

1. Create a database named 'Evidenc3' with Data File and Log File those are to store in a default location including Name, Size, Max Size and File Growth.

2. Create a 3NF database using sample data below

Item	Color	Lot	Quantity	Price	Vat
Denim	Black, Blue	Lot-1	12	15000.00	15%
T-shirt	Red, Green, Blue	Lot-2	12	14500.00	15%

3. Create a CLUSTERED index
4. Create a view
5. Create a stored procedure to insert, Update and Delete data
6. Create a scalar UDF
7. Create a table values UDF
8. Create an AFTER trigger to restrict data insert

Create Database Evidenc3

ON

(

Name = 'Ev_data',

Filename = 'C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\Ev_03.mdf',

Size = 10 MB,

MaxSize = 1 GB,

FileGrowth = 10%

)

Log on

(

Name = 'Ev_Log',

Filename = 'C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS\MSSQL\DATA\Ev_03.ldf',

Size = 10 MB,

MaxSize = 1 GB,

FileGrowth = 10%

)

Go

USE Evidenc3

GO

Create Table item

(

Itemid Int primary key nonclustered,

itemname varchar(30) Not null,

price money not null,

vat float not null

)

Go

Create Table color

(

colorid Int Primary Key,

ColorName Varchar (30) not null

)

Go

Create table lot

(

Lotid int primary key,

lotname varchar (30) Not Null,

Quantity int not null,

Itemid Int references item(Itemid)

)

Go

Create Table itemColor

(

Itemid Int references item(Itemid),

Colorid int references Color (colorid),

primary key(Itemid, Colorid)

)

Go

Insert Into item

Values

```
(1, 'Denim', 15000.00, 0.15),  
(2, 'T-Shirt', 14500.00, 0.15)
```

Go

Insert Into color

Values

```
(1, 'Black'),  
(2, 'Blue'),  
(3, 'Red'),  
(4, 'Green')
```

Go

Insert Into lot

Values

```
( 1, 'Lot-1', 12, 1),  
( 2, 'Lot-2', 12, 2)
```

Go

Insert Into itemColor

Values

```
(1,1),  
(1,2),  
(2,3),  
(2,4),  
(2,2)
```

Go

Create Clustered Index Ix_Item_Name

ON Item (Itemname)

Go

Exec sp_helpindex 'item'

Go

Create View vItemDetails

AS

Select i.Itemid, i.itemname, c.ColorName, l.lotname, i.price, i.vat, l.Quantity

FROM item i

inner join lot l

ON i.Itemid = l.Itemid

Inner Join itemColor ic

ON l.Itemid = ic.Itemid

Inner join color c

ON ic.Colorid = c.colorid

GO

--Justify--

Select * from vItemDetails

Go

```
Create proc spInsertItem    @Itemid Int,  
                           @itemname varchar(30),  
                           @price money,  
                           @vat float
```

AS

Insert into item

Values (@Itemid, @itemname, @price, @vat)

Go

```
Create proc spUpdateItem    @Itemid Int,  
                           @itemname varchar(30),  
                           @price money,  
                           @vat float
```

AS

Update item

```
Set @itemname = itemname,  
    @price = price,
```

```

        @vat = Vat
        Where itemid = @Itemid
GO

Create proc spDeleteItem @Itemid Int
AS
Delete from item
Where Itemid = @Itemid
Go

CREATE FUNCTION fnGetNetPrice (@id INT)
RETURNS MONEY
AS
BEGIN
    DECLARE @netPrice MONEY
    SELECT @netPrice = price * (1+vat) FROM item
    WHERE itemid = @id

    RETURN @netPrice
END
GO
--Justify--
SELECT dbo.fnGetNetPrice(itemid) FROM item
GO

Create Function fnProductDetails (@itemname varchar(30))
Returns Table
AS
Return
Select i.Itemid, i.itemname, c.ColorName, l.lotname, i.price, i.vat, l.Quantity
FROM item i
Inner join lot l
ON i.Itemid = l.Itemid
Inner Join itemColor ic
ON l.Itemid = ic.Itemid
Inner join color c
ON ic.Colorid = c.colorid
Where @itemname = itemname
GO
--Justify--
SELECT * From fnProductDetails ('Denim')
GO
Create trigger trInsertItemRstriction
ON Item
For Insert
AS
Begin
    Declare @price Money, @ItemId int
    Select @price = price, @ItemId = Itemid from inserted
    If @price <=0
    Begin
        Raiserror ('Item price not to be zero', 11, 1)
        Rollback Transaction
    END
END
GO

```