

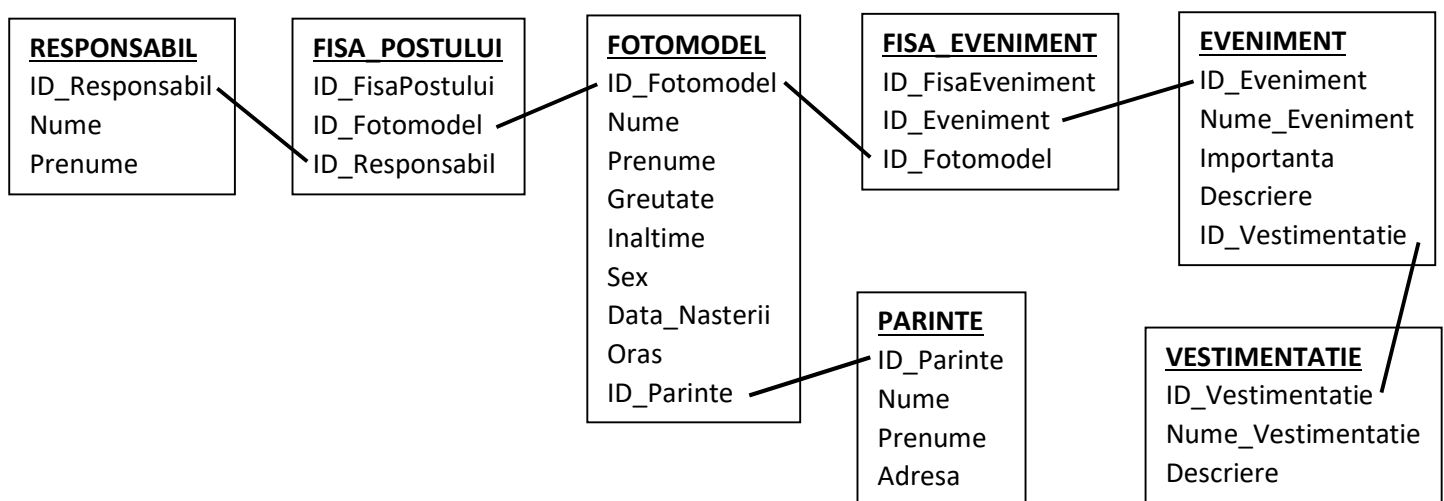
PROIECT BD

Mărgăritescu Vlad – 333AC

1.Cerinta Proiect:

35. Aplicatie pentru evidenta informatiilor legate de o agentie de fotomodele cu urmarirea participarii la anumite evenimente.

2.Diagrama Bazei de Date:



3.Descriere tabele:

Tabele principale: FOTOMODEL, RESPONSABIL, EVENIMENT, PARINTE, VESTIMENTATIE

Tabele de legatura: FISA_POSTULUI, FISA_EVENTIMENT

4.Relatiile intre tabele:

	Fotomodel	Responsabil	Eveniment	Parinte	Vestimentatie
Fotomodel	X	1:N	1:N	1:1	-
Responsabil	1:N	X	-	-	-
Eveniment	1:N	-	X	-	1:1
Parinte	1:N	-	-	X	-
Vestimentatie	-	-	1:N	-	X

X = 1:1 (autoreferire)

1:N = one to many

N:N = many to many

Fotomodel – Responsabil = 1:N + 1:N = N:N (rezolvata prin FISA_POSTULUI)

Fotomodel – Eveniment = 1:N + 1:N = N:N (rezolvata prin FISA_EVENTIMENT)

Fotomodel – Parinte = 1:1 + 1:N = 1:N

Eveniment – Vestimentatie = 1:1 + 1:N = 1:N

5.Concluzie:

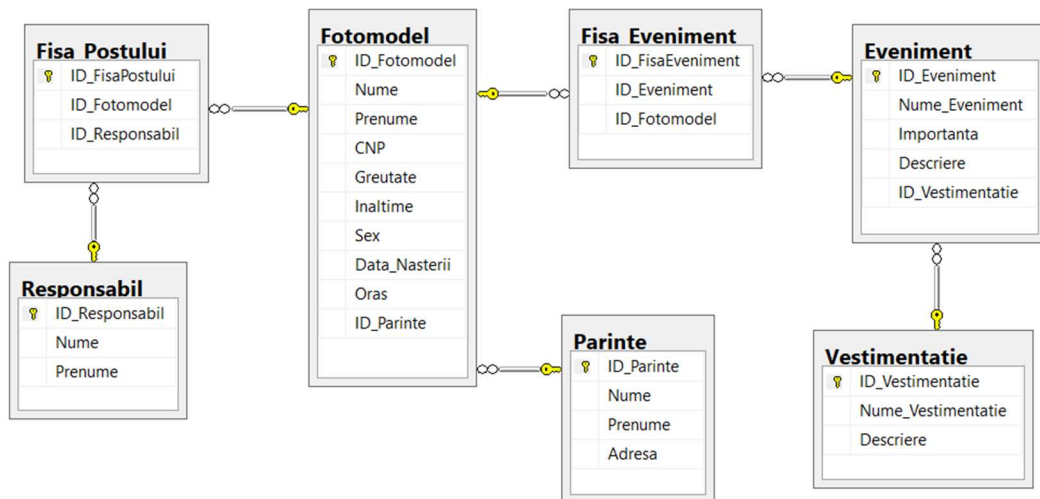
Am folosit 5 tabele principale (fara cele de legatura) in arhitectura Bazei de Date.

Mai mult, am 2 relatii 1:N si 2 relatii N:N. Asadar, cerintele de proiectare sunt indeplinite.

6.Constrangeri (Explicate):

- 1 Fotomodel are mai multi Responsabili.
- 1 Fotomodel participa la mai multe Evenimente.
- 1 Fotomodel are 1 Parinte.
- 1 Responsabil are in grija mai multe Fotomodele.
- 1 Eveniment are mai multe Fotomodele care participa.
- 1 Eveniment are 1 Vestimentatie.
- 1 Parinte are mai multe Fotomodele.
- 1 Vestimentatie este folosita la mai multe Evenimente.

7.Schema Bazei de Date in aplicatia SQL Server Management:



8.Codul folosit in toate fisierele .php pentru a ma conecta la Baza de Date:

```
<?php
$servername = "DESKTOP-8VHH2QI\\SQLEXPRESS";
$options = [
    "Database" => "agentie_fotomodele",
    "Uid" => "",
    "PWD" => "",
    'encrypt' => 'no'
];
$conn = sqlsrv_connect($servername, $options);
if($conn == false)
    die(print_r(sqlsrv_errors(), true));
else echo 'Conectat cu Succes ';
```

9.Crearea BD, a Tabelelor, precum si a constrangerilor:

Tabelul FOTOMODEL

```
CREATE TABLE [dbo].[Fotomodel](
    [ID_Fotomodel] [int] IDENTITY(1,1) NOT NULL,
    [Nume] [nvarchar](50) NOT NULL,
    [Prenume] [nvarchar](50) NOT NULL,
    [CNP] [char](13) NOT NULL,
    [Greutate] [int] NOT NULL,
    [Inaltime] [int] NOT NULL,
    [Sex] [char](1) NOT NULL CONSTRAINT [DF_Fotomodel_Sex] DEFAULT ('F'),
    [Data_Nasterii] [datetime] NULL,
    [Oras] [nvarchar](50) NULL,
    [ID_Parinte] [int] NULL,
    CONSTRAINT [PK_Fotomodel] PRIMARY KEY CLUSTERED
(
    [ID_Fotomodel] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [IX_Fotomodel] UNIQUE NONCLUSTERED
(
    [ID_Fotomodel] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[Fotomodel] WITH CHECK ADD CONSTRAINT [FK_Fotomodel_Parinte] FOREIGN
KEY([ID_Parinte])
REFERENCES [dbo].[Parinte] ([ID_Parinte])
GO

ALTER TABLE [dbo].[Fotomodel] CHECK CONSTRAINT [FK_Fotomodel_Parinte]
GO

ALTER TABLE [dbo].[Fotomodel] WITH CHECK ADD CONSTRAINT [CK_Fotomodel] CHECK ((([Sex]='F'
OR [Sex]='M'))
GO

ALTER TABLE [dbo].[Fotomodel] CHECK CONSTRAINT [CK_Fotomodel]
```

Tabelul EVENIEMENT

```
CREATE TABLE [dbo].[Eveniment](
    [ID_Eveniment] [int] NOT NULL,
    [Nume_Eveniment] [nvarchar](50) NOT NULL,
    [Importanta] [int] NULL,
    [Descriere] [nvarchar](50) NULL,
    [ID_Vestimentatie] [int] NOT NULL,
    CONSTRAINT [PK_Eveniment] PRIMARY KEY CLUSTERED
(
    [ID_Eveniment] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[Eveniment] WITH CHECK ADD CONSTRAINT [FK_Eveniment_Vestimentatie]
FOREIGN KEY([ID_Vestimentatie])
REFERENCES [dbo].[Vestimentatie] ([ID_Vestimentatie])
GO
```

```
ALTER TABLE [dbo].[Eveniment] CHECK CONSTRAINT [FK_Eveniment_Vestimentatie]
```

Tabelul FISA_EVENIEMENT

```
CREATE TABLE [dbo].[Fisa_Eveniment](
    [ID_FisaEveniment] [int] NOT NULL,
    [ID_Eveniment] [int] NOT NULL,
    [ID_Fotomodel] [int] NOT NULL,
    CONSTRAINT [PK_Fisa_Eveniment] PRIMARY KEY CLUSTERED
(
    [ID_FisaEveniment] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[Fisa_Eveniment] WITH CHECK ADD CONSTRAINT
[FK_Fisa_Eveniment_Eveniment] FOREIGN KEY([ID_Eveniment])
REFERENCES [dbo].[Eveniment] ([ID_Eveniment])
GO
```

```
ALTER TABLE [dbo].[Fisa_Eveniment] CHECK CONSTRAINT [FK_Fisa_Eveniment_Eveniment]
GO
```

```
ALTER TABLE [dbo].[Fisa_Eveniment] WITH CHECK ADD CONSTRAINT
[FK_Fisa_Eveniment_Fotomodel] FOREIGN KEY([ID_Fotomodel])
REFERENCES [dbo].[Fotomodel] ([ID_Fotomodel])
GO
```

```
ALTER TABLE [dbo].[Fisa_Eveniment] CHECK CONSTRAINT [FK_Fisa_Eveniment_Fotomodel]
```

Tabelul FISA_POSTULUI

```
CREATE TABLE [dbo].[Fisa_Postului](
    [ID_FisaPostului] [int] NOT NULL,
    [ID_Fotomodel] [int] NOT NULL,
    [ID_Responsabil] [int] NOT NULL,
    CONSTRAINT [PK_Fisa_Postului] PRIMARY KEY CLUSTERED
(
    [ID_FisaPostului] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Fisa_Postului] WITH CHECK ADD CONSTRAINT [FK_Fisa_Postului_Fotomodel]
FOREIGN KEY([ID_Fotomodel])
REFERENCES [dbo].[Fotomodel] ([ID_Fotomodel])
GO

ALTER TABLE [dbo].[Fisa_Postului] CHECK CONSTRAINT [FK_Fisa_Postului_Fotomodel]
GO

ALTER TABLE [dbo].[Fisa_Postului] WITH CHECK ADD CONSTRAINT [FK_Fisa_Postului_Responsabil]
FOREIGN KEY([ID_Responsabil])
REFERENCES [dbo].[Responsabil] ([ID_Responsabil])
GO

ALTER TABLE [dbo].[Fisa_Postului] CHECK CONSTRAINT [FK_Fisa_Postului_Responsabil]
```

Tabelul RESPONSABIL

```
CREATE TABLE [dbo].[Responsabil](
    [ID_Responsabil] [int] NOT NULL,
    [Nume] [nvarchar](50) NOT NULL,
    [Prenume] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Responsabil] PRIMARY KEY CLUSTERED
(
    [ID_Responsabil] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Tabelul VESTIMENTATIE

```
CREATE TABLE [dbo].[Vestimentatie](
    [ID_Vestimentatie] [int] NOT NULL,
    [Nume_Vestimentatie] [nchar](10) NOT NULL,
    [Descriere] [nchar](10) NULL,
    CONSTRAINT [PK_Vestimentatie] PRIMARY KEY CLUSTERED
(
    [ID_Vestimentatie] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Tabelul PARINTE

```
CREATE TABLE [dbo].[Parinte](
    [ID_Parinte] [int] NOT NULL,
    [Nume] [nvarchar](50) NOT NULL,
    [Prenume] [nvarchar](50) NOT NULL,
    [Adresa] [nvarchar](50) NULL,
    CONSTRAINT [PK_Parinte] PRIMARY KEY CLUSTERED
(
    [ID_Parinte] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

10.Introducere date in BD:

-Introducere in Tabelul Parinte:

```
INSERT INTO Parinte(ID_Parinte, Nume, Prenume, Adresa) VALUES ('1','Ionescu','Marin','Iasi');
INSERT INTO Parinte(ID_Parinte, Nume, Prenume, Adresa) VALUES
('2','Georgescu','Ioana','Bucuresti');
INSERT INTO Parinte(ID_Parinte, Nume, Prenume, Adresa) VALUES ('3','Basescu','Vasile','Cluj-
Napoca');
INSERT INTO Parinte(ID_Parinte, Nume, Prenume, Adresa) VALUES ('4','Mandarici','Diana','Buzau');
INSERT INTO Parinte(ID_Parinte, Nume, Prenume, Adresa) VALUES ('5','Anton','Mihai','Chisinau');
INSERT INTO Parinte(ID_Parinte, Nume, Prenume, Adresa) VALUES ('6','Iancu','Iulian','Oradea');
INSERT INTO Parinte(ID_Parinte, Nume, Prenume, Adresa) VALUES ('7','Encescu','Mariana','Pitesti');
```

-Introducere in Tabelul Vestimentatie:

```
INSERT INTO Vestimentatie(ID_Vestimentatie, Nume_Vestimentatie, Descriere) VALUES
('1','Elegant','Seara');
INSERT INTO Vestimentatie(ID_Vestimentatie, Nume_Vestimentatie, Descriere) VALUES
('2','Sport','Relaxare');
INSERT INTO Vestimentatie(ID_Vestimentatie, Nume_Vestimentatie, Descriere) VALUES
('3','Costum','Ocazii');
```

-Introducere in Tabelul Eveniment:

```
INSERT INTO Eveniment(ID_Eveniment, Nume_Eveniment, Importanta, Descriere, ID_Vestimentatie)
VALUES ('1','Untold','1','Frumos','1');
INSERT INTO Eveniment(ID_Eveniment, Nume_Eveniment, Importanta, Descriere, ID_Vestimentatie)
VALUES ('2','Maneliada','2','Sistem','3');
INSERT INTO Eveniment(ID_Eveniment, Nume_Eveniment, Importanta, Descriere, ID_Vestimentatie)
VALUES ('3','Concert','3','Spectalos','1');
INSERT INTO Eveniment(ID_Eveniment, Nume_Eveniment, Importanta, Descriere, ID_Vestimentatie)
VALUES ('4','Parada','2','Impresionant','2');
INSERT INTO Eveniment(ID_Eveniment, Nume_Eveniment, Importanta, Descriere, ID_Vestimentatie)
VALUES ('5','ZiuaOrasului','1','Aniversare','3');
```

-Introducere in Tabelul Responsabil:

```
INSERT INTO Responsabil(ID_Responsabil, Nume, Prenume) VALUES ('1','Dumitrescu','Ionut');
INSERT INTO Responsabil(ID_Responsabil, Nume, Prenume) VALUES ('2','Marinceci','Gabriel');
INSERT INTO Responsabil(ID_Responsabil, Nume, Prenume) VALUES ('3','Avram','Iancu');
INSERT INTO Responsabil(ID_Responsabil, Nume, Prenume) VALUES ('4','Patrascu','Andrei');
```

-Introducere in Tabelul Fotomodel:

```
SET IDENTITY_INSERT Fotomodel ON
INSERT INTO Fotomodel(ID_Fotomodel, Nume, Prenume, CNP, Greutate, Inaltime, Sex,
Data_Nasterii, Oras, ID_Parinte) VALUES ('1','Marinescu','Diana','6010101010101','45','165','F','2001-
01-01','Vaslui','2');
INSERT INTO Fotomodel(ID_Fotomodel, Nume, Prenume, CNP, Greutate, Inaltime, Sex,
Data_Nasterii, Oras, ID_Parinte) VALUES ('2','Ionescu','Maria','6020202020202','62','175','F','2002-
02-02','Botosani','1');
INSERT INTO Fotomodel(ID_Fotomodel, Nume, Prenume, CNP, Greutate, Inaltime, Sex,
Data_Nasterii, Oras, ID_Parinte) VALUES ('3','Molotov','Ion','5030303030303','70','182','M','2003-03-
03','Bucuresti','1');
INSERT INTO Fotomodel(ID_Fotomodel, Nume, Prenume, CNP, Greutate, Inaltime, Sex,
Data_Nasterii, Oras, ID_Parinte) VALUES
('4','Avramescu','Catalin','5020104051254','76','185','M','2002-01-04','Pitesti','3');
INSERT INTO Fotomodel(ID_Fotomodel, Nume, Prenume, CNP, Greutate, Inaltime, Sex,
Data_Nasterii, Oras, ID_Parinte) VALUES ('5','Popescu','Ioana','6030107051254','45','164','F','2003-
01-07','Pitesti','4');
SET IDENTITY_INSERT Fotomodel OFF
```

-Introducere in Tabelul Fisa_Postului:

```
INSERT INTO Fisa_Postului(ID_FisaPostului, ID_Fotomodel, ID_Responsabil) VALUES ('1','1','1');
INSERT INTO Fisa_Postului(ID_FisaPostului, ID_Fotomodel, ID_Responsabil) VALUES ('2','3','1');
INSERT INTO Fisa_Postului(ID_FisaPostului, ID_Fotomodel, ID_Responsabil) VALUES ('3','1','2');
INSERT INTO Fisa_Postului(ID_FisaPostului, ID_Fotomodel, ID_Responsabil) VALUES ('4','2','4');
INSERT INTO Fisa_Postului(ID_FisaPostului, ID_Fotomodel, ID_Responsabil) VALUES ('5','4','3');
```

-Introducere in Tabelul Fisa_Eveniment:

```
INSERT INTO Fisa_Eveniment(ID_FisaEveniment, ID_Eveniment, ID_Fotomodel) VALUES ('1','1','1');
INSERT INTO Fisa_Eveniment(ID_FisaEveniment, ID_Eveniment, ID_Fotomodel) VALUES ('2','3','2');
INSERT INTO Fisa_Eveniment(ID_FisaEveniment, ID_Eveniment, ID_Fotomodel) VALUES ('3','2','2');
INSERT INTO Fisa_Eveniment(ID_FisaEveniment, ID_Eveniment, ID_Fotomodel) VALUES ('4','3','5');
INSERT INTO Fisa_Eveniment(ID_FisaEveniment, ID_Eveniment, ID_Fotomodel) VALUES ('5','1','3');
```

11.INSERT in BD (la 2 tabele):

Tabelul Parinte:

```
<?php
// verifica daca exista date transmise
if ($_POST['Id_Parinte'] != ""
    && $_POST['Nume'] != ""
    && $_POST['Prenume'] != ""
    && $_POST['Adresa'] != ""
)
    // preia datele din formular
    $Id_Parinte = $_POST['Id_Parinte'];
    $Nume = $_POST['Nume'];
    $Prenume = $_POST['Prenume'];
    $Adresa = $_POST['Adresa'];

    $query = "INSERT INTO Parinte(ID_Parinte, Nume, Prenume, Adresa) VALUES
    ('".$_Id_Parinte."', '".$_Nume."', '".$_Prenume."', '".$_Adresa."');";
    $results = sqlsrv_query($conn,$query);

    if($results)
        echo ' Inserare cu Succes';
    else
        echo ' Eroare la Inserare';
?>
```

Tabelul Vestimentatie:

```
<?php
// verifica daca exista date transmise
if ($_POST['ID_Vestimentatie'] != ""
    && $_POST['Nume_Vestimentatie'] != ""
    && $_POST['Descriere'] != ""
)
    // preia datele din formular
    $ID_Vestimentatie = $_POST['ID_Vestimentatie'];
    $Nume_Vestimentatie = $_POST['Nume_Vestimentatie'];
    $Descriere = $_POST['Descriere'];

    $query = "INSERT INTO Vestimentatie(ID_Vestimentatie, Nume_Vestimentatie, Descriere) VALUES
    ('".$_ID_Vestimentatie."', '".$_Nume_Vestimentatie."', '".$_Descriere."');";
    $results = sqlsrv_query($conn,$query);

    if($results)
        echo ' Inserare cu Succes';
    else
        echo ' Eroare la Inserare';
?>
```


12.UPDATE in BD (la 2 tabele):

Tabelul Parinte:

```
<?php
// verifica daca exista date transmise
if ($_POST['Id_Parinte'] != ""
    && $_POST['Nume'] != ""
    && $_POST['Prenume'] != ""
    && $_POST['Adresa'] != ""
)
    // preia datele din formular
    $Id_Parinte = $_POST['Id_Parinte'];
    $Nume = $_POST['Nume'];
    $Prenume = $_POST['Prenume'];
    $Adresa = $_POST['Adresa'];

    $query = "UPDATE Parinte SET Id_Parinte='$Id_Parinte', Nume='$Nume', Prenume='$Prenume',
Adresa='$Adresa' WHERE Id_Parinte='$Id_Parinte'";
    $results = sqlsrv_query($conn,$query);

    if($results)
        echo ' Update cu Succes';
    else
        echo ' Eroare la Update';
?>
```

Tabelul Vestimentatie:

```
<?php
// verifica daca exista date transmise
if ($_POST['ID_Vestimentatie'] != ""
    && $_POST['Nume_Vestimentatie'] != ""
    && $_POST['Descriere'] != ""
)
    // preia datele din formular
    $ID_Vestimentatie = $_POST['ID_Vestimentatie'];
    $Nume_Vestimentatie = $_POST['Nume_Vestimentatie'];
    $Descriere = $_POST['Descriere'];

    $query = "UPDATE Vestimentatie SET ID_Vestimentatie='$ID_Vestimentatie',
Nume_Vestimentatie='$Nume_Vestimentatie', Descriere='$Descriere' WHERE
ID_Vestimentatie='$ID_Vestimentatie'";
    $results = sqlsrv_query($conn,$query);

    if($results)
        echo ' Update cu Succes';
    else
        echo ' Eroare la Update';
?>
```

13.DELETE in BD (la 2 tabele):

Tabelul Parinte:

```
<?php
if(isset($_POST["Id_Parinte"])) //se verifica id-ul din link
{
    $Id_Parinte = $_POST["Id_Parinte"]; //aici se pune

    $sql = "DELETE FROM Parinte WHERE Id_Parinte= '$Id_Parinte'";
    $results = sqlsrv_query($conn,$sql);
    if($results)
        echo ' Delete cu Succes';
    else
        echo ' Eroare la Delete';
}
?>
```

Tabelul Vestimentatie:

```
<?php
if(isset($_POST["ID_Vestimentatie"])) //se verifica id-ul din link
{
    $ID_Vestimentatie = $_POST["ID_Vestimentatie"]; //aici se pune

    $sql = "DELETE FROM Vestimentatie WHERE ID_Vestimentatie= '$ID_Vestimentatie'";
    $results = sqlsrv_query($conn,$sql);
    if($results)
        echo ' Delete cu Succes';
    else
        echo ' Eroare la Delete';
}
?>
```

14. Interogari Simple (6 exemple):

a) Fotomodele dintr-un anumit oras si informatii despre Parintii lor

\$Oras = \$_POST['Oras'];

```
$sql = "SELECT F.ID_Fotomodel, F.Oras, F.ID_Parinte,P.Nume,P.Prenume, P.Adresa
FROM Fotomodel F JOIN Parinte P ON (F.ID_Parinte=P.ID_Parinte)
WHERE F.Oras ='$Oras'";
```

b) Un Fotomodel si toti Responsabilii care se ocupa de acesta

\$ID_Fotomodel = \$_POST['ID_Fotomodel'];

```
$sql = "SELECT FP.ID_FisaPostului, FP.ID_Fotomodel, FP.ID_Responsabil, R.Nume, R.Prenume
FROM Fisa_Postului FP JOIN Responsabil R ON (FP.ID_Responsabil = R.ID_Responsabil)
WHERE FP.ID_Fotomodel ='$ID_Fotomodel'";
```

c) Un Parinte si toti copiii sai (Fotomodele) din BD-ul nostru

\$ID_Parinte = \$_POST['ID_Parinte'];

```
$sql = "SELECT P.ID_Parinte, F.ID_Fotomodel, F.Nume, F.Prenume, F.CNP
FROM Parinte P JOIN Fotomodel F ON (F.ID_Parinte = P.ID_Parinte)
WHERE P.ID_Parinte ='$ID_Parinte'";
```

d) Un Responsabil si toate Fotomodelele pe care le are in grija

\$ID_Responsabil = \$_POST['ID_Responsabil'];

```
$sql = "SELECT FP.ID_FisaPostului, FP.ID_Fotomodel, FP.ID_Responsabil, F.Nume, F.Prenume
FROM Fisa_Postului FP JOIN Fotomodel F ON (FP.ID_Fotomodel = F.ID_Fotomodel)
WHERE FP.ID_Responsabil ='$ID_Responsabil'";
```

e) Un Eveniment si toate Fotomodelele care participa la acesta

\$ID_Eveniment = \$_POST['ID_Eveniment'];

```
$sql = "SELECT FE.ID_FisaEveniment, FE.ID_Eveniment, FE.ID_Fotomodel, F.Nume, F.Prenume
FROM Fisa_Eveniment FE JOIN Fotomodel F ON (FE.ID_Fotomodel = F.ID_Fotomodel)
WHERE FE.ID_Eveniment ='$ID_Eveniment'";
```

f) O Vestimentatie si toate Evenimentele unde a fost utilizata

\$ID_Vestimentatie = \$_POST['ID_Vestimentatie'];

```
$sql = "SELECT V.ID_Vestimentatie, V.Nume_Vestimentatie, E.ID_Eveniment, E.Nume_Eveniment,
E.Importanta
FROM Vestimentatie V JOIN Eveniment E ON (E.ID_Vestimentatie = V.ID_Vestimentatie)
WHERE V.ID_Vestimentatie ='$ID_Vestimentatie'";
```

15. Interogari Complexe (4 exemple):

a) Lista primelor 3 Fotomodele in functie de inaltime

```
$sql = "SELECT F.ID_Fotomodel, F.Nume, F.Prenume, F.Inaltime  
FROM Fotomodel F  
WHERE F.Inaltime IN (SELECT distinct top 3 Inaltime from Fotomodel order by Inaltime DESC)  
ORDER BY Inaltime DESC";
```

b) Lista Fotomodelelor care au masa sub 50kg

```
$sql = "SELECT F.ID_Fotomodel, F.Nume, F.Prenume, F.Greutate  
FROM Fotomodel F  
WHERE F.Greutate < 50  
ORDER BY F.Greutate DESC";
```

c) Lista Fotomodelelor mai inalte decat media tuturor din BD

```
$sql = "SELECT F.ID_Fotomodel, F.Nume, F.Prenume, F.Inaltime  
FROM Fotomodel F  
WHERE F.Inaltime > ALL (SELECT AVG(Inaltime) FROM Fotomodel  
ORDER BY Inaltime DESC);
```

d) Lista Parintilor care nu au Fotomodele in BD-ul nostru

```
$sql = "SELECT P.ID_Parinte, P.Nume, P.Prenume, P.Adresa  
FROM Parinte P  
WHERE ID_Parinte NOT IN (SELECT P.ID_Parinte  
FROM Parinte P, Fotomodel F  
WHERE F.ID_Parinte = P.ID_Parinte and ID_Parinte IS NOT NULL );
```