CREATE TABLE tbl\_Student

(

SSO INT PRIMARY KEY,

StudentName VARCHAR(50),

Gender CHAR(1) CHECK (Gender = 'M' OR Gender = 'F'),

Degree VARCHAR(10),

Email VARCHAR(50),

ContactNumber BIGINT,

DOB Date,

Passwd VARCHAR(50),

)

DROP TABLE tbl\_Student

INSERT INTO tbl\_Student VALUES(12426434,'Sumanth Gunda','M','Masters','sgnw8@mail.umkc.edu',8166823758,'1993-07-30','UMKC@Sum@nth4032')

INSERT INTO tbl\_Student VALUES(16186322,'Goutham Donthu','M','Masters','gd6d9@mail.umkc.edu',8166823463,'03-27-1992','gou')

INSERT INTO tbl\_Student VALUES(12426418,'Siddharth Gutha','M','Masters','sid@mail.umkc.edu',8166823718,'1993-07-18','sid')

INSERT INTO tbl\_Student VALUES(12426401,'Rajesh Jonnalagadda','M','Masters','raj@mail.umkc.edu',8166823701,'1993-07-01','UMKC@Sum@nth4032')

INSERT INTO tbl\_Student VALUES(12426402,'Umamaheshwara Rao Kothuri','M','Masters','uma@mail.umkc.edu',8166823702,'1993-07-02','uma')

INSERT INTO tbl\_Student VALUES(12426403,'Satish Chowdary Anumolu','M','Masters','sat@mail.umkc.edu',8166823703,'1993-07-03','sat')

INSERT INTO tbl\_Student VALUES(12426404,'Rishabh Bhojak','M','Masters','ris@mail.umkc.edu',8166823704,'1993-07-04','ris')

INSERT INTO tbl\_Student VALUES(12426405,'Bala Sai Teja Gaddam','M','Masters','bal@mail.umkc.edu',8166823705,'1993-07-05','bal')

INSERT INTO tbl\_Student VALUES(12426406,'Leela Naga Devi Gajula','F','Masters','lee@mail.umkc.edu',8166823706,'1993-07-06','lee')

INSERT INTO tbl\_Student VALUES(12426407,'Bhargava Gellaboina','M','Masters','bha@mail.umkc.edu',8166823707,'1993-07-07','bha')

INSERT INTO tbl\_Student VALUES(12426408,'Surya Prabha Ghanta','F','Masters','sur@mail.umkc.edu',8166823708,'1993-07-08','sur')

INSERT INTO tbl\_Student VALUES(12426409,'Gori Mehaboob','M','Masters','gor@mail.umkc.edu',8166823709,'1993-07-09','gor')

INSERT INTO tbl\_Student VALUES(12426410,'Raghu vital Gummadi','M','Masters','rag@mail.umkc.edu',8166823710,'1993-07-10','rag')

INSERT INTO tbl\_Student VALUES(12426411,'Haricharana Kanamarlapudi','F','Masters','har@mail.umkc.edu',8166823711,'1993-07-11','har')

INSERT INTO tbl\_Student VALUES(12426412,'Rajesh Kapa','M','Masters','kap@mail.umkc.edu',8166823712,'1993-07-12','kap')

INSERT INTO tbl\_Student VALUES(12426413,'Rahul Ponnada','M','Masters','rah@mail.umkc.edu',8166823713,'1993-07-13','rah')

INSERT INTO tbl\_Student VALUES(12426414,'Sandhya Sannapareddy','F','Masters','san@mail.umkc.edu',8166823714,'1993-07-14','san')

INSERT INTO tbl\_Student VALUES(12426415,'Anvesh Tummala','M','Masters','anv@mail.umkc.edu',8166823715,'1993-07-15','anv')

INSERT INTO tbl\_Student VALUES(12426416,'Pooja Agrawal','F','Masters','poo@mail.umkc.edu',8166823716,'1993-07-16','poo')

INSERT INTO tbl\_Student VALUES(12426417,'Jordan Larson','M','Masters','jor@mail.umkc.edu',8166823717,'1993-07-17','jor')

SELECT \* FROM tbl\_Student

CREATE TABLE tbl\_Instructor

(

InstructorId INT PRIMARY KEY,

InstructorName VARCHAR(50),

Gender CHAR(1) CHECK (Gender = 'M' OR Gender = 'F'),

Email VARCHAR(50),

ContactNumber BIGINT,

DOB Date,

Passwd VARCHAR(50),

)

SELECT \* FROM tbl\_Instructor

DROP TABLE tbl\_Instructor

INSERT INTO tbl\_Instructor VALUES(111,'BaekYoungChoi','F','choiby@mail.umkc.edu',9999999999,'1983-07-30','choi')

INSERT INTO tbl\_Instructor VALUES(112,'Lee','F','lee@mail.umkc.edu',9999999999,'1983-07-30','lee')

CREATE PROCEDURE usp\_Login(@Email VARCHAR(50),@Passwd VARCHAR(50))

AS

BEGIN

BEGIN TRY

IF EXISTS(SELECT Email FROM tbl\_Student WHERE Email = @Email)

BEGIN

IF EXISTS(SELECT Email,Passwd FROM tbl\_Student WHERE Email = @Email AND Passwd = @Passwd)

BEGIN

RETURN 1

END

ELSE

BEGIN

RETURN 2

END

END

ELSE IF EXISTS(SELECT Email FROM tbl\_Instructor WHERE Email = @Email)

BEGIN

IF EXISTS(SELECT Email,Passwd FROM tbl\_Instructor WHERE Email = @Email AND Passwd = @Passwd)

BEGIN

RETURN 3

END

ELSE

BEGIN

RETURN 4

END

END

ELSE

BEGIN

RETURN 5

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE PROCEDURE usp\_LoginService(@Name VARCHAR(50),@Passwd VARCHAR(50))

AS

BEGIN

BEGIN TRY

IF EXISTS(SELECT StudentName FROM tbl\_Student WHERE StudentName = @Name)

BEGIN

IF EXISTS(SELECT StudentName,Passwd FROM tbl\_Student WHERE StudentName = @Name AND Passwd = @Passwd)

BEGIN

RETURN 1

END

ELSE

BEGIN

RETURN 2

END

END

ELSE IF EXISTS(SELECT InstructorName FROM tbl\_Instructor WHERE InstructorName = @Name)

BEGIN

IF EXISTS(SELECT InstructorName,Passwd FROM tbl\_Instructor WHERE InstructorName = @Name AND Passwd = @Passwd)

BEGIN

RETURN 3

END

ELSE

BEGIN

RETURN 4

END

END

ELSE

BEGIN

RETURN 5

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE TABLE tbl\_SetAppointments

(

InstructorId INT CONSTRAINT srkey foreign key references tbl\_Instructor(InstructorId),

AppointmentDate DATE,

FromTime TIME(0),

ToTime TIME(0),

AppointmentDuration INT,

MaxAppointments INT,

PRIMARY KEY(InstructorId,AppointmentDate,FromTime,ToTime)

)

CREATE TABLE tbl\_AppointmentRequests

(

AppointmentId INT IDENTITY(1,1) PRIMARY KEY,

SSO INT CONSTRAINT srkey1 foreign key references tbl\_Student(SSO),

AppointmentDate DATE,

InstructorName VARCHAR(50),

AppointmentTime VARCHAR(50),

AppointmentType VARCHAR(50),

Description VARCHAR(200),

AppointmentStatus VARCHAR(20)

)

SELECT \* FROM tbl\_AppointmentRequests

DROP TABLE tbl\_AppointmentRequests

INSERT INTO tbl\_SetAppointments VALUES(111,'03-08-2015','05-08-2015','1:00','2:00',10,4)

SELECT \* FROM tbl\_SetAppointments

DROP TABLE tbl\_SetAppointments

CREATE PROCEDURE usp\_GetInstructorId(@Email VARCHAR(50),@InstructorId INT OUT)

AS

BEGIN

SET @InstructorId = 1

BEGIN TRY

SET @InstructorId = (SELECT InstructorId FROm tbl\_Instructor WHERE Email = @Email)

RETURN 1

END TRY

BEGIN CATCH

RETURN -1

END CATCH

END

CREATE PROCEDURE usp\_GetStudentId(@Email VARCHAR(50),@SSO INT OUT)

AS

BEGIN

SET @SSO = 1

BEGIN TRY

SET @SSO = (SELECT SSO FROm tbl\_Student WHERE Email = @Email)

RETURN 1

END TRY

BEGIN CATCH

RETURN -1

END CATCH

END

DROP PROCEDURE usp\_GetStudentId

CREATE PROCEDURE usp\_GetInstructorIdByname(@InstructorName VARCHAR(50),@InstructorId INT OUT)

AS

BEGIN

SET @InstructorId = 1

BEGIN TRY

SET @InstructorId = (SELECT InstructorId FROM tbl\_Instructor WHERE InstructorName = @InstructorName)

RETURN 1

END TRY

BEGIN CATCH

RETURN -1

END CATCH

END

CREATE PROCEDURE usp\_GetInstructorNameByEmail(@Email VARCHAR(50),@InstructorName VARCHAR(50) OUT)

AS

BEGIN

SET @InstructorName = 'Instructor'

BEGIN TRY

SET @InstructorName = (SELECT InstructorName FROM tbl\_Instructor WHERE Email = @Email)

RETURN 1

END TRY

BEGIN CATCH

RETURN -1

END CATCH

END

CREATE PROCEDURE usp\_SetAppointments(@InstructorId INT,@FromDate DATE,@ToDate DATE,@FromTime VARCHAR(20),@ToTime VARCHAR(20),@AppointmentDuration INT,@MaxAppointments INT)

AS

BEGIN

DECLARE @AppointmentDate DATE

DECLARE @Name VARCHAR(10)

SET @AppointmentDate = @FromDate

BEGIN TRY

WHILE(@AppointmentDate <= @ToDate)

BEGIN

SET @Name = (SELECT DATENAME(weekday,@AppointmentDate))

IF (@Name = 'Sunday' OR @Name = 'Saturday')

BEGIN

SET @AppointmentDate = (SELECT CAST((SELECT DATEADD(DAY,1,@AppointmentDate)) AS DATE))

END

ELSE

BEGIN

INSERT INTO tbl\_SetAppointments(InstructorId,AppointmentDate,FromTime,ToTime,AppointmentDuration,MaxAppointments) VALUES(@InstructorId,@AppointmentDate,@FromTime,@ToTime,@AppointmentDuration,@MaxAppointments)

SET @AppointmentDate = (SELECT CAST((SELECT DATEADD(DAY,1,@AppointmentDate)) AS DATE))

END

END

RETURN 1

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

SELECT DATENAME(weekday,'2015-03-07')

SELECT CAST((SELECT DATEADD(DAY,1,'2015-03-01')) AS DATE)

DROP PROCEDURE usp\_SetAppointments

SELECT AppointmentDate FROM tbl\_SetAppointments WHERE AppointmentDate = '2015-03-04' AND FromTime = '2:00' AND ToTime = '3:00'

SELECT t.SSO,t.StudentName,a.AppointmentDate,a.AppointmentTime,a.AppointmentType,a.Description,a.AppointmentStatus FROM tbl\_Student t INNER JOIN tbl\_AppointmentRequests a ON t.SSO = a.SSO AND a.InstructorName = 'BaekYoungChoi'

CREATE PROCEDURE AppRejAppointments(@AppointmentId INT,@StatusUpdate VARCHAR(50))

AS

BEGIN

BEGIN TRY

IF(@StatusUpdate = 'Approved')

BEGIN

UPDATE tbl\_AppointmentRequests SET AppointmentStatus = @StatusUpdate WHERE AppointmentId = @AppointmentId

RETURN 1

END

ELSE IF(@StatusUpdate = 'Rejected')

BEGIN

UPDATE tbl\_AppointmentRequests SET AppointmentStatus = @StatusUpdate WHERE AppointmentId = @AppointmentId

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE TABLE tbl\_Course

(

CourseId VARCHAR(20) PRIMARY KEY,

CourseName VARCHAR(100),

StartDate DATE,

EndDate DATE,

Credits INT,

CourseDepartment VARCHAR(100)

)

SELECT \* FROM tbl\_Course

DROP TABLE tbl\_Course

INSERT INTO tbl\_Course VALUES('CS551MT','Software Meathods and Tools','01-20-2015','05-20-2015',3,'Computer Science And Electrical'),('CS550SE','Advanced Software Engineering','01-20-2015','05-20-2015',3,'Computer Science And Electrical')

INSERT INTO tbl\_Course VALUES('CS551NA','Network Architecture','01-20-2015','05-20-2015',3,'Computer Science And Electrical')

CREATE TABLE tbl\_CourseAllotment

(

CourseId VARCHAR(20) CONSTRAINT srkey5 foreign key references tbl\_Course(CourseId),

InstructorId INT CONSTRAINT srkey6 foreign key references tbl\_Instructor(InstructorId)

)

INSERT INTO tbl\_CourseAllotment VALUES('CS551MT',111),('CS550SE',112)

INSERT INTO tbl\_CourseAllotment VALUES('CS551NA',111)

CREATE TABLE tbl\_CoursePlan

(

UniqueId INT PRIMARY KEY IDENTITY(1,1),

CourseId VARCHAR(20) CONSTRAINT srkey8 foreign key references tbl\_Course(CourseId),

TopicDate DATE,

TopicName VARCHAR(100),

Assignment VARCHAR(50),

AssignmentDeadline DATE

)

INSERT INTO tbl\_CoursePlan VALUES('CS551MT','01-20-2015','Course Introduction','No Silver Bullet','01-26-2015'),

('CS551MT','01-22-2015','Software Development Process and Activities','',NULL),

('CS551MT','01-27-2015','Lab 1','Assignment 1','02-02-2015'),

('CS551MT','01-29-2015','UML Modelling 1','',NULL),

('CS551MT','02-03-2015','Lab 2','Assignment 2','02-09-2015'),

('CS551MT','02-05-2015','UML Modelling 2','',NULL),

('CS551MT','02-10-2015','Lab 3','Assignment 3','02-23-2015'),

('CS551MT','02-12-2015','Software Architecture and Design 1','',NULL),

('CS551MT','02-17-2015','Software Architecture and Design 2','',NULL),

('CS551MT','02-19-2015','Arch Studio','',NULL),

('CS551MT','02-24-2015','Lab 4','Assignment 4','03-11-2015'),

('CS551MT','02-26-2015','IDE and Eclipse','',NULL),

('CS551MT','03-03-2015','Eclipse Plug ins 1','',NULL),

('CS551MT','03-05-2015','Guest Lecture','',NULL),

('CS551MT','03-10-2015','Eclipse Plug ins 2','',NULL),

('CS551MT','03-12-2015','Lab 5','Assignment 5','04-15-2015'),

('CS551MT','03-17-2015','MidTerm Review','',NULL),

('CS551MT','03-19-2015','Midterm','',NULL),

('CS551MT','03-31-2015','Assignment Demo','',NULL),

('CS551MT','04-02-2015','Emacs','',NULL),

('CS551MT','04-07-2015','Lab 6','',NULL),

('CS551MT','04-09-2015','Testing','',NULL),

('CS551MT','04-14-2015','Junit','',NULL),

('CS551MT','04-16-2015','Lab 7','Assignment 7','04-27-2015'),

('CS551MT','04-21-2015','Version Control','',NULL),

('CS551MT','04-23-2015','Sub Version','',NULL),

('CS551MT','04-28-2015','Lab 8','Assignment 8','05-04-2015'),

('CS551MT','04-30-2015','GIT','',NULL),

('CS551MT','05-05-2015','Lab 9','Assignment 9','05-08-2015'),

('CS551MT','05-07-2015','Course Review','',NULL),

('CS551MT','05-11-2015','Final Exam','',NULL)

DROP TABLE tbl\_CoursePlan

SELECT \* FROM tbl\_CoursePlan

CREATE TABLE tbl\_CourseEnrollment

(

SSO INT CONSTRAINT srkey11 foreign key references tbl\_Student(SSO),

CourseId VARCHAR(20) CONSTRAINT srkey12 foreign key references tbl\_Course(CourseId)

)

SELECT \* FROM tbl\_CourseEnrollment

SELECT CourseId FROm tbl\_CourseEnrollment WHERE SSO = 12426434

DROP TABLE tbl\_CourseEnrollment

INSERT INTO tbl\_CourseEnrollment VALUES(12426434,'CS551MT'),(16186322,'CS551MT'),(16186322,'CS551NA')

INSERT INTO tbl\_CourseEnrollment VALUES(12426401,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426402,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426403,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426404,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426405,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426406,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426407,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426408,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426409,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426410,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426411,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426412,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426413,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426414,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426415,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426416,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426417,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426417,'CS551MT')

INSERT INTO tbl\_CourseEnrollment VALUES(12426418,'CS551MT')

SELECT CourseId,Assignment,AssignmentDeadline FROM tbl\_CoursePlan WHERE Assignment!= '' AND AssignmentDeadline >= GETDATE() AND AssignmentDeadline < DATEADD(dd,14,GETDATE()) AND CourseId IN (SELECT CourseId FROM tbl\_CourseEnrollment WHERE SSO = 16186322)

SELECT CourseId FROM tbl\_CourseEnrollment WHERE SSO = 16186322

SELECT TopicDate,TopicName FROM tbl\_CoursePlan WHERE TopicDate BETWEEN GETDATE() AND DATEADD(dd,45,GETDATE()) AND (TopicName LIKE 'Midterm' OR TopicName LIKE '%Exam%') AND CourseId IN (SELECT CourseId FROM tbl\_CourseEnrollment WHERE SSO = 16186322)

SELECT tbl\_Course.CourseId, tbl\_Course.CourseName, tbl\_Course.StartDate, tbl\_Course.EndDate, tbl\_Course.Credits

FROM tbl\_Course INNER JOIN

tbl\_CourseAllotment ON tbl\_Course.CourseId = tbl\_CourseAllotment.CourseId AND tbl\_CourseAllotment.InstructorId = '111'

CREATE TABLE tbl\_Performance

(

SSO INT CONSTRAINT srkey21 foreign key references tbl\_Student(SSO),

CourseId VARCHAR(20) CONSTRAINT srkey22 foreign key references tbl\_Course(CourseId),

TopicName VARCHAR(50),

TotalMarks INT,

MarksObtained INT,

Percentage INT,

Comments VARCHAR(200)

)

SELECT \* FROM tbl\_Performance

DROP TABLE tbl\_Performance

SELECT TopicName FROM tbl\_CoursePlan WHERE CourseId ='CS551MT' AND TopicName LIKE '%Lab%'

SELECT Assignment FROM tbl\_CoursePlan WHERE CourseId ='CS551MT' AND Assignment != ''

SELECT \* FROM tbl\_Performance WHERE SSO = 16186322

SELECT COUNT(\*) FROm tbl\_Performance WHERE CourseId ='CS551MT' AND TopicName = 'Assignment 1' AND Percentage > 90 AND Percentage <=100

CREATE PROCEDURE usp\_AnalyzePerformance(@CourseId VARCHAR(10),@TopicName VARCHAR(50),@Top1 INT OUT,@Top2 INT OUT,@Top3 INT OUT,@Top4 INT OUT,@Top5 INT OUT)

AS

BEGIN

SET @Top1 = 0

SET @Top2 = 0

SET @Top3 = 0

SET @Top4 = 0

SET @Top5 = 0

BEGIN TRY

SET @Top1 = (SELECT COUNT(\*) FROm tbl\_Performance WHERE TopicName = @TopicName AND CourseId = @CourseId AND Percentage > 90 AND Percentage <=100)

SET @Top2 = (SELECT COUNT(\*) FROm tbl\_Performance WHERE TopicName = @TopicName AND CourseId = @CourseId AND Percentage > 80 AND Percentage <=90)

SET @Top3 = (SELECT COUNT(\*) FROm tbl\_Performance WHERE TopicName = @TopicName AND CourseId = @CourseId AND Percentage > 70 AND Percentage <=80)

SET @Top4 = (SELECT COUNT(\*) FROm tbl\_Performance WHERE TopicName = @TopicName AND CourseId = @CourseId AND Percentage > 60 AND Percentage <=70)

SET @Top5 = (SELECT COUNT(\*) FROm tbl\_Performance WHERE TopicName = @TopicName AND CourseId = @CourseId AND Percentage <=60)

RETURN 1

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE PROCEDURE usp\_GetStrength(@CourseId VARCHAR(50),@Strength INT OUT)

AS

BEGIN

SET @Strength = 1

BEGIN TRY

SET @Strength = (SELECT COUNT(\*) FROM tbl\_CourseEnrollment WHERE CourseId =@CourseId)

RETURN 1

END TRY

BEGIN CATCH

RETURN -1

END CATCH

END

CREATE PROCEDURE usp\_GetPosPer(@TopicName VARCHAR(50),@CourseId VARCHAR(50),@SSO INT,@Position INT OUT,@Percentage INT OUT)

AS

BEGIN

SET @Position = 1

SET @Percentage = 1

BEGIN TRY

SET @Position = (SELECT RowNumber FROM (SELECT ROW\_NUMBER() OVER(ORDER BY Percentage DESC) AS 'RowNumber',Percentage,SSO FROM tbl\_Performance WHERE TopicName = @TopicName AND CourseId=@CourseId) t WHERE SSO =@SSO)

SET @Percentage = (SELECT Percentage FROM (SELECT ROW\_NUMBER() OVER(ORDER BY Percentage DESC) AS 'RowNumber',Percentage,SSO FROM tbl\_Performance WHERE TopicName = @TopicName AND CourseId=@CourseId) t WHERE SSO =@SSO)

RETURN 1

END TRY

BEGIN CATCH

RETURN -1

END CATCH

END

SELECT RowNumber,Percentage FROM (SELECT ROW\_NUMBER() OVER(ORDER BY Percentage DESC) AS 'RowNumber',Percentage,SSO FROM tbl\_Performance WHERE TopicName = 'Assignment 3' AND CourseId='CS551MT') t WHERE SSO =16186322

SELECT TopicName,COUNT(\*) AS Count FROM tbl\_Performance WHERE CourseId = 'CS551MT' GROUP BY TopicName

SELECT COUNT(TopicName) FROM tbl\_Performance

SELECT COUNT(SSO) FROm tbl\_CourseEnrollment WHERE CourseId = 'CS551MT'

SELECT \* FROM tbl\_Performance

SELECT Count(\*) AS Count FROM (SELECT SSO,AVG(Percentage) AS Average FROM tbl\_Performance WHERe CourseId = 'CS551MT' GROUP By SSO) t WHERE Average > 90 AND Average <=100

CREATE PROCEDURE tbl\_AnalyzeCoursePerformance(@CourseId VARCHAR(50),@Top1 INT OUT,@Top2 INT OUT,@Top3 INT OUT,@Top4 INT OUT,@Top5 INT OUT)

AS

BEGIN

SET @Top1 = 0

SET @Top2 = 0

SET @Top3 = 0

SET @Top4 = 0

SET @Top5 = 0

BEGIN TRY

SET @Top1 = (SELECT Count(\*) AS Count FROM (SELECT SSO,AVG(Percentage) AS Average FROM tbl\_Performance WHERe CourseId = @CourseId GROUP By SSO) t WHERE Average > 90 AND Average <=100)

SET @Top2 = (SELECT Count(\*) AS Count FROM (SELECT SSO,AVG(Percentage) AS Average FROM tbl\_Performance WHERe CourseId = @CourseId GROUP By SSO) t WHERE Average > 80 AND Average <=90)

SET @Top3 = (SELECT Count(\*) AS Count FROM (SELECT SSO,AVG(Percentage) AS Average FROM tbl\_Performance WHERe CourseId = @CourseId GROUP By SSO) t WHERE Average > 70 AND Average <=80)

SET @Top4 = (SELECT Count(\*) AS Count FROM (SELECT SSO,AVG(Percentage) AS Average FROM tbl\_Performance WHERe CourseId = @CourseId GROUP By SSO) t WHERE Average > 60 AND Average <=70)

SET @Top5 = (SELECT Count(\*) AS Count FROM (SELECT SSO,AVG(Percentage) AS Average FROM tbl\_Performance WHERe CourseId = @CourseId GROUP By SSO) t WHERE Average < 60)

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE TABLE tbl\_Attendance

(

SSO INT CONSTRAINT srkey31 foreign key references tbl\_Student(SSO),

CourseId VARCHAR(20) CONSTRAINT srkey32 foreign key references tbl\_Course(CourseId),

AttendanceDate DATE,

AttendanceStatus VARCHAR(10)

)

SELECT \* FROM tbl\_Attendance

DROP TABLE tbl\_Attendance

SELECT SSO FROM tbl\_CourseEnrollment WHERE CourseId = 'CS551MT' AND SSO NOT IN(SELECT SSO FROM tbl\_Attendance WHERE CourseId='CS551MT' AND AttendanceDate = '2015-04-30')

CREATE TABLE tbl\_SetAttendance

(

Aid INT PRIMARY KEY IDENTITY(1,1),

CourseId VARCHAR(20) CONSTRAINT srkey33 foreign key references tbl\_Course(CourseId),

InstructorId INT CONSTRAINT srkey44 foreign key references tbl\_Instructor(InstructorId),

AttendanceDate DATE,

Latitude INT,

Longitude INT,

RandomCode VARCHAR(20),

StartTime VARCHAR(20),

EndTime VARCHAR(20)

)

DROP TABLE tbl\_SetAttendance

SELECT \* FROM tbl\_SetAttendance

CREATE PROCEDURE usp\_MarkAttendance(@SSO INT,@CourseId VARCHAR(20),@AttendanceDate DATE,@RandomCode VARCHAR(20),@Latitude INT,@Longitude INT)

AS

BEGIN

DECLARE @AttendanceStatus VARCHAR(20)

DECLARE @RandomCodeOriginal VARCHAR(20)

DECLARE @Latitude1 INT

DECLARE @Longitude1 INT

SET @AttendanceStatus = 'Present'

SET @RandomCodeOriginal = (SELECT RandomCode FROM tbl\_SetAttendance WHERE CourseId = @CourseId AND AttendanceDate = @AttendanceDate)

SET @Latitude1 = (SELECT Latitude FROM tbl\_SetAttendance WHERE CourseId = @CourseId AND AttendanceDate = @AttendanceDate)

SET @Longitude1 = (SELECT Longitude FROM tbl\_SetAttendance WHERE CourseId = @CourseId AND AttendanceDate = @AttendanceDate)

BEGIN TRY

IF(@RandomCodeOriginal = @RandomCode)

BEGIN

IF(@Latitude = @Latitude1 AND @Longitude = @Longitude1)

BEGIN

INSERT INTO tbl\_Attendance(SSO,CourseId,AttendanceDate,AttendanceStatus) VALUES(@SSO,@CourseId,@AttendanceDate,@AttendanceStatus)

RETURN 1

END

ELSE

BEGIN

RETURN 3

END

END

ELSE

BEGIN

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

ALTER PROCEDURE usp\_GetStartEndTime(@CourseId VARCHAR(20),@AttendanceDate DATE,@StartTime VARCHAR(20) OUT,@EndTime VARCHAR(20) OUT)

AS

BEGIN

SET @StartTime = ''

SET @EndTime = ''

BEGIN TRY

IF EXISTS(SELECT \* FROM tbl\_SetAttendance WHERE CourseId = @CourseId AND AttendanceDate = @AttendanceDate)

BEGIN

SET @StartTime = (SELECT StartTime FROm tbl\_SetAttendance WHERE CourseId = @CourseId AND AttendanceDate = @AttendanceDate)

SET @EndTime = (SELECT EndTime FROm tbl\_SetAttendance WHERE CourseId = @CourseId AND AttendanceDate = @AttendanceDate)

RETURN 1

END

ELSE

BEGIN

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -1

END CATCH

END

ALTER PROCEDURE usp\_AnalyzeAttendance(@CourseId VARCHAR(10),@AttendanceDate VARCHAR(50),@Present INT OUT,@Absent INT OUT)

AS

BEGIN

SET @Present = 0

SET @Absent = 0

BEGIN TRY

SET @Present = (SELECT COUNT(\*) FROm tbl\_Attendance WHERE AttendanceDate = @AttendanceDate AND CourseId = @CourseId AND AttendanceStatus = 'Present')

SET @Absent = (SELECT COUNT(\*) FROm tbl\_Attendance WHERE AttendanceDate = @AttendanceDate AND CourseId = @CourseId AND AttendanceStatus = 'Absent')

RETURN 1

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END