  
, nnul as  
(  
    select IDSport, count(*) as nractivi from InregistrariSportiv as  
i
```

```

        where DataFinalizare is not null
        group by IDSport
    )
delete from Sporturi where IDSport=
(select nul.IDSport from nul
inner join nnul
on nul.IDSport=nnul.IDSport
where nul.nrplecati=nnul.nractivi)

```

Datorita triggerului ce adaugă expirat pentru sporturile șterse, sportul nu va fi șters cu adevărat ci i se va adăuga „_expirat” la final.

85. Ștergeți din ultima lună taxele sportivilor ce au 3 premii în acea lună.

```

with c as
(
    select month(e.DataIncepere) as luna, p.IDSportiv, COUNT(*) as
nrpremii from Premii as P
    inner join Evenimente as e
    on e.IDEveniment=p.IDEveniment
    group by E.DataIncepere, p.IDSportiv, e.IDEveniment
),a as
(
    select luna,IDSportiv, nrpremii from c
    where nrpremii>=3
)
delete TaxeSportivi from TaxeSportivi
inner join a
on TaxeSportivi.IDSportiv=a.IDSportiv
where MONTH(Data) =a.luna

```

86. Ștergeți sursa de venit ce nu a adus o suma mai mare de 300lei.

Pentru aceasta am introdus o nouă sursă de venit și un venit de la aceasta.

```

insert into SursaVenituri(Nume) values('MEGA')

insert into Venituri(IDSursa,IDTipVenit,Valoare,Data)
values (20,4,200,'2017-03-14')

```

```

with c as
(
    select S.IDSursa,sum(v.Valoare) as val from Venituri as v
    inner join SursaVenituri as S
    on V.IDSursa=S.IDSursa
    group by S.IDSursa
)
delete Venituri from Venituri
inner join c
on Venituri.IDSursa=c.IDSursa
where c.val<300

with c(id) as
(

```

```

        select IDSursa from SursaVenituri
    except
    select IDSursa from Venituri
)
delete SursaVenituri from SursaVenituri
inner join c
on c.id=SursaVenituri.IDSursa

```

Am șters mai întâi toate veniturile surselor ce nu au adus 300 lei pentru a nu încălca constrângerile de key externe și apoi toate sursele ce nu au adus vreun venit.

87. Ștergeți tipul de venit Donație și toate veniturile de acest tip.

```

delete Venituri from Venituri as V
inner join TipVenituri as T
on T.IDTip=V.IDTipVenit
where T.Denumire like 'Donatii'

```

```

delete from TipVenituri
where Denumire like 'Donatii'

```

88. Ștergeți toate stagiile de la sportul BJJ, cat și toate înregistrările cu sportivi și angajați ce au participat la acesta.

```

delete Insotitori from Evenimente
inner join TipEveniment
on Evenimente.IDTip=TipEveniment.IDTip
inner join Sporturi
on Sporturi.IDSport=Evenimente.IDTipSport
inner join Insotitori
on Insotitori.IDEveniment=Evenimente.IDEveniment
where Sporturi.Denumire like 'BJJ'
and TipEveniment.Denumire like 'Stagiu'

```

```

delete Premii from Evenimente
inner join TipEveniment
on Evenimente.IDTip=TipEveniment.IDTip
inner join Sporturi
on Sporturi.IDSport=Evenimente.IDTipSport
inner join Premii
on Premii.IDEveniment=Evenimente.IDEveniment
where Sporturi.Denumire like 'BJJ'
and TipEveniment.Denumire like 'Stagiu'

```

```

delete Evenimente from Evenimente
inner join TipEveniment
on Evenimente.IDTip=TipEveniment.IDTip
inner join Sporturi
on Sporturi.IDSport=Evenimente.IDTipSport
where Sporturi.Denumire like 'BJJ'
and TipEveniment.Denumire like 'Stagiu'

```

89. Ștergeți toate evenimentele la care nu au participat nici o persoană.

```
with c as
(
    select IDEveniment from Evenimente
except
select e.IDEveniment from Evenimente as e
inner join Premii as p
on e.IDEveniment=p.IDEveniment
group by e.IDEveniment
)
delete TaxeEvenimente from TaxeEvenimente
inner join c
on c.IDEveniment=TaxeEvenimente.IDEveniment
```

```
with c as
(
    select IDEveniment from Evenimente
except
select e.IDEveniment from Evenimente as e
inner join Premii as p
on e.IDEveniment=p.IDEveniment
group by e.IDEveniment
)
delete Insotitori from Insotitori
inner join c
on c.IDEveniment=Insotitori.IDEveniment
```

```
with c as
(
    select IDEveniment from Evenimente
except
select e.IDEveniment from Evenimente as e
inner join Premii as p
on e.IDEveniment=p.IDEveniment
group by e.IDEveniment
)
delete Evenimente from Evenimente
inner join c
on c.IDEveniment=Evenimente.IDEveniment
```

90. Ștergeți toate taxele de la sportivi din luna Mai a anului 2017.

```
delete from TaxeSportivi
where Data like '2017-05%'
```


91. Ștergeți toate taxele primite de la sportivul Mihalache Cristina din anul curent.

```
delete TaxeSportivi from TaxeSportivi
inner join Sportivi
on Sportivi.IDSportiv=TaxeSportivi.IDSportiv
where Sportivi.Nume like 'Mihalache '
and Sportivi.Prenume like 'Cristina'
```

92. Ștergeți sportivul cu cele mai puține premii care participă la Karate. Pentru această ștergere se va activa triggerul existent și se vor șterge toate datele din tabelele necesare.

```
with c as
(
    select S.IDSportiv, count(*) as nrpremii from Premii as P
    inner join Sportivi as S
    on S.IDSportiv=P.IDSportiv
    inner join InregistrariSportiv as I
    on I.IDSportiv=S.IDSportiv
    inner join Sporturi as Sp
    on Sp.IDSport=I.IDSport
    where Sp.Denumire like 'Karate'
    group by S.IDSportiv
), a as
(
    select MIN(nrpremii) as m from c
)
delete Sportivi from Sportivi
inner join c
on c.IDSportiv=Sportivi.IDSportiv
inner join a
on c.nrpremii=a.m
```

93. Ștergeți angajatul care a participat la evenimentul Cupa Junior din locația 1 Mai.

```
delete Angajat from Angajat as a
inner join Insotitori as i
on a.IDAngajat=i.IDAngajat
inner join Evenimente as e
on e.IDEveniment=i.IDEveniment
where e.Nume like 'Cupa Junior' and e.Locatie like '1 Mai'
```

94. Ștergeți ultima taxă a sportivului care a luat două premii 1 la competiția Campionatul Național din Cluj de la data de 2017-01-14.

```
with c as
(
    select P.IDSportiv, e.IDEveniment,
           count(*) as nrpremii from Premii as P
    inner join Evenimente as e
    on p.IDEveniment=e.IDEveniment
    where p.Premiu like '%1%' and e.Nume like 'Campionatul National'
    and e.Locatie like 'Cluj' and e.DataIncepere like '2017-01-14'
```

```

        GROUP by P.IDSportiv,e.IDEveniment
    )

```

```

delete TaxeSportivi from c
inner join TaxeSportivi as t
on c.IDSportiv=t.IDSportiv
where nrpremii >=2

```

95. La adaugarea datelor s-a produs o greșeala, unui sportivi apărând astfel că au plătit taxa de mai multe ori în aceeași lună.Ștergeți aceste taxe în plus.

```

with c as
(
select IDSportiv, Data, ROW_NUMBER() over
(
    partition by
        IDSportiv,
        Data
    order by
        IDSportiv,
        Data
) nr
from TaxeSportivi
)
delete from c
where c.nr>=2

```

96. Ștergeți participarea la Cupa Junior din locația 1 mai a sportivului Manea Robert deoarece acesta s-a îmbolnavit și nu a mai ajuns.

```

delete Premii from Premii as p
inner join Evenimente as e
on p.IDEveniment=e.IDEveniment
inner join Sportivi as s
on s.IDSportiv=p.IDSportiv
where s.Nume like 'Manea' and s.Prenume like 'Robert'
and e.Nume like 'Cupa Junior' and e.Locatie like '1 Mai'

```

97. Ștergeți toate taxele ale caror id-uri aparțin intervalului [93,94].

```

delete from TaxeEvenimente
where IDTaxa between 93 and 94

```

27	91	19	200	2017-02-25	Taxe
28	92	21	180	2017-02-03	Masa
29	95	19	160	2017-02-28	Taxe

TRANZACȚII

98. Să se creeze o tranzacție de inserare a unui sportiv doar în cazul în care acesta este nascut înainte de 2013. Să se trateze această eroare cu textul „Copilul este prea mic pentru a lua parte la antrenamente”

```
begin tran VerifCNP_Sportiv
```

```
    insert into Sportivi(Nume,Prenume,CNP,DataInscriere)
    values ('Ghenea','Viorel','5171024142334',GETDATE());
    declare @CNP char(13);
    SELECT @CNP=CNP FROM Sportivi where Nume like 'Ghenea' and Prenume
like 'Viorel';
    if SUBSTRING(@CNP,2,2) > '13'
        begin
            print 'Copilul este prea mic pentru a face antrenamente';
            rollback;
        end
    else
        begin
            commit;
        end
end
```

99. Să se creeze o tranzacție pentru ștergerea unei categorii de factură, aceasta se poate șterge după ce se crează o categorie echivalentă și toate înregistrările specifice acesteia se transferă la noua categorie.

```
declare @errnum as int;
begin tran StergereFactura
    delete from TipFactura
    where Denumire='Curent';
    set @errnum=@@ERROR
    if @errnum<>0
        begin
            print 'eroare la stergere'
            begin tran
                insert into TipFactura (Denumire) values ('Electricitate');
                set @errnum=@@ERROR;
                if @errnum <> 0
                    begin
                        print 'Eroare la insertie';
                        rollback;
                    end
            end
        else
            begin
                commit;
                begin tran
                    update Facturi
                    set IDTip= (select IDTip from TipFactura where
Denumire like 'Electricitate')
```

```
where IDTip=(select IDTip from TipFactura where
Denumire like 'Curent')
set @errnum=@@ERROR
if @errnum<>0
begin
    print 'Eroare la update';
    rollback;
end
else
begin
    commit;
end
end
delete from TipFactura where Denumire like 'Curent';
end
else
begin
    commit;
end
```

100. Să se verifice ca la inserarea unui nou angajat acesta să nu existe, în cazul în care există se va da un mesaj corespunzător și se va rescrie în data de încheiere contract NULL. În cazul în care nu există angajatul atunci se va insera în baza de date. Pentru aceasta se va folosi triggerul *InainteDeAngajat*.

```
begin try
begin tran inserareAngajat
insert into Angajat(Nume,Prenume,DataIncepereContract,IdTip,CNP)
values ('Crivat','Robert','2018-01-04',7,'1781215187662');
commit;
end try
begin catch
if ERROR_NUMBER() =50000
begin
    print 'Angajatul exista. Se prelungeste contractul';
    rollback;
    update Angajat
    set DataFinalizareContract=NULL
    where CNP like '2781215187662'
end
end catch
```

101. Să se efectueze o tranzacție ca la ștergerea unui eveniment, în cazul în care se încalcă (cheia externa) să trimită un mesaj specific și să nu efectueze ștergerea, iar în cazul în care nu există referiri la acesta să se poată șterge și să se returneze un mesaj specific.

```
begin try
begin tran
    delete from Evenimente
    where Nume like 'TEST' and Locatie like 'test' and DataIncepere
like '2019-02-02'
    print 'Eveniment sters cu succes';
    commit;
end try
begin catch
if ERROR_NUMBER() = 547
begin
    print 'Acest eveniment nu se poate sterge';
    rollback;
end

(0 rows affected)
Acest eveniment nu se poate sterge

(1 row affected)
Eveniment sters cu succes
```

102. Să se creeze o tranzacție la inserarea unei noi categorii de angajați ca aceasta să nu existe deja, se va verifica denumirea categoriei și se va returna un mesaj corespunzător.

```
begin tran
    if exists(select COUNT(*) from TipAngajat where Denumire like
'Instructor' having count(*)>=1 )
    begin
        print 'Acest tip nu se poate introduce, verificati sa nu
fie deja introdusa'
        rollback;
    end
    else
    begin
        insert into TipAngajat(Denumire) values ('Instructor')
        commit;
    end
end
```

103. Să se creeze o tranzacție la introducerea unui însoțitor, să existe angajatul și evenimentul. Tratați aceste erori.

```
begin tran
insert into Insotitori(IDAngajat,IDEveniment)
select a.IDAngajat,e.IDEveniment from Angajat as a, Evenimente as
e
where a.Nume like 'Stoihca' and a.Prenume like 'Ionut' and
      e.Nume like 'Cupa Dino' and Locatie like 'Cluj';
if @@ROWCOUNT =0
begin
print 'Nu exista posibilitatea ca acest angajat sa
participe la acest eveniment.'
rollback
end
else
begin
print 'Insotitor adaugat cu succes'
commit;
end;
```

104. Să se creeze o tranzacție la introducerea unui premiu, să existe atât sportivul(care să fie activ la data curentă) cât și evenimentul.

```
begin try
begin tran
insert into Premii(IDEveniment,IDSportiv,Categorie)
select E.IDEveniment, S.IDSportiv, '-' from Sportivi as S,
Evenimente as e
where s.Nume like 'Stoica' and S.Prenume like 'Andrei' and
s.DataPlecare is null and
      e.Nume like 'Cupa Dino' and Locatie like 'Cluj';
if @@ROWCOUNT =0
begin
print 'Nu exista posibilitatea ca acest sportiv sa
participe la acest eveniment. Nu mai este inscris la acest sport'
rollback
end
else
begin
print 'Sportiv adaugat cu succes'
commit;
end;
end try
begin catch
if ERROR_NUMBER()=50002
begin
print 'Sportivul nu participa la tipul de sport care se
sustine la acest eveniment'
rollback
end
end catch
```

105. Să se creeze o tranzacție care să trateze erorile de încălcare a cheii străine la ștergerea unui angajat.

```
begin try
begin tran
    delete from Angajat
    where nume like 'Stoica' and Prenume like 'Ionut'
    commit;
end try
begin catch
IF ERROR_NUMBER()=547
BEGIN
    PRINT 'Nu se poate sterge angajatul, daca doriti sa fie scos din
        organizatie setati la data finalizarii contractului data
        concedierii.'
    rollback;
END
end catch
```

106. Să se creeze o tranzacție care să returneze un mesaj corespunzător dacă există acea persoană în baza de date.

```
begin tran
    select IDAngajat from Angajat
    where Nume like 'Popescu' and Prenume like 'Stefan'
    union all
    select IDSportiv from Sportivi
    where Nume like 'Popescu' and Prenume like 'Stefan'
    if @@ROWCOUNT =0
    begin
        print 'Persoana cautata nu exista in baza de date'
        rollback;
    end
    else
    begin
        print 'Persoana cautata exsita in baza de date'
        commit;
    end
end
```

107. Să se facă o tranzacție ce va insera un sport. În cazul în care în care există acest sport dar e expirat atunci se va șterge expirat din numele acestuia.

```
begin tran
insert into Sporturi(Denumire)
values ('BJJ')
if exists(select count(*) from Sporturi where Denumire like '%BJJ%')
having count(*)>1
begin
    rollback;
    begin tran
    update Sporturi
    set Denumire= SUBSTRING(Denumire,0,len(Denumire)-7)
    where Denumire like '%BJJ%' and Denumire like '%expirat%'
    commit
```

```
end  
else  
begin  
commit  
end
```

CTE

108. Să se afișeze id-ul, numele și prenumele și punctajul sportivilor care au fost la competiția Campionatul National din perioada 14-16 martie 2017, grupați în funcție de numărul de puncte acumulate (locul 1=5, locul 2=3, locul 3=2, doar participare=1)

```
with Punctaj as  
(  
    SELECT S.IDSportiv as id, s.Nume as Nume, S.Prenume as Prenume,  
    COUNT(case when P.Premiu like '%1%' then 5 end) as locul1,  
    COUNT(case when P.Premiu like '%2%' then 3 end) as locul2,  
    COUNT(case when P.Premiu like '%3%' then 2 end) as locul3,  
    COUNT(case when P.Premiu like '%participare%' then 1 end) as  
participare  
    from Sportivi AS S  
    INNER JOIN Premii AS P  
    ON P.IDSportiv=S.IDSportiv  
    INNER JOIN Evenimente AS E  
    ON E.IDEveniment=P.IDEveniment  
    INNER JOIN TipEveniment AS T  
    ON T.IDTip =E.IDTip  
    WHERE T.Denumire LIKE 'Competitie'  
    and E.Nume like 'Campionatul National'  
    and E.DataIncepere like convert(date, '2017-03-14')  
    group by s.Nume, s.Prenume, s.IDSportiv  
)  
select id, Nume, Prenume, (locul1*5+locul2*3+locul3*2+participare) as  
punctaj from Punctaj  
order by punctaj desc
```

	id	Nume	Prenume	punctaj
1	1	Popescu	Stefan	10
2	3	Stoica	Andrei	8
3	16	Negoiu	Raluca	7
4	19	Ionescu	Roberta	6
5	37	Novi	Luna	5
6	18	Manea	Robert	5
7	34	Mihalache	Cristina	4
8	22	Anolo	Mihai	3

109. Afișați numele, prenumele, sexul și salariul angajatului cu cel mai mare salariu actual. Prin Salariu se face referire la valoarea netă.

```
with c as
(select A.Nume, A.Prenume, S.IDSalariu,
case when A.CNP like '2%' then 'Feminin'
      when A.CNP like '1%' then 'Masculin'
      when A.CNP like '5%' then 'Masculin'
      when A.CNP like '6%' then 'Feminin'
      else 'Nedefinit'
end as Gen,
S.ValoareBruta *(100- S.ValoareImpozit) /100 as ValoareaNeta
from Angajat as A
inner join Salarii as S
on A.IDAngajat =S.IDAngajat
where S.LaData is null
),
a as
(select MAX(ValoareaNeta) as ValoareaNeta from c)
select c.Nume, c.Prenume,c.Gen, c.ValoareaNeta from a
inner join c
on c.ValoareaNeta=a.ValoareaNeta
```

	Nume	Prenume	Gen	ValoareaNeta
1	Stoica	Ionut	Masculin	1360
2	Princa	Cristina	Masculin	1360

110. Afișați numele și prenumele tuturor sportivilor care au luat locul 1 și cate locuri 1 a luat, dar și numărul total de locul 1,2 și 3 la competiția Campionatul National din data de 2017-03-14, in ordinea alfabetică a numelui.

```
with c as
(
SELECT S.IDSportiv as id,s.Nume as Nume, S.Prenume as Prenume,
COUNT(case when P.Premiu like '%1%' then 5 end) as locul1,
COUNT(case when P.Premiu like '%2%' then 3 end) as locul2,
COUNT(case when P.Premiu like '%3%' then 2 end) as locul3,
COUNT(case when P.Premiu like '%participare%' then 1 end) as
participare
from Sportivi AS S
INNER JOIN Premii AS P
ON P.IDSportiv=S.IDSportiv
INNER JOIN Evenimente AS E
ON E.IDEveniment=P.IDEveniment
INNER JOIN TipEveniment AS T
ON T.IDTip =E.IDTip
WHERE T.Denumire LIKE 'Competitie'
and E.Nume like 'Campionatul National'
and E.DataIncepere like convert(date,'2017-03-14')
group by s.Nume, s.Prenume, s.IDSportiv
)
select c.Nume, C.Prenume, C.Locul1, (C.locul1+C.locul2 + C.locul3) as
LocuriTotale from c
```

```
where C.locul1 >0  
order by c.Nume, c.Prenume
```

	Nume	Prenume	Locul1	LocuriTotale
1	Ionescu	Roberta	1	1
2	Negoiu	Raluca	1	2
3	Popescu	Stefan	2	2
4	Stoica	Andrei	1	2

111. Afișați id-ul, numele, prenumele și punctajul primilor 3 sportivi, ordonați descrescător după numărul de puncte adunate în 2016 (locul 1=5; locul 2=3; locul 3=2; participarea la o competiție=1)

```
with c as  
(  
    select p.IDSportiv,  
    count( case when P.Premiu like '%1%' then 1 end )as locuri1,  
    count( case when P.Premiu like '%2%' then 1 end )as locuri2,  
    count( case when P.Premiu like '%3%' then 1 end )as locuri3,  
    count( case when P.Premiu like '%participare%' then 1 end )as  
    participare  
    from Premii as P  
    inner join Evenimente as E  
    on P.IDEveniment=E.IDEveniment  
    where e.DataIncepere>'2016-01-01' and e.DataIncepere <'2016-12-  
31'  
    group by IDSportiv  
)  
select top 3 S.IDSportiv, s.Nume,S.Prenume  
,locuri1*5+locuri2*3+locuri3*2+participare as punctaj from c  
inner join Sportivi as S  
on c.IDSportiv=S.IDSportiv  
order by punctaj desc
```

112. În urma ultimei competiții la care a participat, sportivul Popescu Stefan primește un premiu special de la federație, acest lucru se va trece la observațiile acestuia legate de acea competiție.

```
with lastEvent as (  
    select top 1 Sportivi.IDSportiv,  
    Evenimente.IDEveniment,Evenimente.Nume, Evenimente.Locatie from  
    Evenimente  
    inner join TipEveniment  
    on TipEveniment.IDTip=Evenimente.IDTip  
    inner join Premii  
    on Premii.IDEveniment =Evenimente.IDEveniment  
    inner join Sportivi  
    on Sportivi.IDSportiv = Premii.IDSportiv  
    where TipEveniment.Denumire like 'Competitie' and Sportivi.Nume like  
    'Popescu' and Sportivi.Prenume like 'Stefan'  
    group by Sportivi.IDSportiv, Evenimente.IDEveniment, Evenimente.Nume,  
    Evenimente.Locatie  
    order by Evenimente.IDEveniment desc  
)
```

```
update P
set P.Observatii='Premiu special FRAM'
from Premii as P
inner join lastEvent
on P.IDEveniment=lastEvent.IDEveniment
where P.IDSportiv=lastEvent.IDSportiv
```

113. Afișați numărul de participanți de la fiecare competiție

```
with c as
(
    select IDSportiv,IDEveniment, count(*) as nrprobe from Premii
    group by IDSportiv, IDEveniment
)
select e.Nume, e.Locatie, count(*) as Nrparticipanti from c
inner join Evenimente as e
on e.IDEveniment=c.IDEveniment
inner join TipEveniment as T
on T.IDTip=E.IDTip
where t.Denumire like 'Competitie'
group by e.Nume, e.Locatie
```

	Nume	Locatie	Nrparticipanti
1	Cupa Junior	1 Mai	3
2	Cupa Brasov	Brasov	4
3	Cupa Kokoro	Brasov	5
4	Campionatul National de Box	Bucuresti	4
5	Campionatul National	Bucuresti	12
6	Campionatul National	Cluj	8
7	Cupa Romana	Cluj-Napoca	1
8	Cupa Aquila	Giurgiu	8
9	Campionatul National	Sibiu	3
10	Cupa Valcea	Valcea	8