USE ModelAGENCY

GO

--valid function of a field

--function that validates the no of hotels to be between 150-500

GO

CREATE FUNCTION validate\_no\_of\_hotels(@noOfHotels INT)

RETURNS INT AS

BEGIN

DECLARE @no INT

IF @noOfHotels >=150 and @noOfHotels <=500

SET @no = 1

ELSE

SET @no = 0

RETURN @no

END

--validate function of a field

--function that validates the no of flights to be between 2500-8000

GO

CREATE FUNCTION validate\_no\_of\_flights(@noOfFlights INT)

RETURNS INT AS

BEGIN

DECLARE @no INT

IF @noOfFlights >= 2500 and @noOfFlights <=8000

SET @no = 1

ELSE

SET @no = 0

RETURN @no

END

--procedure that adds data in the table

--adds new travel that includes no of hotels and no of flights instead of destination and hotel

GO

CREATE PROCEDURE ups\_add\_travel @Tid INT, @noOfHotels INT, @noOfFlights INT

AS

BEGIN

IF dbo.validate\_no\_of\_hotels(@noOfHotels) = 1 and dbo.validate\_no\_of\_flights(@noOfFlights) = 1

BEGIN

INSERT INTO Travel(Tid, Flight, Hotel) VALUES (@Tid, @noOfHotels, @noOfFlights)

PRINT 'Successfully added'

END

ELSE

BEGIN

PRINT 'Insert other parameters, this ones are wrong'

END

END

SET IDENTITY\_INSERT Travel ON

EXEC ups\_add\_travel 9,200,3600

EXEC ups\_add\_travel 10,370,7400

EXEC ups\_add\_travel 11,410,5700

SET IDENTITY\_INSERT Travel OFF

SELECT \* FROM Travel

--validate function of a field

--function that checks that the name is made out of letters

GO

CREATE FUNCTION validate\_mName(@mName VARCHAR(50))

RETURNS bit AS

BEGIN

DECLARE @b bit

IF @mName LIKE '[a-z]%[a-z]'

SET @b = 1

ELSE

SET @b = 0

RETURN @b

END

--validate function of a field

--function that checks that the surname is made out of letters

GO

CREATE FUNCTION validate\_mSurname(@mSurname VARCHAR(50))

RETURNS bit AS

BEGIN

DECLARE @b bit

IF @mSurname LIKE '[a-z]%[a-z]'

SET @b = 1

ELSE

SET @b = 0

RETURN @b

END

--validate function of a field

--function that checks that the height is between 150-200

GO

CREATE FUNCTION validate\_height(@height INT)

RETURNS INT AS

BEGIN

DECLARE @valid INT

IF @height >= 150 and @height <=200

SET @valid = 1

ELSE

SET @valid = 0

RETURN @valid

END

--validate function of a field

--function that checks that the weight is between 40-70

GO

CREATE FUNCTION validate\_weight(@weight INT)

RETURNS INT AS

BEGIN

DECLARE @valid INT

IF @weight >=40 and @weight <=70

SET @valid = 1

ELSE

SET @valid = 0

RETURN @valid

END

--validate function of a field

--function that checks that the nationality is made out of letters

GO

CREATE FUNCTION validate\_nationality(@nationality VARCHAR(50))

RETURNS bit AS

BEGIN

DECLARE @b bit

IF @nationality LIKE '[a-z]%[a-z]'

SET @b = 1

ELSE

SET @b = 0

RETURN @b

END

--procedure that adds data in the table

--adds the new model with all the dates validated

GO

CREATE PROCEDURE usp\_add\_model @Mid INT, @mName VARCHAR(50), @mSurname VARCHAR(50), @height INT, @weight INT, @nationality VARCHAR(50)

AS

BEGIN

IF dbo.validate\_mName(@mName) = 1 and dbo.validate\_mSurname(@mSurname) = 1 and dbo.validate\_height(@height) = 1 and dbo.validate\_weight(@weight) = 1 and dbo.validate\_nationality(@nationality) = 1

BEGIN

INSERT INTO Model(Mid, MName, MSurname, Height, Weight, Nationality) VALUES (@Mid, @mName, @mSurname, @height, @weight, @nationality)

PRINT 'Successfully added'

END

ELSE

BEGIN

PRINT 'Insert other parameters, this ones are wrong'

END

END

SET IDENTITY\_INSERT Model ON

EXEC usp\_add\_model 9,'Bendre','Sandra',175,43,'romanian'

EXEC usp\_add\_model 10,'Chisiu','Ioana',180,52,'arabian'

EXEC usp\_add\_model 11,'Gregor','Madeline',190,60,'american'

SET IDENTITY\_INSERT Model OFF

SELECT \* FROM Model

--view

GO

CREATE VIEW view\_model

AS

SELECT M.MName,M.MSurname

FROM Model M

INNER JOIN Payments p ON p.Mid = p.Pid

INNER JOIN Travel t ON t.Mid = t.Tid

INNER JOIN ModelContract mc ON mc.Mid = mc.Cid

INNER JOIN Contracts c ON c.Cid = mc.Cid AND c.MoneyOfC = 120000

SELECT \* FROM view\_model

--create table of triggers

CREATE TABLE Logs

(Lid INT NOT NULL, TriggerDate DATE, TriggerType VARCHAR(50), NameAffectedTable VARCHAR(50), NoAMDRows INT)

--create a copy of the model table

--here we make the changes

CREATE TABLE Model\_copy

(Mid INT NOT NULL, MName VARCHAR(50), MSurname VARCHAR(50), Height INT, Weight INT, Nationality VARCHAR(50))

--the trigger that adds in the same time models in tables

GO

CREATE TRIGGER add\_model ON Model

FOR INSERT

AS

BEGIN

INSERT INTO Model\_copy(Mid, MName, MSurname, Height, Weight, Nationality)

SELECT Mid,MName,MSurname,Height,Weight,Nationality

FROM inserted

INSERT INTO Logs(Lid, TriggerDate, TriggerType, NameAffectedTable, NoAMDRows) VALUES (1, GETDATE(), 'INSERT', 'Model', @@ROWCOUNT)

END

SET IDENTITY\_INSERT Model\_copy ON

SET IDENTITY\_INSERT Model\_copy OFF

SET IDENTITY\_INSERT Model ON

SET IDENTITY\_INSERT Model OFF

INSERT INTO Model(Mid,MName,MSurname,Height,Weight,Nationality) VALUES (12,'Graham','Johannah',184,44,'american')

INSERT INTO Model(Mid,MName,MSurname,Height,Weight,Nationality) VALUES (13,'Kokkinu','Elli',178,48,'greek')

SET IDENTITY\_INSERT Model OFF

SELECT \* FROM Model

SELECT \* FROM Model\_copy

SELECT \* FROM Logs

CREATE TABLE Model\_copy

(Mid INT NOT NULL, MName VARCHAR(50), MSurname VARCHAR(50), Height INT, Weight INT, Nationality VARCHAR(50))

--remove trigger

CREATE TRIGGER remove\_model ON Model

FOR DELETE

AS

BEGIN

DELETE FROM Model\_copy

WHERE Mid = (SELECT Mid FROM deleted)

INSERT INTO dbo.Logs(Lid, TriggerDate, TriggerType, NameAffectedTable, NoAMDRows) VALUES (1, GETDATE(), 'DELETE', 'Model', @@ROWCOUNT)

END

--update trigger

CREATE TRIGGER update\_model ON Model

FOR UPDATE

AS

BEGIN

UPDATE Model\_copy

SET MName = (SELECT MName FROM inserted), MSurname = (SELECT MSurname FROM inserted)

WHERE Mid = (SELECT Mid FROM inserted)

INSERT INTO dbo.Logs(Lid, TriggerDate, TriggerType, NameAffectedTable, NoAMDRows) VALUES (1, GETDATE(), 'UPDATE', 'Model', @@ROWCOUNT)

END

DELETE FROM Model\_copy

WHERE Mid = 11

UPDATE Model\_copy

SET MName = 'Yilmaz' , MSurname = 'Sophia'

WHERE Mid = 10

SELECT \* FROM Model

SELECT \* FROM Model\_copy

SELECT \* FROM Logs

--checks

SELECT \* FROM sys.sql\_modules

IF object\_id('Models') is not null

PRINT 'Present!'

ELSE

PRINT 'Not accounted for'

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = N'Models')

BEGIN

PRINT 'Table exists'

END