

ROMÂNIA
MINISTERUL APĂRĂRII NAȚIONALE
ACADEMIA TEHNICĂ MILITARĂ "FERDINAND I"
FACULTATEA DE SISTEME INFORMATICE SI SECURITATE
CIBERNETICA



Galerie de artă

-Proiect Baze de Date-

Realizat de:

Std. cap. Sava Iulian Victor

Cuprins

Context	3
Diagramă	4
Creare baza de date.....	4
Organizarea tabelor	5
Select	25
Insert.....	33
Update	36
Delete	40
Trigger.....	43
View	50
Proceduri stocate.....	53
Tranzactii	58

Context

Această bază de date își propune să gestioneze fluxul de informații generat de organizarea unei galerii de artă.

Galeria de artă este formată din mai multe expoziții, acestea având o perioadă delimitată în care sunt expuse. Fiecare expoziție conține lucrările originale ale autorilor; acestea fiind unicate, nu se pot regăsi simultan în mai multe expoziții.

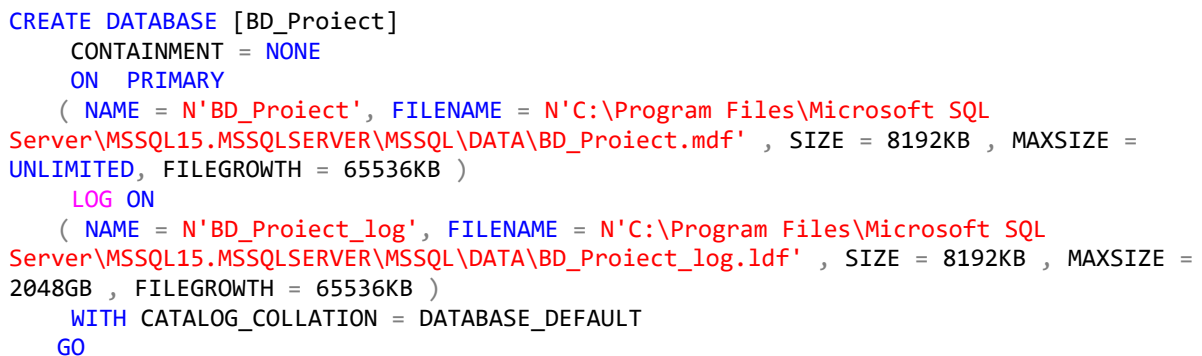
Galeria este concepută cu scopul de a pune în lumină munca artiștilor și de asemenea pentru a oferi ocazia pasionaților de a intra în posesia unei copii a exponatelor. Aceste copii sunt amplasate în multiple depozite, fiecare copie regăsindu-se în mai multe mărimi, însă având un număr limitat.

Fiecare pictură expusă are atașată informații referitoare la anul în care a fost concepută, numele pictorului, prețul acesteia și numele lucrării. De asemenea se regăsesc și informații referitoare la autori.

Pentru ca arta să fie expusă lumii, aceasta este amplasată într-o clădire având nevoie de diverse utilități pentru o funcționare corespunzătoare: apă, lumină, căldură etc. Fiecare unitate are un număr de angajați distribuiți în mai multe departamente, fiecare departament având câte un lider și mulți nuncitori cu diverse funcții.

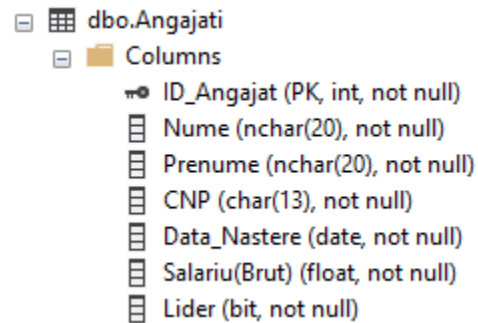
În cazul în care un vizitator dorește să cumpere o copie a unei lucrări, acesta depune o cerere în care specifică adresa (în cazul ridicării personale, aceasta nu mai este necesară), datele de contact și dimensiunea dorită, în vederea livrării.

Creare baza de date



Organizarea tabelelor

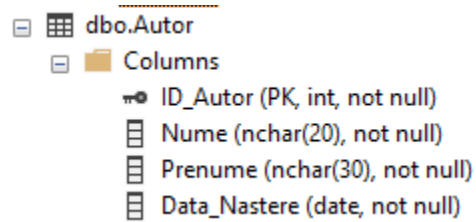
Angajati



dbo.Angajati
Columns
ID_Angajat (PK, int, not null)
Nume (nvarchar(20), not null)
Prenume (nvarchar(20), not null)
CNP (char(13), not null)
Data_Nastere (date, not null)
Salariu(Brut) (float, not null)
Lider (bit, not null)

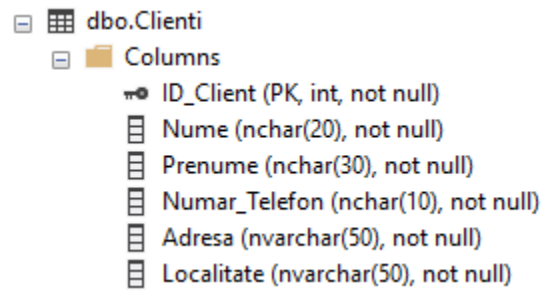
```
IF OBJECT_ID('Angajati','t') IS NOT NULL
    DROP TABLE Angajati
GO
CREATE TABLE [dbo].[Angajati](
    [ID_Angajat] [int] IDENTITY(1,1) NOT NULL,
    [Nume] [nvarchar](20) NOT NULL,
    [Prenume] [nvarchar](20) NOT NULL,
    [CNP] [char](13) NOT NULL,
    [Data_Nastere] [date] NOT NULL,
    [Salariu(Brut)] [float] NOT NULL,
    [Lider] [bit] NOT NULL,
    CONSTRAINT [PK_Angajati] PRIMARY KEY CLUSTERED
(
    [ID_Angajat] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Autor



```
IF OBJECT_ID('Autor','t') IS NOT NULL
    DROP TABLE Autor
GO
CREATE TABLE [dbo].[Autor](
    [ID_Autor] [int] IDENTITY(1,1) NOT NULL,
    [Nume] [nvarchar](20) NOT NULL,
    [Prenume] [nvarchar](30) NOT NULL,
    [Data_Nastere] [date] NOT NULL,
    CONSTRAINT [PK_Autor] PRIMARY KEY CLUSTERED
(
    [ID_Autor] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

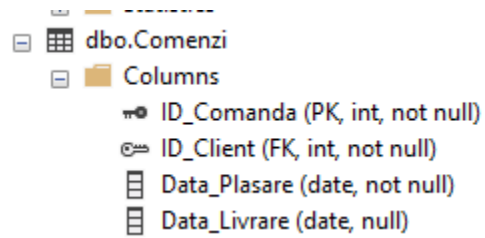
Clienti



dbo.Clienti	
Columns	
ID_Client	(PK, int, not null)
Nume	(nchar(20), not null)
Prenume	(nchar(30), not null)
Numar_Telefon	(nchar(10), not null)
Adresa	(nvarchar(50), not null)
Localitate	(nvarchar(50), not null)

```
IF OBJECT_ID('Clienti','t') IS NOT NULL
    DROP TABLE Clienti
GO
CREATE TABLE [dbo].[Clienti](
    [ID_Client] [int] IDENTITY(1,1) NOT NULL,
    [Nume] [nchar](20) NOT NULL,
    [Prenume] [nchar](30) NOT NULL,
    [Numar_Telefon] [nchar](10) NOT NULL,
    [Adresa] [nvarchar](50) NOT NULL,
    [Localitate] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Clienti] PRIMARY KEY CLUSTERED
(
    [ID_Client] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Comenzi

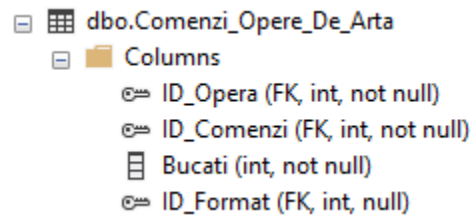


```
IF OBJECT_ID('Comenzi','t') IS NOT NULL
    DROP TABLE Comenzi
GO
CREATE TABLE [dbo].[Comenzi](
    [ID_Comanda] [int] IDENTITY(1,1) NOT NULL,
    [ID_Client] [int] NOT NULL,
    [Data_Plasare] [date] NOT NULL,
    [Data_Livrare] [date] NULL,
    CONSTRAINT [PK_Comenzi] PRIMARY KEY CLUSTERED
(
    [ID_Comanda] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Comenzi] WITH CHECK ADD CONSTRAINT [FK_Comenzi_Clienti] FOREIGN
KEY([ID_Client])
REFERENCES [dbo].[Clienti] ([ID_Client])
GO

ALTER TABLE [dbo].[Comenzi] CHECK CONSTRAINT [FK_Comenzi_Clienti]
GO
```


Comenzi_Opere_De_Arta



```
IF OBJECT_ID('Comenzi_Opere_De_Arta','t') IS NOT NULL
    DROP TABLE Comenzi_Opere_De_Arta
```

```
GO
```

```
CREATE TABLE [dbo].[Comenzi_Opere_De_Arta](
    [ID_Opera] [int] NOT NULL,
    [ID_Comenzi] [int] NOT NULL,
    [Bucati] [int] NOT NULL,
    [ID_Format] [int] NULL
) ON [PRIMARY]
```

```
GO
```

```
ALTER TABLE [dbo].[Comenzi_Opere_De_Arta] WITH CHECK ADD CONSTRAINT
[FK_Comenzi_Opere_De_Arta_Comenzi] FOREIGN KEY([ID_Comenzi])
REFERENCES [dbo].[Comenzi] ([ID_Comanda])
ON DELETE CASCADE
GO
```

```
ALTER TABLE [dbo].[Comenzi_Opere_De_Arta] CHECK CONSTRAINT
[FK_Comenzi_Opere_De_Arta_Comenzi]
GO
```

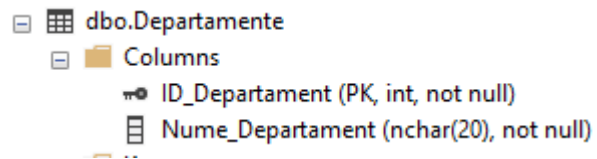
```
ALTER TABLE [dbo].[Comenzi_Opere_De_Arta] WITH CHECK ADD CONSTRAINT
[FK_Comenzi_Opere_De_Arta_Format] FOREIGN KEY([ID_Format])
REFERENCES [dbo].[Format] ([ID_Format])
ON DELETE CASCADE
GO
```

```
ALTER TABLE [dbo].[Comenzi_Opere_De_Arta] CHECK CONSTRAINT
[FK_Comenzi_Opere_De_Arta_Format]
GO
```

```
ALTER TABLE [dbo].[Comenzi_Opere_De_Arta] WITH CHECK ADD CONSTRAINT
[FK_Comenzi_Opere_De_Arta_Opere_De_Arta] FOREIGN KEY([ID_Opera])
REFERENCES [dbo].[Opere_De_Arta] ([ID_Opera])
ON DELETE CASCADE
GO
```

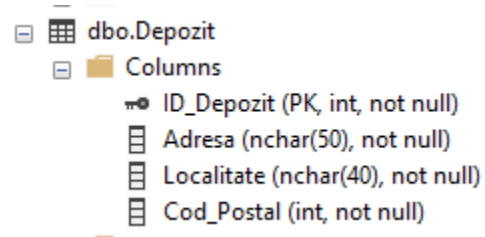
```
ALTER TABLE [dbo].[Comenzi_Opere_De_Arta] CHECK CONSTRAINT
[FK_Comenzi_Opere_De_Arta_Opere_De_Arta]
GO
```

Departamente



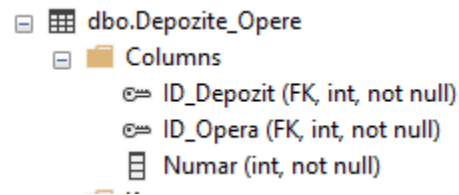
```
IF OBJECT_ID('Departamente','t') IS NOT NULL
    DROP TABLE Departamente
GO
CREATE TABLE [dbo].[Departamente](
    [ID_Departament] [int] IDENTITY(1,1) NOT NULL,
    [Nume_Departament] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Departamente] PRIMARY KEY CLUSTERED
(
    [ID_Departament] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Depozit



```
IF OBJECT_ID('Depozit','t') IS NOT NULL
    DROP TABLE Depozit
GO
CREATE TABLE [dbo].[Depozit](
    [ID_Depozit] [int] IDENTITY(1,1) NOT NULL,
    [Adresa] [nvarchar](50) NOT NULL,
    [Localitate] [nvarchar](40) NOT NULL,
    [Cod_Postal] [int] NOT NULL,
    CONSTRAINT [PK_Depozit] PRIMARY KEY CLUSTERED
(
    [ID_Depozit] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Depozit_Opere



```
IF OBJECT_ID('Depozite_Opere','t') IS NOT NULL
    DROP TABLE Depozite_Opere
GO
CREATE TABLE [dbo].[Depozite_Opere](
    [ID_Depozit] [int] NOT NULL,
    [ID_Opera] [int] NOT NULL,
    [Numar] [int] NOT NULL,
    CONSTRAINT [UQ_ID_Opera] UNIQUE NONCLUSTERED
(
    [ID_Opera] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

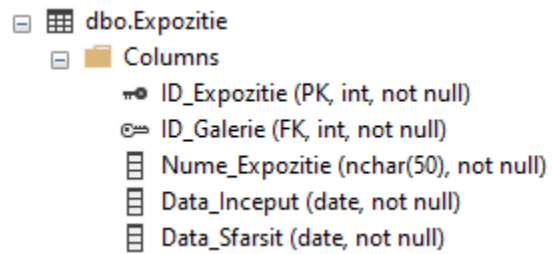
ALTER TABLE [dbo].[Depozite_Opere] WITH CHECK ADD CONSTRAINT
[FK_Depozite_Opere_Depozit] FOREIGN KEY([ID_Depozit])
REFERENCES [dbo].[Depozit] ([ID_Depozit])
GO

ALTER TABLE [dbo].[Depozite_Opere] CHECK CONSTRAINT [FK_Depozite_Opere_Depozit]
GO

ALTER TABLE [dbo].[Depozite_Opere] WITH CHECK ADD CONSTRAINT
[FK_Depozite_Opere_Opere_De_Arta] FOREIGN KEY([ID_Opera])
REFERENCES [dbo].[Operare_De_Arta] ([ID_Opera])
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[Depozite_Opere] CHECK CONSTRAINT [FK_Depozite_Opere_Opere_De_Arta]
GO
```

Expozitie



```
IF OBJECT_ID('Expozitie','t') IS NOT NULL
    DROP TABLE Expozitie
GO
CREATE TABLE [dbo].[Expozitie](
    [ID_Expozitie] [int] IDENTITY(1,1) NOT NULL,
    [ID_Galerie] [int] NOT NULL,
    [Nume_Expozitie] [nvarchar](50) NOT NULL,
    [Data_Inceput] [date] NOT NULL,
    [Data_Sfarsit] [date] NOT NULL,
    CONSTRAINT [PK_Expozitie] PRIMARY KEY CLUSTERED
(
    [ID_Expozitie] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Expozitie] WITH CHECK ADD CONSTRAINT [FK_Expozitie_Galerii] FOREIGN
KEY([ID_Galerie])
REFERENCES [dbo].[Galerii] ([ID_Galerie])
GO

ALTER TABLE [dbo].[Expozitie] CHECK CONSTRAINT [FK_Expozitie_Galerii]
GO
```

Expozitii_Opere_De_Arta



```
IF OBJECT_ID('Expozitii_Opere_De_Arta','t') IS NOT NULL
    DROP TABLE Expozitii_Opere_De_Arta
GO
CREATE TABLE [dbo].[Expozitii_Opere_De_Arta](
    [ID_Expozitie] [int] NOT NULL,
    [ID_Opera] [int] NOT NULL,
    CONSTRAINT [UQ_ID_Opera_] UNIQUE NONCLUSTERED
(
    [ID_Opera] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

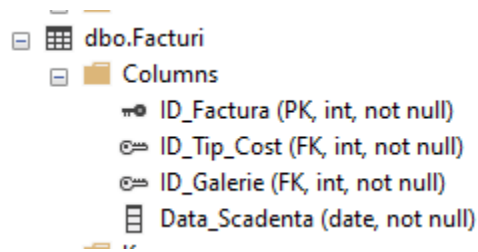
ALTER TABLE [dbo].[Expozitii_Opere_De_Arta] WITH CHECK ADD CONSTRAINT
[FK_Expozitii_Opere_De_Arta_Expozitie] FOREIGN KEY([ID_Expozitie])
REFERENCES [dbo].[Expozitie] ([ID_Expozitie])
GO

ALTER TABLE [dbo].[Expozitii_Opere_De_Arta] CHECK CONSTRAINT
[FK_Expozitii_Opere_De_Arta_Expozitie]
GO

ALTER TABLE [dbo].[Expozitii_Opere_De_Arta] WITH CHECK ADD CONSTRAINT
[FK_Expozitii_Opere_De_Arta_Opere_De_Arta] FOREIGN KEY([ID_Opera])
REFERENCES [dbo].[Opere_De_Arta] ([ID_Opera])
ON UPDATE CASCADE
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[Expozitii_Opere_De_Arta] CHECK CONSTRAINT
[FK_Expozitii_Opere_De_Arta_Opere_De_Arta]
GO
```

Facturi



```
IF OBJECT_ID('Facturi','t') IS NOT NULL
    DROP TABLE Facturi
GO
CREATE TABLE [dbo].[Facturi](
    [ID_Factura] [int] IDENTITY(1,1) NOT NULL,
    [ID_Tip_Cost] [int] NOT NULL,
    [ID_Galerie] [int] NOT NULL,
    [Data_Scadenta] [date] NOT NULL,
    CONSTRAINT [PK_Facturi] PRIMARY KEY CLUSTERED
(
    [ID_Factura] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

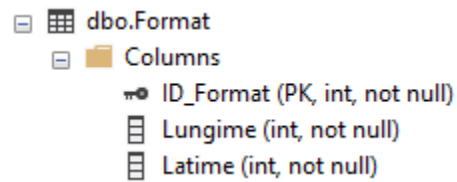
ALTER TABLE [dbo].[Facturi] WITH CHECK ADD CONSTRAINT [FK_Facturi_Galerii] FOREIGN
KEY([ID_Galerie])
REFERENCES [dbo].[Galerii] ([ID_Galerie])
ON UPDATE CASCADE
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[Facturi] CHECK CONSTRAINT [FK_Facturi_Galerii]
GO

ALTER TABLE [dbo].[Facturi] WITH CHECK ADD CONSTRAINT [FK_Facturi_Tip_Cost] FOREIGN
KEY([ID_Tip_Cost])
REFERENCES [dbo].[Tip_Cost] ([ID_Tip_Cost])
GO

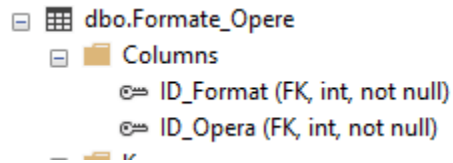
ALTER TABLE [dbo].[Facturi] CHECK CONSTRAINT [FK_Facturi_Tip_Cost]
GO
```

Format



```
IF OBJECT_ID('Format','t') IS NOT NULL
    DROP TABLE Format
GO
CREATE TABLE [dbo].[Format](
    [ID_Format] [int] IDENTITY(1,1) NOT NULL,
    [Lungime] [int] NOT NULL,
    [Latime] [int] NOT NULL,
    CONSTRAINT [PK_Format] PRIMARY KEY CLUSTERED
(
    [ID_Format] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```


Formate_Opere



```
IF OBJECT_ID('Formate_Opere','t') IS NOT NULL
    DROP TABLE Formate_Opere
```

GO

```
CREATE TABLE [dbo].[Formate_Opere](
    [ID_Format] [int] NOT NULL,
    [ID_Opera] [int] NOT NULL
) ON [PRIMARY]
```

GO

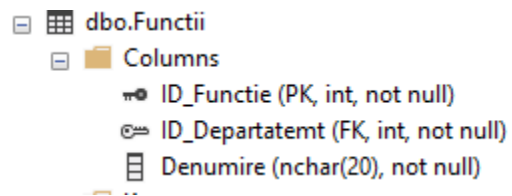
```
ALTER TABLE [dbo].[Formate_Opere] WITH CHECK ADD CONSTRAINT [FK_Formate_Opere_Format]
FOREIGN KEY([ID_Format])
REFERENCES [dbo].[Format] ([ID_Format])
ON DELETE CASCADE
GO
```

```
ALTER TABLE [dbo].[Formate_Opere] CHECK CONSTRAINT [FK_Formate_Opere_Format]
GO
```

```
ALTER TABLE [dbo].[Formate_Opere] WITH CHECK ADD CONSTRAINT
[FK_Formate_Opere_Opere_De_Arta] FOREIGN KEY([ID_Opera])
REFERENCES [dbo].[Opere_De_Arta] ([ID_Opera])
ON DELETE CASCADE
GO
```

```
ALTER TABLE [dbo].[Formate_Opere] CHECK CONSTRAINT [FK_Formate_Opere_Opere_De_Arta]
GO
```

Funcții

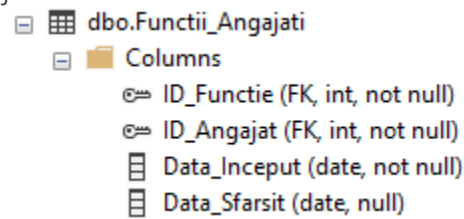


```
IF OBJECT_ID('Funcții', 't') IS NOT NULL
    DROP TABLE Funcții
GO
CREATE TABLE [dbo].[Funcții](
    [ID_Funcție] [int] IDENTITY(1,1) NOT NULL,
    [ID_Departament] [int] NOT NULL,
    [Denumire] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Funcții] PRIMARY KEY CLUSTERED
(
    [ID_Funcție] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Funcții] WITH CHECK ADD CONSTRAINT [FK_Funcții_Departamente] FOREIGN
KEY([ID_Departament])
REFERENCES [dbo].[Departamente] ([ID_Departament])
GO

ALTER TABLE [dbo].[Funcții] CHECK CONSTRAINT [FK_Funcții_Departamente]
GO
```

Funcții_Angajati



```
IF OBJECT_ID('Funcții_Angajati','t') IS NOT NULL
    DROP TABLE Funcții_Angajati
```

```
GO
```

```
CREATE TABLE [dbo].[Funcții_Angajati](
    [ID_Funcție] [int] NOT NULL,
    [ID_Angajat] [int] NOT NULL,
    [Data_Inceput] [date] NOT NULL,
    [Data_Sfarsit] [date] NULL
) ON [PRIMARY]
```

```
GO
```

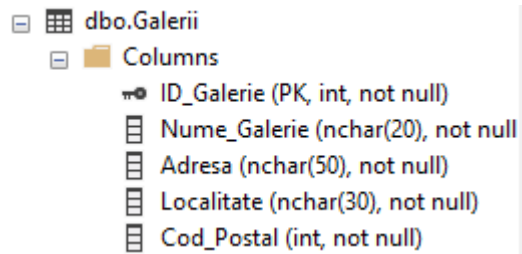
```
ALTER TABLE [dbo].[Funcții_Angajati] WITH CHECK ADD CONSTRAINT
[FK_Funcții_Angajati_Angajati] FOREIGN KEY([ID_Angajat])
REFERENCES [dbo].[Angajati] ([ID_Angajat])
GO
```

```
ALTER TABLE [dbo].[Funcții_Angajati] CHECK CONSTRAINT [FK_Funcții_Angajati_Angajati]
GO
```

```
ALTER TABLE [dbo].[Funcții_Angajati] WITH CHECK ADD CONSTRAINT
[FK_Funcții_Angajati_Funcții] FOREIGN KEY([ID_Funcție])
REFERENCES [dbo].[Funcții] ([ID_Funcție])
ON DELETE CASCADE
GO
```

```
ALTER TABLE [dbo].[Funcții_Angajati] CHECK CONSTRAINT [FK_Funcții_Angajati_Funcții]
GO
```

Galerii



The screenshot shows the 'Columns' folder expanded for the 'dbo.Galerii' table. It lists five columns: ID_Galerie (PK, int, not null), Nume_Galerie (nvarchar(20), not null), Adresa (nvarchar(50), not null), Localitate (nvarchar(30), not null), and Cod_Postal (int, not null).

dbo.Galerii	
Columns	
ID_Galerie	(PK, int, not null)
Nume_Galerie	(nvarchar(20), not null)
Adresa	(nvarchar(50), not null)
Localitate	(nvarchar(30), not null)
Cod_Postal	(int, not null)

```
IF OBJECT_ID('Galerii','t') IS NOT NULL
    DROP TABLE Galerii
GO
CREATE TABLE [dbo].[Galerii](
    [ID_Galerie] [int] IDENTITY(1,1) NOT NULL,
    [Nume_Galerie] [nvarchar](20) NOT NULL,
    [Adresa] [nvarchar](50) NOT NULL,
    [Localitate] [nvarchar](30) NOT NULL,
    [Cod_Postal] [int] NOT NULL,
    CONSTRAINT [PK_Galerii] PRIMARY KEY CLUSTERED
(
    [ID_Galerie] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Galerii_Departamente



dbo.Galerii_Departamente
Columns
ID_Departament (FK, int, not null)
ID_Galerie (FK, int, not null)

```
IF OBJECT_ID('Galerii_Departamente','t') IS NOT NULL
    DROP TABLE Galerii_Departamente
```

GO

```
CREATE TABLE [dbo].[Galerii_Departamente](
    [ID_Departament] [int] NOT NULL,
    [ID_Galerie] [int] NOT NULL
) ON [PRIMARY]
```

GO

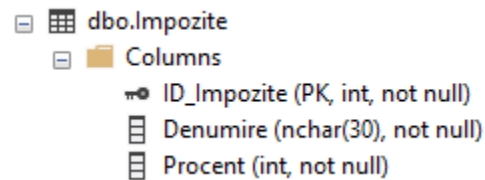
```
ALTER TABLE [dbo].[Galerii_Departamente] WITH CHECK ADD CONSTRAINT
[FK_Galerii_Departamente_Departamente] FOREIGN KEY([ID_Departament])
REFERENCES [dbo].[Departamente] ([ID_Departament])
GO
```

```
ALTER TABLE [dbo].[Galerii_Departamente] CHECK CONSTRAINT
[FK_Galerii_Departamente_Departamente]
GO
```

```
ALTER TABLE [dbo].[Galerii_Departamente] WITH CHECK ADD CONSTRAINT
[FK_Galerii_Departamente_Galerii] FOREIGN KEY([ID_Galerie])
REFERENCES [dbo].[Galerii] ([ID_Galerie])
GO
```

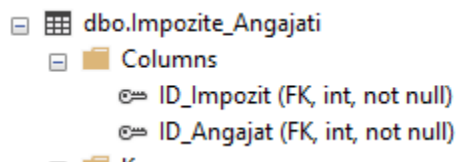
```
ALTER TABLE [dbo].[Galerii_Departamente] CHECK CONSTRAINT
[FK_Galerii_Departamente_Galerii]
GO
```

Impozite



```
IF OBJECT_ID('Impozite','t') IS NOT NULL
    DROP TABLE Impozite
GO
CREATE TABLE [dbo].[Impozite](
    [ID_Impozite] [int] IDENTITY(1,1) NOT NULL,
    [Denumire] [nvarchar](30) NOT NULL,
    [Procent] [int] NOT NULL,
    CONSTRAINT [PK_Impozite] PRIMARY KEY CLUSTERED
(
    [ID_Impozite] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Impozite_Angajati



```
IF OBJECT_ID('Impozite_Angajati','t') IS NOT NULL
    DROP TABLE Impozite_Angajati
GO
CREATE TABLE [dbo].[Impozite_Angajati](
    [ID_Impozit] [int] NOT NULL,
    [ID_Angajat] [int] NOT NULL
) ON [PRIMARY]
GO

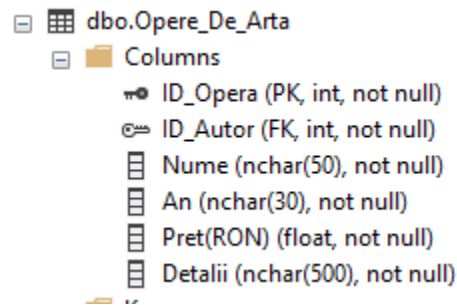
ALTER TABLE [dbo].[Impozite_Angajati] WITH CHECK ADD CONSTRAINT
[FK_Impozite_Angajati_Angajati] FOREIGN KEY([ID_Angajat])
REFERENCES [dbo].[Angajati] ([ID_Angajat])
GO

ALTER TABLE [dbo].[Impozite_Angajati] CHECK CONSTRAINT [FK_Impozite_Angajati_Angajati]
GO

ALTER TABLE [dbo].[Impozite_Angajati] WITH CHECK ADD CONSTRAINT
[FK_Impozite_Angajati_Impozite] FOREIGN KEY([ID_Impozit])
REFERENCES [dbo].[Impozite] ([ID_Impozite])
GO

ALTER TABLE [dbo].[Impozite_Angajati] CHECK CONSTRAINT [FK_Impozite_Angajati_Impozite]
GO
```

Opere_De_Arta



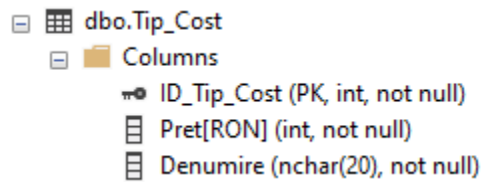
dbo.Opere_De_Arta	
Columns	
ID_Opera	(PK, int, not null)
ID_Autor	(FK, int, not null)
Nume	(nvarchar(50), not null)
An	(nvarchar(30), not null)
Pret(RON)	(float, not null)
Detalii	(nvarchar(500), not null)

```
IF OBJECT_ID('Opere_De_Arta','t') IS NOT NULL
    DROP TABLE Opere_De_Arta
GO
CREATE TABLE [dbo].[Opere_De_Arta](
    [ID_Opera] [int] IDENTITY(1,1) NOT NULL,
    [ID_Autor] [int] NOT NULL,
    [Nume] [nvarchar](50) NOT NULL,
    [An] [nvarchar](30) NOT NULL,
    [Pret(RON)] [float] NOT NULL,
    [Detalii] [nvarchar](500) NOT NULL,
    CONSTRAINT [PK_Opere_De_Arta] PRIMARY KEY CLUSTERED
(
    [ID_Opera] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Opere_De_Arta] WITH CHECK ADD CONSTRAINT [FK_Opere_De_Arta_Autor]
FOREIGN KEY([ID_Autor])
REFERENCES [dbo].[Autor] ([ID_Autor])
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[Opere_De_Arta] CHECK CONSTRAINT [FK_Opere_De_Arta_Autor]
GO
```

Tip_Cost



dbo.Tip_Cost
Columns
ID_Tip_Cost (PK, int, not null)
Pret[RON] (int, not null)
Denumire (nvarchar(20), not null)

```
IF OBJECT_ID('Tip_Cost','t') IS NOT NULL
    DROP TABLE Tip_Cost
```

GO

```
CREATE TABLE [dbo].[Tip_Cost](
    [ID_Tip_Cost] [int] IDENTITY(1,1) NOT NULL,
    [Pret[RON]] [int] NOT NULL,
    [Denumire] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Tip_Cost] PRIMARY KEY CLUSTERED
(
    [ID_Tip_Cost] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```


Select

--1) Afiseaza operele de arta si autorii acestora

```
select O.Nume, A.Prenume+ ' '+A.Nume as [Nume Autor]
from Opere_De_Arta as O
     inner join Autor as A
     on A.ID_Autor=O.ID_Autor
```

	Nume	Nume Autor	
1	Liberty Leading the People	Eugène	Delacroix
2	The Raft of the Medusa	Théodore	Géricault
3	The Coronation of Napoleon	David	Jacques-Louis
4	Death of the Virgin	Caravaggio	Michelangelo Merisi
5	Oath of the Horatii	David	Jacques-Louis
6	The Bolt	Fragonard	Jean-Honoré
7	The Death of Sardanapalus	Eugène	Delacroix
8	Man with a Glove	Tiziano	Vecelli
9	The Barque of Dante	Eugène	Delacroix
10	The Intervention of the Sabine Women	David	Jacques-Louis
11	Portrait of Madame Récamier	David	Jacques-Louis
12	Greece on the Ruins of Missolonghi	Eugène	Delacroix
13	The Sultan of Morocco	Eugène	Delacroix
14	Jacob Wrestling with the Angel	Eugène	Delacroix
15	Orphan Girl at the Cemetery	Eugène	Delacroix
16	Portrait of Frédéric Chopin and George Sand	Eugène	Delacroix
17	The Charging Chasseur	Théodore	Géricault
18	The Wounded Cuirassier	Théodore	Géricault
19	Portrait of a Kleptomaniac	Théodore	Géricault

--2) Afiseaza operele de arta apartinand autorului cu ID 6

```
select O.Nume
from Opere_De_Arta as O
where O.ID_Autor='6'
```

	Nume
1	Death of the Virgin

--3) Afiseaza operele de arta ce au copii cu formatul cu id 3

```
select O.Nume
from Opere_De_Arta as O
     join Formate_Opere as FO
     on O.ID_Opera=FO.ID_Opera
     join Format as F
     on F.ID_Format=FO.ID_Format
where F.ID_Format='3'
```

	Nume
1	Greece on the Ruins of Missolonghi
2	The Raft of the Medusa

--4) Afiseaza operele de arta din depozitul cu ID 1

```
select O.Nume
from Opere_De_Arta as O
     join Depozite_Opere as DO
```

```

on O.ID_Opera=D0.ID_Opera
join Depozit as D
on D.ID_Depozit=D0.ID_Depozit
where D.ID_Depozit='1'

```

	Nume
1	The Raft of the Medusa
2	The Barque of Dante
3	The Sultan of Morocco
4	Portrait of Frédéric Chopin and George Sand
5	The Tennis Court Oath

--5) Afiseaza operele de arta din depozitul cu ID 1 ale autorilor nascuti dupa 1700

```

with
Opere as
(
    select O.Nume, O.ID_Opera
    from Opere_De_Arta as O
    inner join Autor as A
    on O.ID_Autor=A.ID_Autor
    where YEAR(A.Data_Nastere)>'1700'
)
select O.Nume
from Opere as O
inner join Depozite_Opere as D0
on O.ID_Opera=D0.ID_Opera
inner join Depozit as D
on D.ID_Depozit=D0.ID_Depozit
where D.ID_Depozit='1'

```

	Nume
1	The Raft of the Medusa
2	The Barque of Dante
3	The Sultan of Morocco
4	Portrait of Frédéric Chopin and George Sand
5	The Tennis Court Oath

--6) Afiseaza operele de arta care costa mai mult de 8000 Ron, dar mai putin de 10000 Ron

```

select O.Nume,O.[Pret(Ron)]
from Opere_De_Arta as O
where O.[Pret(Ron)]>'8000' AND O.[Pret(Ron)]<'10000'

```

	Nume	Pret(Ron)
1	The Sultan of Morocco	8363
2	Orphan Girl at the Cemetery	8715
3	The Charging Chasseur	8365

--7) Afiseaza autorii care au mai mult de 6 opere

```

with
Nume_Autori (Nume, ID)as
(
    select A.Prenume+' '+A.Nume as Nume, A.ID_Autor
    from Autor as A
)

```

```

select Nume_Autori.Nume, COUNT(*) as [Numar Opere]
from Autor as A
    inner join Nume_Autori
    on Nume_Autori.ID=A.ID_Autor
    inner join Opere_De_Arta as O
    on O.ID_Autor=A.ID_Autor
group by A.ID_Autor, Nume_Autori.Nume
having COUNT(*)>6

```

	Nume	Numar Opere
1	Eugène Delacroix	8

--8) Afiseaza aria formatelor operei de arta cu ID 11

```

select (F.Latime*F.Lungime) as [Arie(cm)]
from Format as F
    inner join Formate_Opere as FO
    on FO.ID_Format=F.ID_Format
    inner join Opere_De_Arta as O
    on O.ID_Opera=FO.ID_Opera

```

	Arie(cm)
1	90000
2	2500
3	900
4	1600
5	2500
6	900
7	5000
8	40000

--9) Afiseaza operele de arta care au formatul cu ID 3

```

select O.Nume
from Opere_De_Arta as O
    inner join Formate_Opere as FO
    on FO.ID_Opera=O.ID_Opera
    inner join Format as F
    on F.ID_Format=FO.ID_Format
where F.ID_Format='3'

```

	Nume
1	Greece on the Ruins of Missolonghi
2	The Raft of the Medusa

--10) Afiseaza operele de arta din galeria de arta cu id 1

```

select O.Nume
from Opere_De_Arta as O
    inner join Expozitii_Opere_De_Arta as EO
    on EO.ID_Opera=O.ID_Opera
    inner join Expozitie as E
    on E.ID_Expozitie=EO.ID_Expozitie
    inner join Galerii as G
    on G.ID_Galerie=E.ID_Galerie
where G.ID_Galerie='1'

```

	Nume
1	Orphan Girl at the Cemetery
2	Portrait of Frédéric Chopin and George Sand

--11) Afiseaza opera de arta care se regaseste pe cele mai multe comenzi

```
select MAX(O.Nume)
from Opere_De_Arta as O
      inner join Comenzi_Opere_De_Arta as CO
on O.ID_Opera=CO.ID_Opera
```

	(No column name)
1	The Coronation of Napoleon

--12) Afiseaza operele de arta ordonate dupa pretul acestora

```
select Nume, [Pret(Ron)]
from Opere_De_Arta
ORDER BY [Pret(Ron)] DESC
```

	Nume	Pret(Ron)
1	Portrait of Madame Récamier	10000
2	Orphan Girl at the Cemetery	8715
3	The Charging Chasseur	8365
4	The Sultan of Morocco	8363
5	Portrait of a Kleptomaniac	7652
6	The Intervention of the Sabine Women	7490
7	Vitruvian Man	7300
8	Oath of the Horatii	7000
9	The Death of Socrates	6757
10	Jacob Wrestling with the Angel	6548
11	The Coronation of Napoleon	6000
12	Mona Lisa	6000
13	The Wounded Cuirassier	5634
14	The Death of Sardanapalus	5000
15	The Barque of Dante	4970
16	The Woman with a Gambling Mania	4657
17	The Bolt	4600
18	Greece on the Ruins of Missolonghi	4363
19	The Raft of the Medusa	4000

--13) Afiseaza autorii care au trait in secolul XVIII

```
select concat(Prenume, ' ', Nume) as Nume, concat(YEAR(Data_Nastere), '(XVIII)') as
SECOL
from Autor
where YEAR(DATA_NasTERE)>1700 AND YEAR(Data_Nastere)<1799
```

	Nume		SECOL
1	Eugène	Delacroix	1798(XVIII)
2	Théodore	Géricault	1791(XVIII)
3	David	Jacques-Louis	1794(XVIII)
4	Fragonard	Jean-Honoré	1732(XVIII)
5	Antoine-Jean	Gros	1771(XVIII)

--14) Afiseaza autorii al caror opere depaseste 7000 Ron

```
select distinct A.Prenume+' '+A.Nume as Nume
from Autor as A
      inner join Opere_De_Arta as O
      on O.ID_Autor=A.ID_Autor
where O.[Pret(RON)]>7000
```

	Nume	
1	David	Jacques-Louis
2	Eugène	Delacroix
3	Leonardo	da Vinci
4	Théodore	Géricault

--15) Afiseaza autorii al caror opere se regasesc in depozitul cu ID 1

```
select distinct A.Prenume+' '+A.Nume as Nume
from Autor as A
      inner join Opere_De_Arta as O
      on O.ID_Autor=A.ID_Autor
      inner join Depozite_Opere as DO
      on DO.ID_Opera=O.ID_Opera
      inner join Depozit as D
      on D.ID_Depozit=DO.ID_Depozit
where D.ID_Depozit='1'
```

	Nume	
1	David	Jacques-Louis
2	Eugène	Delacroix
3	Théodore	Géricault

--16) Afiseaza formatele ale caror arie nu depaseste un metru patrat

```
select F.Latime, F.Lungime, (F.Latime*F.Lungime) as Arie
from Format as F
where F.Latime*F.Lungime>10000
```

	Latime	Lungime	Arie
1	300	300	90000
2	100	400	40000

--17) Afiseaza formatul care se regaseste pe cele mai multe opere de arta

```
with
Dim as
(
      select concat(F.Lungime,'*',F.Latime) as Dimensiune, F.ID_Format as ID
      from Format as F
)
select top(1) D.Dimensiune, count(*) as [Numar Opere]
from Dim as D
```

```

        inner join Formate_Opere as FO
        on FO.ID_Format=D.ID
        inner join Opere_De_Arta as O
        on O.ID_Opera=FO.ID_Opera
    group by D.Dimensiune
    order by count(*) desc

```

	Dimensiune	Numar Opere
1	30*30	2

--18) Afiseaza expozitiile din galeria cu ID 1

```

select E.Nume_Expozitie
from Expozitie as E
    inner join Galerii as G
    on E.ID_Expozitie=G.ID_Galerie
where G.ID_Galerie='1'

```

	Nume_Expozitie
1	Flori de toamna

--19) Afiseaza expozitiile care au loc intre septembrie 2020 si ianuarie 2021

```

select E.Nume_Expozitie
from Expozitie as E
where E.Data_Inceput>'20200109' and E.Data_Sfarsit<'20211201'

```

	Nume_Expozitie
1	Flori de toamna
2	Romantism
3	Marile minuni

--20) Afiseaza numarul de opere din fiecare expozitie

```

select E.Nume_Expozitie, count(*) as [Numar Opere]
from Expozitie as E
    inner join Expozitii_Opere_De_Arta as EO
    on E.ID_Expozitie=EO.ID_Expozitie
group by E.Nume_Expozitie

```

	Nume_Expozitie	Numar Opere
1	Flori de toamna	2
2	Romantism	1

--21) Afiseaza numarul total de bucati de pe fiecare comanda

```

select Cl.Nume, sum(CO.Bucati)
from Comenzi as C
    join Comenzi_Opere_De_Arta as CO
    on CO.ID_Comenzi=C.ID_Comanda
    join Clienti as Cl
    on Cl.ID_Client=C.ID_Client
group by Cl.Nume

```

	Nume	(No column name)
1	Georgescu	4
2	Grasu	2
3	Popescu	8

--22) Afiseaza comenzile care inca nu au fost livrate

```

select *
from Comenzi as C

```

where C.Data_Livrare is null

	ID_Comanda	ID_Client	Data_Plasare	Data_Livrare
1	5	5	2021-07-13	NULL
2	6	1	2021-05-12	NULL

--23) Afiseaza comenzile ale caror valoare depaseste 10000 Ron

```
select C.ID_Comanda, sum(O.[Pret(RO)]) as [Pret Comanda]
from Comenzi as C
      join Comenzi_Opere_De_Arta as CO
      on C.ID_Comanda=CO.ID_Comenzi
      join Opere_De_Arta as O
      on O.ID_Opera=CO.ID_Opera
group by C.ID_Comanda
having sum(O.[Pret(RO)])>10000
```

	ID_Comanda	Pret Comanda
1	4	20000

--24) Afiseaza comenzile unde se regasesc opere ale autorului cu ID 2

```
select C.ID_Comanda, A.Prenume+' '+A.Nume as Nume
from Comenzi as C
      join Comenzi_Opere_De_Arta as CO
      on C.ID_Comanda=CO.ID_Comenzi
      join Opere_De_Arta as O
      on O.ID_Opera=CO.ID_Opera
      join Autor as A
      on A.ID_Autor=O.ID_Autor
where A.ID_Autor='2'
```

	ID_Comanda	Nume
1	1	Eugène Delacroix

--25) Afiseaza clientii din localitatea Bucuresti

```
select *
from Clienti as C
where C.Localitate='Bucuresti'
```

	ID_Client	Nume	Prenume	Numar_Telefon	Adresa	Localitate
1	1	Popescu	Vlad	0761432765	Strada izvorul Inalr,nr.8	Bucuresti

--26) Afiseaza clientii care au facut cel putin o comanda

```
select count(*) as [Numar Comenzi],Cl.Nume
from Clienti as Cl
      join Comenzi as C
      on Cl.ID_Client=C.ID_Comanda
group by Cl.Nume
```

	Numar Comenzi	Nume
1	1	Berbecaru
2	1	Georgescu
3	1	Margaritaru
4	1	Popescu

--27) Afiseaza operele care se cumpara cel mai des din fiecare expozitie

```
with
OpereExpozitii as
(
      select O.Nume, E.Nume_Expozitie, O.ID_Opera as ID
```

```

from Opere_De_Arta as O
    join Expozitii_Opere_De_Arta as EO
    on EO.ID_Opera=O.ID_Opera
    join Expozitie as E
    on E.ID_Expozitie=EO.ID_Expozitie
),
ComenziOpere as
(
    select OE.Nume,OE.Nume_Expozitie, count(OE.Nume) as [Numar Aparitii]
    from OpereExpozitii as OE
        join Comenzi_Opere_De_Arta as CO
        on OE.ID=CO.ID_Opera
    group by OE.Nume,OE.Nume_Expozitie
)
select max(CO.Nume),CO.Nume_Expozitie,max(CO.[Numar Aparitii])
from ComenziOpere as CO
group by CO.Nume_Expozitie

```

Results Messages			
	(No column name)	Nume_Expozitie	(No column name)
1	The Bolt	Romantism	1

```

--28) Afiseaza numarul de opere din fiecare expozitie
select E.Nume_Expozitie,count(E.Nume_Expozitie) as [Numar Opere]
from Expozitie as E
    join Expozitii_Opere_De_Arta as EO
    on E.ID_Expozitie=EO.ID_Expozitie
group by E.Nume_Expozitie

```

Results Messages		
	Nume_Expozitie	Numar Opere
1	Flori de toamna	2
2	Romantism	1

Insert

--1.

```
INSERT INTO Angajati  
VALUES('Popescu', 'Adrian', '1543022335373', '19870807', '7300', '0')
```

(1 row affected)

Completion time: 2021-05-13T23:36:22.1391393+03:00

--2.

```
INSERT INTO Angajati  
VALUES('Antonescu', 'Cosmin', '15743367022', '19990412', '6000', '1')
```

(1 row affected)

Completion time: 2021-05-13T23:36:13.4599968+03:00

--3.

```
INSERT INTO Facturi  
VALUES('2', '3', '20210406')
```

Msg 50000, Level 16, State 1, Procedure FacturaInvalida, Line 14 [Batch Start Line 14]
Data inregistrata nu este valida.

Msg 3609, Level 16, State 1, Line 16

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

Completion time: 2021-05-13T23:35:38.3013432+03:00

--4.

```
INSERT INTO Autor  
VALUES('Ion', 'Andreescu', '18500215')
```

(1 row affected)

Completion time: 2021-05-13T23:29:47.9768879+03:00

--5.

```
INSERT INTO Clienti  
VALUES('Georgescu', 'Alin', '0773257944', 'Strada Principala nr. 7', 'Braila')
```

(1 row affected)

Completion time: 2021-05-13T23:29:39.0212000+03:00

--6.

```
INSERT INTO Comenzi  
VALUES('4', '20200406', '20210101')
```

(1 row affected)

Completion time: 2021-05-13T23:29:30.2561887+03:00

--7.

```
INSERT INTO Depozit
VALUES('Strada Ion Lungu nr 67','Arges','401832')
```

(1 row affected)

Completion time: 2021-05-13T23:29:22.4126007+03:00

--8.

```
INSERT INTO Format
VALUES('30','80')
```

(1 row affected)

Completion time: 2021-05-13T23:29:11.8018746+03:00

--9.

```
INSERT INTO Galerii
VALUES('Ucigasul de Lupi','Strada Marinescu al 3-lea','Bucuresti','543754')
```

(1 row affected)

Completion time: 2021-05-13T23:29:02.0808960+03:00

--10.

```
INSERT INTO Functii
VALUES('4','Secretar')
```

Msg 50000, Level 16, State 1, Procedure LiderMultiplu, Line 19 [Batch Start Line 49]:
Deja exista aceasta functie in departamentul cu ID 4
Msg 3609, Level 16, State 1, Line 51
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2021-05-13T23:28:47.4416917+03:00

--11.

```
INSERT INTO Clienti
VALUES('Margaritaru','Andrei','0732098111','Strada Milosteniei nr.
10','Botosani')
```

(1 row affected)

Completion time: 2021-05-13T23:28:38.0079575+03:00

--12.

```
INSERT INTO Impozite  
VALUES('Educatie','3')
```

(1 row affected)

Completion time: 2021-05-13T23:28:12.3170889+03:00

--13.

```
INSERT INTO Functii_Angajati  
VALUES('9','15','20200303',NULL)
```

(1 row affected)

Completion time: 2021-05-13T23:27:57.9803898+03:00

--14.

```
INSERT INTO Format  
VALUES('40','40')
```

(1 row affected)

Completion time: 2021-05-13T23:27:46.0101553+03:00

--15.

```
INSERT INTO Expozitii_Opere_De_Arta  
VALUES('3','95')
```

(1 row affected)

Completion time: 2021-05-13T23:27:11.4000249+03:00

Update

```
--1. Adauga finalul activitati pentru angajatul cu id 2
begin tran
UPDATE Functii_Angajati
  set Data_Sfarsit = '20210315'
  where ID_Angajat = '2'
rollback tran
```

(0 rows affected)

Completion time: 2021-05-13T23:22:12.2990002+03:00

```
--2 Update procentul impozitului cu id 4
begin tran
UPDATE Impozite
  set Procent = '3'
  where ID_Impozite = '2'
rollback tran
```

(1 row affected)

Completion time: 2021-05-13T23:22:03.8356451+03:00

```
--3 Update pretul operelor ce costa mai mult de 7000 RON din depozitul cu id 1 cu 200
RON
begin tran
UPDATE Opere_De_Arta
  set [Pret(RON)]=[Pret(RON)]-'200'
  from Opere_De_Arta as O
  inner join Depozite_Opere as DO
    on O.ID_Opera=DO.ID_Opera
    inner join Depozit as D
      on D.ID_Depozit=DO.ID_Depozit
  where O.[Pret(RON)]>'7000'
rollback tran
```

(3 rows affected)

Completion time: 2021-05-13T23:21:52.9993220+03:00

```
--4 Update data scadenta a facturii cu id 5
begin tran
UPDATE Facturi
  set Data_Scadenta='20201212'
  where ID_Factura='5'
rollback tran
```

(1 row affected)

Completion time: 2021-05-13T23:21:26.2980712+03:00

```
--5 Update salariu angajati cu 200 RON
begin tran
UPDATE Angajati
```

```
set [Salariu(Brut)]=[Salariu(Brut)]+200
rollback tran
```

(12 rows affected)

Completion time: 2021-05-13T23:20:57.9311786+03:00

```
--6 Update data inceput pentru expozitia cu id 1
begin tran
UPDATE Expozitie
set Data_Inceput='20211020'
where Nume_Expozitie='Flori de toamna'
rollback tran
```

(1 row affected)

Completion time: 2021-05-13T23:20:44.9732813+03:00

```
--7 Update numarul de opere din depozitul 1 al operei cu id 43
begin tran
UPDATE Depozite_Opere
set Numar='4'
where ID_Depozit='1' and ID_Opera='43'
rollback tran
```

(1 row affected)

Completion time: 2021-05-13T23:20:33.0728543+03:00

```
--8 Update adresa clientului cu id 2
begin tran
UPDATE Clienti
set Adresa='Strada Patriei Unite nr. 20'
where ID_Client=2
rollback tran
```

(1 row affected)

Completion time: 2021-05-13T23:20:21.3965956+03:00

```
--9 Update nume angajat
begin tran
UPDATE Angajati
set Nume='Ionasanu'
where Nume='Lupascu'
rollback tran
```

(0 rows affected)

Completion time: 2021-05-13T23:19:39.3512983+03:00

```
--10 Update numar copii de pe comanda ale unui client
begin tran
UPDATE Comenzi_Opere_De_Arta
    set Bucati='3'
    where ID_Comenzi='1' and ID_Opera='12'
rollback tran
```

(1 row affected)

Completion time: 2021-05-13T23:19:31.0079374+03:00

```
--11 Update data livrare pentru o comanda
begin tran
UPDATE Comenzi
    set Data_Livrare='20211010'
    where ID_Comanda='1'
rollback tran
```

(1 row affected)

Completion time: 2021-05-13T23:19:07.1805783+03:00

```
--12 Update cod postal pentru o galerie
begin tran
UPDATE Galerii
    set Cod_Postal='302611'
    where Localitate='Iasi'
rollback tran
```

(3 rows affected)

Completion time: 2021-05-13T23:18:56.7719164+03:00

```
--13 Update numar de telefon pentru clientul cu id 1
begin tran
UPDATE Clienti
    set Numar_Telefon='0761432765'
    where ID_Client='1'
rollback tran
```

(1 row affected)

Completion time: 2021-05-13T23:18:42.0674882+03:00

```
--14 Update pret opera de arta din galeria cu id 1
begin tran
```

```

UPDATE Opere_De_Arta
set [Pret(ROn)]=[Pret(ROn)]+250
from Opere_De_Arta as O
    join Expozitii_Opere_De_Arta as EO
    on EO.ID_Opera=O.ID_Opera
    join Expozitie as E
    on E.ID_Expozitie=EO.ID_Expozitie
    join Galerii as G
    on G.ID_Galerie=E.ID_Galerie
where G.ID_Galerie='1'

```

(2 rows affected)

Completion time: 2021-05-13T23:18:10.6624934+03:00

```

rollback tran
--15 Update functie angajat id 8
begin tran
UPDATE Functii_Angajati
set ID_Functie='6'
from Functii_Angajati as FA
    join Angajati as A
    on A.ID_Angajat=FA.ID_Angajat
where A.ID_Angajat='8'
rollback tran

```

(1 row affected)

Completion time: 2021-05-13T23:17:57.8784545+03:00

Delete

```
--1.Stergere opera de arta cu cel mai mare pret al autorului cu id 1
delete O
from Opere_De_Arta as O
    join (select max(O.[Pret(RON)]) as Pret, max(O.ID_Autor) as ID_Autor
        from Opere_De_Arta as O where O.ID_Autor='1') t1
    on t1.ID_Autor=O.ID_Autor

(2 rows affected)

Completion time: 2021-05-13T23:13:37.8432449+03:00

--2.Stergere angajat care si a incetat activitatea cu mai mult de un an fata de data
prezenta
delete A
from Angajati as A
    join (select FA.Data_Sfarsit, FA.ID_Angajat
        from Functii_Angajati as FA where FA.Data_Sfarsit>cast(getdate() as
date)) t1
    on t1.ID_Angajat=A.ID_Angajat

(0 rows affected)

Completion time: 2021-05-13T23:13:24.9115641+03:00

--3.Stergere comenzile livrate de mai mult de un an
delete C
from Comenzi as C
where C.Data_Livrare>cast(getdate() as date)

(1 row affected)

Completion time: 2021-05-13T23:13:15.3940212+03:00

--4.Stergere opera de arta care costa mai mult de 10000 RON
delete O
from Opere_De_Arta as O
where O.[Pret(RON)]>10000

(0 rows affected)

Completion time: 2021-05-13T23:13:05.5026728+03:00

--5.Stergere comenzile care contin mai mult de 4 opere
delete C
from Comenzi as C
    join(select sum(CO.Bucati) as [Numar Bucati],CO.ID_Comenzi
        from Comenzi_Opere_De_Arta as CO group by CO.ID_Comenzi) t1
    on t1.ID_Comenzi=C.ID_Comanda
```


(3 rows affected)

Completion time: 2021-05-13T23:12:49.4576243+03:00

--6.Stergere autorul care nu are opere existente

```
delete A
from Autor as A
      join(select count(A.ID_Autor) as [Numar Opere], A.ID_Autor
           from Opere_De_Arta as A group by A.ID_Autor) t1
      on t1.ID_Autor=A.ID_Autor
where t1.[Numar Opere] is null
```

(0 rows affected)

Completion time: 2021-05-13T23:11:39.8684183+03:00

--7.Stergere formatele care nu se regasesc pe nicio opera

```
delete F
from Format as F
      full outer join(select FO.ID_Format
                     from Formate_Opere as FO) t1
      on t1.ID_Format=F.ID_Format
where t1.ID_Format is NULL
```

Formatul a fost sters cu succes!

(1 row affected)

Completion time: 2021-05-13T23:11:23.2314660+03:00

--8.Stergere expozitiile care nu au opere.

```
delete E
from Expozitie as E
      join(select count(EO.ID_Opera) as [Numar Opere], EO.ID_Expozitie
           from Expozitii_Opere_De_Arta as EO group by EO.ID_Expozitie) t1
      on t1.ID_Expozitie=E.ID_Expozitie
where t1.[Numar Opere] is null
```

(0 rows affected)

Completion time: 2021-05-13T23:11:14.9143811+03:00

--9.Stergere facturile care au fost emise pana in 2021

```
delete F
from Facturi as F
where datepart(year,F.Data_Scadata)<2021
```

(0 rows affected)

Completion time: 2021-05-13T23:11:03.2545794+03:00

--10.Stergere clientii care nu au nicio comanda

```
delete C
from Clienti as C
      full outer join(select C.ID_Client
                      from Comenzi as C) t1
      on t1.ID_Client=C.ID_Client
where t1.ID_Client is NULL
```

(2 rows affected)

Completion time: 2021-05-13T23:10:48.8945108+03:00

Trigger

```
--1. Trigger pentru restricție preț minim
if object_id('PretMinim','tr') is not null
    drop trigger PretMinim
go

create trigger PretMinim
on Opere_De_Arta
for insert, update
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @@pret int

    select @@pret=I.[Pret(ron)]
    from Inserted as I

    if(@@pret<1200)
    begin
        raiserror('Pretul de %d este mai mic decat valoarea minima(1200
RON)',16,1,@@pret)
        rollback transaction
    end
end
```

```
--test
insert into Opere_De_Arta
values('2','Test','2020','1000','Test')
```

Messages

```
Msg 50000, Level 16, State 1, Procedure PretMinim, Line 15 [Batch Start Line 27]
Pretul de 1000 este mai mic decat valoarea minima
Msg 3609, Level 16, State 1, Line 29
The transaction ended in the trigger. The batch has been aborted.
```

```
--2. Trigger pentru restricție salariu minim economie(2000 RON)
if object_id('SalariuMinim','tr') is not null
    drop trigger SalariuMinim
go

create trigger SalariuMinim
on Angajati
for insert, update
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @@salariu int

    select @@salariu=I.[Salariu(Brut)]
    from Inserted as I

    if(@@salariu<2000)
    begin
        raiserror('Salariul de %d este mai mic decât salariul minim pe
economie!',16,1,@@salariu)
```

```

        rollback transaction
    end
end

--test
insert into Angajati
values('Test','Test','Test','19890101','1000','False')

```

Messages

```

Msg 50000, Level 16, State 1, Procedure SalariuMinim, Line 16 [Batch Start Line 56]
Salariul de 1000 este mai mic decât salariul minim pe economie!
Msg 3609, Level 16, State 1, Line 58
The transaction ended in the trigger. The batch has been aborted.

```

Completion time: 2021-05-13T23:04:43.5646773+03:00

```

--3. Trigger pentru număr maxim de opere pe comandă(10)
if object_id('NumarOpere','tr') is not null
    drop trigger NumarOpere
go

create trigger NumarOpere
on Comenzi_Opere_De_Arta
for insert, update
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    if exists(
        select sum(CO.Bucati),CO.ID_Comenzi
        from Comenzi_Opere_De_Arta as CO, inserted as I
        group by CO.ID_Comenzi,I.ID_Comenzi
        having sum(CO.Bucati)>10 and I.ID_Comenzi=CO.ID_Comenzi
    )
    begin
        raiserror('Comanda selectata are numarul maxim de opere
adaugate!',16,1)
        rollback transaction
    end
end

--Test
update Comenzi_Opere_De_Arta
set Bucati=Bucati+3
where ID_Comenzi=1 and ID_Opera='12'

```

Messages

```

Msg 50000, Level 16, State 1, Procedure NumarOpere, Line 16 [Batch Start Line 85]
Comanda selectata are numarul maxim de opere adaugate!
Msg 3609, Level 16, State 1, Line 87
The transaction ended in the trigger. The batch has been aborted.

```

Completion time: 2021-05-13T23:05:16.1887964+03:00

```

--4. Trigger pentru ștergerea unui autor care are opere în baza de date
if object_id('AutorCuOpere','tr') is not null
    drop trigger AutorCuOpere
go

```

```

create trigger AutorCuOpere
on Autor
instead of delete
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @Nume nvarchar(20)
    select @Nume=D.Nume
    from deleted as D
    if exists(
        select count(*)
        from Opere_De_Arta as O
        join Autor as A
        on A.ID_Autor=O.ID_Autor
        group by A.Nume
        having A.Nume=@Nume and count(*)>0
    )
    begin
        raiserror('Autorul selectat are opere in baza de date!',16,1)
        rollback transaction
    end
    else
    begin
        print('Autorul a fost sters cu succes!')
        delete A
        from Autor as A
        where A.Nume=@Nume
    end
end

--Test
begin tran
delete A
from Autor as A
where A.Nume='Delacroix'
rollback tran

```

Messages

Msg 50000, Level 16, State 1, Procedure AutorCuOpere, Line 21 [Batch Start Line 127]
 Autorul selectat are opere in baza de date!
 Msg 3609, Level 16, State 1, Line 129
 The transaction ended in the trigger. The batch has been aborted.

Completion time: 2021-05-13T23:05:38.5756589+03:00

```

--5. Trigger pentru restricție impozit maxim 4%
if object_id('ImpozitMaxim','tr') is not null
    drop trigger ImpozitMaxim
go

create trigger ImpozitMaxim
on Impozite
for insert, update
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @@ProcentImpozit int

```

```

select @@ProcentImpozit=I.Procent
from inserted as I
if (@@ProcentImpozit>4)
begin
    raiserror('Procentul selectat depaseste limita admisa!',16,1)
    rollback transaction
end
end

--Test
insert into Impozite
values ('Calatorie','5')

```

Messages

Msg 50000, Level 16, State 1, Procedure ImpozitMaxim, Line 14 [Batch Start Line 155]
Procentul selectat depaseste limita admisa!

Msg 3609, Level 16, State 1, Line 157
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2021-05-13T23:06:11.1700495+03:00

```

--6. Trigger pentru ștergerea unui format care se regăsește pe mai multe opere
if object_id('FormatFolosit','tr') is not null
    drop trigger FormatFolosit
go

create trigger FormatFolosit
on Format
instead of delete
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @NumarOpere int,
            @ID_Format int,
            @NumarAparitii int
    select @ID_Format=D.ID_Format
    from deleted as D

    if exists
    (
        select count(*)
        from Opere_De_Arta as O
        join Formate_Opere as FO
        on FO.ID_Opera=O.ID_Opera
        where FO.ID_Format=@ID_Format
        group by FO.ID_Format
        having count(*)>0
    )

    begin
        raiserror('Formatul selectat spre stergere este utilizat!',16,1)
        rollback transaction
    end
    else
    begin
        print('Formatul a fost sters cu succes!')
        delete F
    end
end

```

```

        from Format as F
        where F.ID_Format=@ID_Format
    end
end
--test
begin tran
delete F
from Format as F
where F.ID_Format=4
rollback tran

```

Messages

```

Msg 50000, Level 16, State 1, Procedure FormatFolositi, Line 27 [Batch Start Line 201]
Formatul selectat spre stergere este utilizat!
Msg 3609, Level 16, State 1, Line 203
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2021-05-13T23:06:54.0746465+03:00

```

```

--7. Trigger pentru ștergerea unei funcții pe care există angajați
if object_id('FunctieCuAngajat','tr') is not null
    drop trigger FunctieCuAngajat
go

create trigger FunctieCuAngajat
on Functii
instead of delete
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @ID_Functie int

    select @ID_Functie=D.ID_Functie
    from deleted as D

    if exists(
        select F.ID_Functie, F.Denumire, FA.ID_Angajat
        from Functii as F
            left join Functii_Angajati as FA
            on FA.ID_Functie=F.ID_Functie
        where FA.ID_Angajat is not NULL and F.ID_Functie=@ID_Functie
    )
    begin
        raiserror('Functia selectata spre stergere are angajati
asignati!',16,1)
        rollback tran
    end
    else
    begin
        print('Functia a fost stearsa')
        delete F
        from Functii as F
        where F.ID_Functie=@ID_Functie
    end
end
--test
begin tran
delete F

```

```

from Functii as F
where F.Denumire='Secretar'
rollback tran

```

Messages

```

Msg 50000, Level 16, State 1, Procedure FunctieCuAngajat, Line 22 [Batch Start Line 243]
Functia selectata spre stergere are angajati asignati!
Msg 3609, Level 16, State 1, Line 245
The transaction ended in the trigger. The batch has been aborted.

```

Completion time: 2021-05-13T23:07:12.3617035+03:00

--8. Trigger pentru inserarea unei facturi cu o dată trecută de cea curentă

```

if object_id('FacturaInvalida','tr') is not null
    drop trigger FacturaInvalida
go

```

```

create trigger FacturaInvalida
on Facturi
for insert, update
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @@DataFactura date
    select @@DataFactura=I.Data_Scadenta
    from inserted as I
    if(@@DataFactura<CONVERT(DATE, GETDATE()))
    begin
        raiserror('Data inregistrata nu este valida.',16,1)
        rollback transaction
    end
end
end

```

```

--test
insert into Facturi
values ('1','1','20201010')

```

Messages

```

Msg 50000, Level 16, State 1, Procedure FacturaInvalida, Line 14 [Batch Start Line 272]
Data inregistrata nu este valida.
Msg 3609, Level 16, State 1, Line 274
The transaction ended in the trigger. The batch has been aborted.

```

Completion time: 2021-05-13T23:07:50.5301573+03:00

--9. Trigger pentru adăugarea unui angajat minor

```

if object_id('AngajatMinor','tr') is not null
    drop trigger AngajatMinor
go

```

```

create trigger AngajatMinor
on Angajati
for insert, update
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @@Varsta int
    select @@Varsta=datepart(year,GETDATE())-datepart(year,I.Data_Nastere)

```



```

from inserted as I
if(@@Varsta<18)
begin
    raiserror('Persoana selectata pentru angajare este minora.',16,1)
    rollback transaction
end
end

```

```

--test
insert into Angajati
values('Test','Test','Test','20040101','1300','False')

```

Messages

```

Msg 50000, Level 16, State 1, Procedure AngajatMinor, Line 14 [Batch Start Line 299]
Persoana selectata pentru angajare este minora.
Msg 3609, Level 16, State 1, Line 301
The transaction ended in the trigger. The batch has been aborted.

```

Completion time: 2021-05-13T23:08:31.0562946+03:00

--10.Trigger pentru inserarea aceleasi functii de 2 ori pe acelasi departament

```

if object_id('LiderMultiplu','tr') is not null
    drop trigger LiderMultiplu
go

create trigger LiderMultiplu
on Functii
for insert, update
as
begin
    if @@rowcount = 0 return;
    set nocount on;
    declare @@ID_Departament int,
            @@Denumire nvarchar(30)
    select @@ID_Departament=I.ID_Departatemt, @@Denumire=I.Denumire
    from inserted as I
    if exists(
        select F.Denumire,F.ID_Departatemt
        from Functii as F
        where F.Denumire=@@Denumire and F.ID_Departatemt=@@ID_Departament
    )
    begin
        raiserror('Deja exista aceasta functie in departamentul cu ID
%d',16,1,@@ID_Departament)
        rollback transaction
    end
end

--test
insert into Functii
values ('4','Secretar')

```

Messages

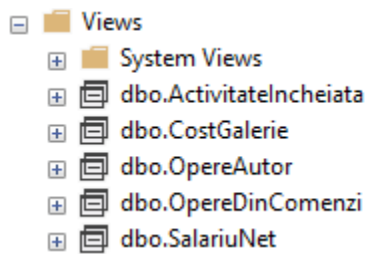
```

Msg 50000, Level 16, State 1, Procedure LiderMultiplu, Line 19 [Batch Start Line 331]
Deja exista aceasta functie in departamentul cu ID 4
Msg 3609, Level 16, State 1, Line 333
The transaction ended in the trigger. The batch has been aborted.

```

Completion time: 2021-05-13T23:08:53.6519383+03:00

View



```
--1.View pentru afisarea salariului net dupa aplicarea impozitelor
IF OBJECT_ID('SalariuNet', 'V') IS NOT NULL
    DROP VIEW SalariuNet
go
create view SalariuNet(Salariu,Nume_Angajat)
as
    with
    ImpozitTotal as
    (
        select sum(I.Procent) as Impozit, IA.ID_Angajat as ID_Angajat
        from Impozite as I
            join Impozite_Angajati as IA
            on IA.ID_Impozit=I.ID_Impozite
        group by IA.ID_Angajat
    )
    select A.Nume+' '+A.Prenume as [Nume Angajat], A.[Salariu(Brut)]-
    (A.[Salariu(Brut)]*IT.Impozit/100) as [Salariu Net]
    from Angajati as A
        join ImpozitTotal as IT
        on IT.ID_Angajat=A.ID_Angajat
```

	Salariu		Nume_Angajat
1	Gheorge	Andrei	8091
2	Anica	Alin	9016
3	Alenie	Cosmin	7614
4	Dascalescu	Lucian	7520

```
--2.View pentru afisarea costului lunar total pentru o galerie.
```

```
IF OBJECT_ID('CostGalerie', 'V') IS NOT NULL
    DROP VIEW CostGalerie
go
create view CostGalerie(CostTotal)
as
    select sum(Tp.[Pret[RON]]) as CostTotal
    from Tip_Cost as TP
```

	CostTotal
1	1750

```
--3.View pentru afisarea angajatilor care si au incetat activitea
```

```
IF OBJECT_ID('ActivitateIncheiata', 'V') IS NOT NULL
    DROP VIEW ActivitateIncheiata
go
create view ActivitateIncheiata([Nume Angajat],[Data Sfarsit])
as
```

```

select A.Nume+' '+A.Prenume as [Nume Angajat], FA.Data_Sfarsit
from Angajati as A
      join Functii_Angajati as FA
      on FA.ID_Angajat=A.ID_Angajat
where FA.Data_Sfarsit is not null

```

	Nume Angajat		Data Sfarsit
1	Ionescu	Mihai	2019-06-04

--4.View pentru afisarea operelor de arta din comenzi alaturi de numarul disponibil in depozite

```

IF OBJECT_ID('OpereDinComenzi', 'V') IS NOT NULL
    DROP VIEW OpereDinComenzi
go
create view OpereDinComenzi([Nume Opera],[Bucati],[Dimensiune],[Numar Copii])
as
select O.Nume, CA.Bucati, concat(F.Latime,'*',F.Lungime) as Dimensiune,
DO.Numar
from Opere_De_Arta as O
      join Comenzi_Opere_De_Arta as CA
      on O.ID_Opera=CA.ID_Opera
      join Format as F
      on F.ID_Format=CA.ID_Format
      join Depozite_Opere as DO
      on DO.ID_Opera=O.ID_Opera

```

	Nume Opera	Bucati	Dimensiune	Numar Copii
1	The Coronation of Napoleon	5	40*40	6

--5.View pentru gruparea operelor in functie de autorul acestora

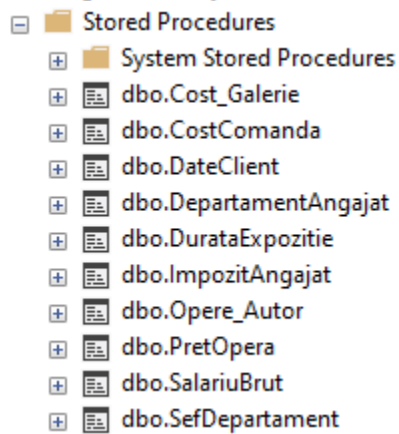
```

IF OBJECT_ID('OpereAutor', 'V') IS NOT NULL
    DROP VIEW OpereAutor
go
create view OpereAutor([Nume Opera],[Nume Autor])
as
select O.Nume, concat(A.Prenume,' ',A.Nume) as [Nume Autor]
from Opere_De_Arta as O
      join Autor as A
      on A.ID_Autor=O.ID_Autor

```

	Nume Opera	Nume Autor	
1	Liberty Leading the People	Eugène	Delacroix
2	The Raft of the Medusa	Théodore	Géricault
3	The Coronation of Napoleon	David	Jacques-Louis
4	Death of the Virgin	Caravaggio	Michelangelo Merisi
5	Oath of the Horatii	David	Jacques-Louis
6	The Bolt	Fragonard	Jean-Honoré
7	The Death of Sardanapalus	Eugène	Delacroix
8	Man with a Glove	Tiziano	Vecelli
9	The Barque of Dante	Eugène	Delacroix
10	The Intervention of the Sabine Women	David	Jacques-Louis
11	Portrait of Madame Récamier	David	Jacques-Louis
12	Greece on the Ruins of Missolonghi	Eugène	Delacroix
13	The Sultan of Morocco	Eugène	Delacroix
14	Jacob Wrestling with the Angel	Eugène	Delacroix
15	Orphan Girl at the Cemetery	Eugène	Delacroix
16	Portrait of Frédéric Chopin and George Sand	Eugène	Delacroix
17	The Charging Chasseur	Théodore	Géricault
18	The Wounded Cuirassier	Théodore	Géricault
19	Portrait of a Kleptomaniac	Théodore	Géricault

Proceduri stocate



```
--1.Aflarea salariului(BRUT) unui angajat
if object_id('SalariuBrut','p') is not null
    drop proc SalariuBrut;
go

create proc SalariuBrut
    @Nume_Angajat nvarchar(30),
    @salariu as int = 0 output
as
begin
    select @salariu=A.[Salariu(Brut)]
    from Angajati as A
    where A.Nume=@Nume_Angajat
    return;
end

declare @salariu_ int;

Exec SalariuBrut
    @Nume_Angajat='Anica',
    @salariu = @salariu_ OUTPUT;

select @salariu_ as 'Salariu_Angajat'
```

```
--2.Aflarea pretului unei opere
if object_id('PretOpera','p') is not null
    drop proc PretOpera
go

create proc PretOpera
    @Nume_Opera nvarchar(50),
    @Pret int output
as
begin
    select @Pret=O.[Pret(RON)]
    from Opere_De_Arta as O
    where O.Nume=@Nume_Opera
    return
end
```

```

declare @Pret_ int

Exec PretOpera
    @Nume_Opera='Liberty Leading the People',
    @Pret=@Pret_ output;

select @Pret_ as 'Pret Opera de arta'

--3.Aflarea costului total al unei comenzi
if object_id('CostComanda','p') is not null
    drop proc CostComanda
go

create proc CostComanda
    @ID_Comanda int,
    @Cost int output
as
begin
    select @Cost=sum(O.[Pret(RON)])
    from Comenzi_Opere_De_Arta as C
        join Opere_De_Arta as O
        on C.ID_Opera=O.ID_Opera
    group by C.ID_Comenzi
    having C.ID_Comenzi=@ID_Comanda
    return
end

declare @Cost_Comanda int

Exec CostComanda
    @ID_Comanda='1',
    @Cost=@Cost_Comanda output;

select @Cost_Comanda as 'Cost total comanda'

--4.Aflarea sefului de pe un anumit departament
if object_id('SefDepartament','p') is not null
    drop proc SefDepartament
go

create proc SefDepartament
    @Nume_Departament nvarchar(20),
    @Sef_Departament nvarchar(50) output
as
begin
    select @Sef_Departament=concat(A.Nume,' ',A.Prenume)
    from Departamente as D
        join Functii as F
        on F.ID_Departament=D.ID_Departament
        join Functii_Angajati as FA
        on FA.ID_Functie=F.ID_Functie
        join Angajati as A
        on FA.ID_Angajat=A.ID_Angajat
    where A.Lider='True' and D.Nume_Departament=@Nume_Departament
    return
end

declare @SefDepartament nvarchar(50)

```

```

Exec SefDepartament
    @Nume_Departament='Contabilitate',
    @Sef_Departament=@SefDepartament output;

select @SefDepartament as 'Sef departament'

--5.Aflarea costului total dintr o galerie pe un an
if object_id('Cost_Galerie','p') is not null
    drop proc Cost_Galerie
go

create proc Cost_Galerie
    @NumeGalerie nvarchar(30),
    @An int,
    @Cost int output
as
begin
    select @Cost=sum(T.[Pret[RON]])
    from Galerii as G
        join Facturi as F
        on F.ID_Galerie=G.ID_Galerie
        join Tip_Cost as T
        on F.ID_Tip_Cost=T.ID_Tip_Cost
    where datepart(year,F.Data_Scadenta)=@An
        and G.Nume_Galerie=@NumeGalerie
    return
end

declare @CostGalerie int

Exec Cost_Galerie
    @NumeGalerie='Theodor Pallady',
    @An='2021',
    @Cost=@CostGalerie output;
select @CostGalerie as 'Cost galerie pe un an'

--6.Aflarea celei mai noi opere a unui autor
if object_id('Opere_Autor','p') is not null
    drop proc Opere_Autor
go

create proc Opere_Autor
    @Autor nvarchar(20),
    @Opere nvarchar(30) output
as
begin
    select top(1) @Opere=O.Nume
    from Opere_De_Arta as O
        join Autor as A
        on A.ID_Autor=O.ID_Autor
    where A.Nume=@Autor
    order by O.An desc
    return
end

declare @NumeOpere nvarchar(30)

```

```

Exec Opere_Autor
    @Autor='Delacroix',
    @Opere=@NumeOpere output;
select @NumeOpere as 'Opere autor'

--7.Aflarea duratei unei expozitii
if object_id('DurataExpozitie','p') is not null
    drop proc DurataExpozitie
go

create proc DurataExpozitie
    @NumeExpozitie nvarchar(50),
    @Durata int output
as
begin
    select @Durata= datediff(day,E.Data_Inceput,E.Data_Sfarsit)
    from Expozitie as E
    where E.Nume_Expozitie=@NumeExpozitie
    return
end

declare @Durata_ int

Exec DurataExpozitie
    @NumeExpozitie='Romantism',
    @Durata=@Durata_ output;
select @Durata_ as 'Durata Expozitie'

--8.Aflarea impozitului total pentru un angajat
if object_id('ImpozitAngajat','p') is not null
    drop proc ImpozitAngajat
go

create proc ImpozitAngajat
    @NumeAngajat nvarchar(20),
    @ImpozitTotal int output
as
begin
    select @ImpozitTotal=sum(I.Procent)
    from Angajati as A
        join Impozite_Angajati as IA
        on A.ID_Angajat=IA.ID_Angajat
        join Impozite as I
        on I.ID_Impozite=IA.ID_Impozit
    group by A.Nume
    having A.Nume=@NumeAngajat
end

declare @Impozit int

Exec ImpozitAngajat
    @NumeAngajat='Anica',
    @ImpozitTotal=@Impozit output;
select @Impozit as 'Impozit Angajat'

--9.Aflarea adresa pentru un client
if object_id('DateClient','p') is not null
    drop proc DateClient

```



```

go

create proc DateClient
    @NumeClient nvarchar(20),
    @Adresa nvarchar(50) output
as
begin
    select @Adresa=C.Adresa
    from Clienti as C
    where C.Nume=@NumeClient
    return
end

declare @AdresaClient nvarchar(50)

Exec DateClient
    @NumeClient='Popescu',
    @Adresa=@AdresaClient output;

select @AdresaClient as 'Adresa Client'

--10.Aflarea departament de care apartine un angajat
if object_id('DepartamentAngajat','p') is not null
    drop proc DepartamentAngajat
go

create proc DepartamentAngajat
    @NumeAngajat nvarchar(20),
    @Departament nvarchar(30) output
as
begin
    select @Departament=D.Nume_Departament
    from Departamente as D
        join Functii as F
        on D.ID_Departament=F.ID_Departament
        join Functii_Angajati as FA
        on FA.ID_Functie=F.ID_Functie
        join Angajati as A
        on A.ID_Angajat=FA.ID_Angajat
    where A.Nume=@NumeAngajat
    return
end

declare @Departament_ nvarchar(30)

Exec DepartamentAngajat
    @NumeAngajat='Anica',
    @Departament=@Departament_ output;

select @Departament_ as 'Departament angajat'

```

Tranzactii

```
--1. Inserare in Opere_De_Arta
declare @errnum as int;
begin tran
    insert into Opere_De_Arta
    values
        (1, 'Mona Lisa', '1503', 6000, 'The Mona Lisa is a half-length portrait painting by
the Italian artist Leonardo da Vinci. It is considered an archetypal masterpiece of the
Italian Renaissance, and has been described as "the best known, the most visited, the
most written about, the most sung about, the most parodied work of art in the world."')
    insert into Opere_De_Arta
    values
        (1, 'Vitruvian Man', '1490', 7300, 'The Vitruvian Man is a drawing made by the
Italian polymath Leonardo da Vinci in about 1490. It is accompanied by notes based on the
work of the Roman architect Vitruvius.')

set @errnum = @@ERROR
if @errnum <> 0
begin
    print 'Inserarea in Opere_De_Arta a esuat cu eroare: ' + CAST(@errnum as varchar);
    rollback
end
else
begin
    commit
end;

(1 row affected)

(1 row affected)

Completion time: 2021-05-13T22:26:30.4537625+03:00
```

```
--2. Inserare in Clienti
declare @errnum as int;
begin tran
    insert into Clienti
    values ('Berbecaru', 'Florin', '0732531331', 'Strada Petalelor nr. 5', 'Jibou'),
        ('Mihailescu', 'Arnold', '0725123765', 'Strada Renasterii nr. 10', 'Zalau')
set @errnum = @@ERROR
if @errnum <> 0
begin
    print 'Inserarea in Clienti a esuat cu eroare: ' + CAST(@errnum as varchar);
    rollback
end
else
begin
    commit
end;
```

(2 rows affected)

Completion time: 2021-05-13T22:27:41.4973492+03:00

```
--3. Inserare in Expozitie
declare @errnum as int;
begin tran
    insert into Expozitie
    values (3,'Marile minuni','20201004','20210510'),
           (3,'Petale aurii','20190314','20190520')
set @errnum = @@ERROR
if @errnum <> 0
begin
    print 'Inserarea in Expozitie a esuat cu eroare: ' + CAST(@errnum as varchar);
    rollback
end
else
begin
    commit
end;
```

(2 rows affected)

Completion time: 2021-05-13T22:29:31.6847423+03:00

```
--4. Inserare Angajat, Impozite si Functie
declare @errnum as int;
begin tran
    begin tran
        set identity_insert Angajati on;
        insert into Angajati
        (ID_Angajat,Nume,Prenume,CNP>Data_Nastere,[Salariu(Brut)],Lider)
        values (16,'Dascalescu','Lucian','5001242659320','20001010',7000,0)
        set identity_insert Angajati off;
        commit;

    begin tran
        insert into Impozite_Angajati
        values(1,16),
              (2,16),
              (3,16)

        commit

    begin tran
        insert into Functii_Angajati
        values(2,16,'20210101',NULL)

        commit
set @errnum = @@ERROR
if @errnum <> 0
```

```

begin
    print 'Inserare esuata cu cod: ' + CAST(@errnum as varchar);
    rollback
end
else
begin
    commit
end;

(1 row affected)

(3 rows affected)

(1 row affected)

Completion time: 2021-05-13T22:52:55.1355013+03:00

```

```

--5.Stergere opera cu pret mai mare de 10000
declare @errnum as int;
begin tran
    delete O
    from Opere_De_Arta as O
    where O.[Pret(RON)]>10000

set @errnum = @@ERROR
if @errnum <> 0
begin
    print 'Stergerea din Opere_De_Arta a esuat cu eroare: ' + CAST(@errnum as
varchar);
    rollback
end
else
begin
    commit
end;

(0 rows affected)

Completion time: 2021-05-13T22:47:25.0958420+03:00

```

```

--6. Adaugare bonus salar 200 RON celor nascuti inainte de 1900
declare @errnum as int;
begin tran
    update Angajati
    set [Salariu(Brut)]=[Salariu(Brut)]+200
    where DATEPART(year,Data_Nastere)<1900

set @errnum = @@ERROR
if @errnum <> 0
begin

```

```

        print 'Cresterea salariilor a esuat cu eroare: ' + CAST(@errnum as varchar);
        rollback
    end
else
begin
    commit
end;

```

(0 rows affected)

Completion time: 2021-05-13T22:48:09.5962139+03:00

--7. Concediere Sef Contabilitate, impozitele sale si atribuire functie altui angajat

```

declare @errnum as int;
begin tran
    begin tran
        delete IA
        from Impozite_Angajati as IA
            join Angajati as A
            on A.ID_Angajat=IA.ID_Angajat
            join Functii_Angajati as FA
            on FA.ID_Angajat=A.ID_Angajat
            join Functii as F
            on F.ID_Functie=FA.ID_Functie
        where F.Denumire='Sef Contabilitate'
    commit tran

    begin tran
        delete A
        from Angajati as A
            join Functii_Angajati as FA
            on FA.ID_Angajat=A.ID_Angajat
            join Functii as F
            on F.ID_Functie=FA.ID_Functie
        where F.Denumire='Sef Contabilitate'
    commit

    begin tran
        update Angajati
        set Lider='True',[Salariu(Brut)]=[Salariu(Brut)]+1000
        where Nume='Dascalescu'
    commit tran

    begin tran
        insert into Functii_Angajati
        values( 7,16,'20201010',NULL)
    commit tran

set @errnum = @@ERROR
if @errnum <> 0
begin
    print 'Modificarea a esuat cu eroare: ' + CAST(@errnum as varchar);
    rollback
end

```

```
else  
begin
```

```
    commit  
end;
```

```
(0 rows affected)
```

```
(0 rows affected)
```

```
(1 row affected)
```

```
(1 row affected)
```

```
Completion time: 2021-05-13T22:54:18.3842866+03:00
```