Note: refer Assignment1 for table structures

1. Re-create the Customers and Orders tables, enhancing their definition with all primary and foreign keys constraints.
2. Using the ALTER TABLE statement, create an integrity constraint that limits the possible values of the quantity column in the Orders table to values between 1 and 30.
3. With the help of the corresponding system procedures and the Transact-SQL statements CREATE DEFAULT and CREATE RULE, create the alias data type “Western Countries”. The possible values for the new data type are CA(for California), WA( for Washington), OR( for Oregon), and NM( for New Mexico). The default value is CA. Finally, create a table called Regions with the columns City and Country using the new data type for the later.
4. Display all integrity constraints for the Orders table.
5. Delete the primary key of the Customers table. Why isn’t that working?
6. Delete the integrity constraint defined in Step-2.
7. Write a function that will return the age of the person given his or her date of birth.
8. Write a procedure that accepts name and data of birth of the student and inserts the data in the student table with the date computed. The SID should be largest sid in the table +1.

Query:

USE FSE

--1. Re-create the Customers and Orders tables, enhancing their definition with all primary and foreign keys constraints.

IF EXISTS (SELECT name FROM sys.tables

WHERE name = N'Customers')

DROP TABLE [dbo].[Customers]

GO

--Table level constraint creation

GO

Create table Customers(customerid char(5) not null,

CompanyName varchar(40) not null,

contactName char(30) null,

Address varchar(60) null,

City char(15) null,

Phone char(24) null,

Fax char(24) null

CONSTRAINT pk\_custid PRIMARY KEY(customerid),

CONSTRAINT fk\_CompName FOREIGN KEY(City) REFERENCES zone(City)

)

IF EXISTS (SELECT name FROM sys.tables

WHERE name = N'Orders')

DROP TABLE [dbo].[Orders]

GO

--Column level constraint creation

GO

Create table Orders(OrderId integer not null,

customerId char(5) not null PRIMARY KEY,

Orderdate datetime null,

Shippeddate datetime null,

Freight money null,

Shipname varchar(40) null,

Shipaddres varchar(60) null,

Quantity integer null)

--2. Using the ALTER TABLE statement, create an integrity constraint that limits the possible values of the quantity column in the Orders table

--to values between 1 and 30

GO

ALTER TABLE Orders

ADD CONSTRAINT ck\_quantity CHECK (Quantity >= 1 and Quantity <= 30)

GO

--3. With the help of the corresponding system procedures and the Transact-SQL statements CREATE DEFAULT and CREATE RULE, create the alias data

--type “Western Countries”. The possible values for the new data type are CA(for California), WA( for Washington), OR( for Oregon), and NM(for

--New Mexico). Default value is CA. Finally, create a table called Regions with the columns City & Country using new data type for later.

CREATE TYPE "WesterCountries"

FROM varchar(2) NOT NULL;

GO

CREATE RULE list\_rule

AS

@list IN ('CA', 'WA', 'OR', 'NM');

GO

sp\_bindrule 'list\_rule', 'WesterCountries'

GO

CREATE DEFAULT westCountry AS 'CA';

GO

GO

sp\_bindefault 'westCountry', 'WesterCountries';

GO

--4. Display all integrity constraints for the Orders table.

SELECT \*

FROM sys.indexes

WHERE object\_id = OBJECT\_ID('dbo.Orders')

SELECT \*

FROM sys.key\_constraints

WHERE object\_id = OBJECT\_ID('dbo.Orders')

SELECT \*

FROM sys.check\_constraints

WHERE object\_id = OBJECT\_ID('dbo.Orders')

SELECT \*

FROM sys.default\_constraints

WHERE object\_id = OBJECT\_ID('dbo.Orders')

SELECT \*

FROM sys.foreign\_keys

WHERE object\_id = OBJECT\_ID('dbo.Orders')

--5. Delete the primary key of the Customers table. Why isn’t that working?

GO

ALTER TABLE Persons

DROP CONSTRAINT PK\_Person;

GO

--6. Delete the integrity constraint defined in Step-2.

GO

ALTER TABLE dbo.Orders

DROP CONSTRAINT ck\_quantity;

GO

--7. Write a function that will return the age of the person given his or her date of birth.

GO

CREATE FUNCTION AGE(@DateOfBirth AS DATETIME)

RETURNS INT

AS

BEGIN

DECLARE @Years AS INT

DECLARE @BirthdayDate AS DATETIME

DECLARE @Age AS INT

--Calculate difference in years

SET @Years = DATEDIFF(YY,@DateOfBirth,GETDATE())

--Add years to DateOfBirth

SET @BirthdayDate = DATEADD(YY,@Years,@DateOfBirth)

--Subtract a year if birthday is after today

SET @Age = @Years -

CASE

WHEN @BirthdayDate>GETDATE() THEN 1

ELSE 0

END

--Return the result

RETURN @Age

END

GO

SELECT [dbo].[AGE] (

'1991-03-13')

GO

--8. Write a procedure that accepts name and data of birth of the student and inserts the data in the student table with the date computed.

--The SID should be largest sid in the table +1

CREATE TABLE Students(SID int IDENTITY(1,1),

Student\_name varchar(25),

Student\_DOB datetime,

Age int)

GO

CREATE PROC sp\_student @name varchar(25), @dob datetime AS

DECLARE @ageofStudent int = 0;

SET @ageofStudent = (SELECT [dbo].[AGE] (@dob));

INSERT INTO Students(Student\_name, Student\_DOB, Age) Values(@name, @dob, @ageofStudent)

GO

sp\_student 'Balaji Singh', '1991-03-13'

select \* from Students