**Table Creation Scripts:**

CREATE TABLE CourseList

(

CourseID int IDENTITY(1,1) not null,

CourseDescription varchar(500) not null,

CourseCost money not null,

CourseDurationYears decimal(2,1) not null,

Notes varchar (500) not null

)

CREATE TABLE Student\_Courses

(

StudentCoursesID int IDENTITY(1,1) not null,

StudentID int not null,

CourseID int not null,

CourseStartDate date not null,

CourseComplete date not null

)

CREATE TABLE StudentInformation

(

StudentID int IDENTITY(1,1) not null,

Title varchar (20) not null,

FirstName varchar(20) not null,

LastName varchar(20) not null,

Address1 varchar(50) not null,

Address2 varchar(50) null,

City varchar(50) not null,

County varchar(20) not null,

Zip varchar(10) not null,

Country varchar(20) not null,

Telephone varchar (15) not null,

AltTelephone varchar(15) null,

Email varchar(30) not null,

DOB date not null,

Enrolled date not null,

LastChanged date null,

UpdatedBy varchar (20) null

)

CREATE TABLE ContactType

(

ContactTypeID int IDENTITY(1,1)not null,

ContactType varchar(20) not null

)

CREATE TABLE Contacts

(

ContactID int IDENTITY(1,1) not null,

StudentID int not null,

ContactTypeID int not null,

ContactDate date not null,

EmployeeID int not null,

ContactDetails varchar(500) not null

)

CREATE TABLE Employees

(

EmployeeID int IDENTITY(1,1) not null,

EmployeeName varchar(50) not null,

EmployeePositionID int not null,

EmployeePassword varchar(20) not null,

Access varchar(20) not null

)

CREATE TABLE EmpJobPosition

(

EmployeePositionID int IDENTITY(1,1) not null,

EmployeePostition varchar(50) not null,

)

**Primary Key Scripts**

ALTER TABLE Contacts

ADD PRIMARY KEY (ContactID)

ALTER TABLE CourseList

ADD PRIMARY KEY (CourseID)

ALTER TABLE Student\_Courses

ADD PRIMARY KEY (StudentCoursesID)

ALTER TABLE StudentInformation

ADD PRIMARY KEY (StudentID)

ALTER TABLE ContactType

ADD PRIMARY KEY (ContactTypeID)

ALTER TABLE Employees

ADD PRIMARY KEY (EmployeeID)

ALTER TABLE EmpJobPosition

ADD PRIMARY KEY (EmployeePositionID)

**Foreign Key Scripts/Identity included in table Creation**

ALTER TABLE Contacts ADD

CONSTRAINT FK\_Contacts\_StudentInformation

FOREIGN KEY (StudentID) REFERENCES StudentInformation (StudentID)

ALTER TABLE Contacts ADD

CONSTRAINT FK\_Contacts\_ContactType

FOREIGN KEY (ContactTypeID) REFERENCES ContactType (ContactTypeID)

ALTER TABLE Contacts ADD

CONSTRAINT FK\_Contacts\_Employees

FOREIGN KEY (EmployeeID) REFERENCES Employees (EmployeeID)

ALTER TABLE Employees ADD

CONSTRAINT FK\_Employees\_EmpJobPosition

FOREIGN KEY (EmployeePositionID) REFERENCES EmpJobPosition (EmployeePositionID)

ALTER TABLE Student\_Courses ADD

CONSTRAINT FK\_StudentCourses\_StudentInformation

FOREIGN KEY (StudentID) REFERENCES StudentInformation (StudentID)

ALTER TABLE Student\_Courses ADD

CONSTRAINT FK\_StudentCourses\_CourseList

FOREIGN KEY (CourseID) REFERENCES CourseList (CourseID)

**Index Script**

CREATE INDEX LastName\_index

ON StudentInformation (LastName)

**Script for Questions 5 and 6**

ALTER TABLE StudentInformation

ADD CreatedDateTime timestamp

ALTER TABLE StudentInformation

DROP COLUMN AltTelephone

**Stored Procedure for 7a.**

CREATE PROCEDURE AddStudentRecord

@title varchar(20),

@firstname varchar(20),

@lastname varchar(20),

@address1 varchar(50),

@city varchar(50),

@county varchar(20),

@zip varchar(10),

@country varchar(20),

@telephone varchar(15),

@email varchar(30),

@dob date,

@enrolled date

AS

INSERT INTO StudentInformation

(Title,

FirstName,

LastName,

Address1,

City,

County,

Zip,

Country,

Telephone,

Email,

DOB,

Enrolled)

VALUES

(@title,

@firstname,

@lastname,

@address1,

@city,

@county,

@zip,

@country,

@telephone,

@email,

@dob,

@enrolled)

EXECUTE AddStudentRecord

'Mr',

'Doug',

'Frye',

'314 S. Market',

'St. Louis',

'Jefferson',

'63106',

'USA',

'314-265-3045',

'dougfrye@yahoo.com',

'2/05/1987',

'1.23.2014'

EXECUTE AddStudentRecord

'Mr',

'Tom',

'Jones',

'2287 Washington Ave',

'St. Louis',

'Franklin',

'63103',

'USA',

'314-589-3230',

'tomjones@yahoo.com',

'12/5/1982',

'1.22.2013'

EXECUTE AddStudentRecord

'Mrs',

'Kathy',

'Smith',

'2254 South Tenth St.',

'St. Louis',

'Jefferson',

'63104',

'USA',

'314-589-6211',

'kathysmith@yahoo.com',

'12/5/1922',

'1.22.2013'

**Stored Procedure for 7b**

CREATE PROCEDURE UpdateCounty @studentid int

AS

UPDATE StudentInformation

SET County = 'Bruin City'

WHERE StudentID = @studentid

EXECUTE UpdateCounty 4

**Stored Procedure for 7c**

CREATE PROCEDURE DeleteStudentRecord @studentid int

AS

DELETE FROM StudentInformation

WHERE StudentID = @studentid

EXECUTE DeleteStudentRecord 2

**Created View for 8**

CREATE VIEW StudentName AS

SELECT FirstName, LastName

FROM StudentInformation

**Scripts for Question 9**

CREATE FUNCTION TotalName (@StudentID int)

RETURNS varchar (255)

AS

BEGIN

RETURN (

SELECT RTRIM(FirstName) + RTRIM(LastName)

FROM StudentInformation

Where StudentID = @StudentID

)

END;

**Scripts for Question 10**

CREATE TRIGGER UpperCounty ON StudentInformation

FOR INSERT

AS

BEGIN

UPDATE StudentInformation

SET County = UPPER(County)

WHERE StudentID = (Select StudentID from inserted)

END