

## **Fall 2020 – CSE3055 Database Systems OSI (Online Self-Improvement) (Step-3)**

### **Project Description**

Online Self-Improvement is a system that allows students to take private lessons from the course they are registered in. This system organizes meetings according to the course the student is enrolled in and the branch of the teacher. The lessons taught in the meetings are handled under course topics. The time and duration of the meetings are determined by the admin and the duration and date of meetings can be updated by the admin. However, it can save the student's course exam grades in the system. This system can calculate the exam grade according to the number of correct and incorrect values in the exam when the exam grades are not entered. After finding the exam grades, It can calculate these grades according to the weighted exam average and calculate the main grade the student will receive at the end of the course.

### **Scope**

#### **Features Excluded from the System**

- We had a table named topic between the meeting and the course. We realized that our system would not support this table, and we added it as a multivalued attribute to our course table. We removed the Description value from the system.
- Considering that our system does not support many to many relationships between parent and student, we made the relationship between student and parent one to one. After doing this, the relationship between student and parent became specialized.

#### **Features Included to the System**

- In addition to the attributes in our Student table, we have added the payment the student paid to the education center and the date of registration.
- We added a table called exam between the student and the course, and this table keeps the number of questions in the exams.
- We were just holding an attribute named grade between our student and course tables. We started creating this attribute between student and exam, and additionally added the correctAnswers and FalseAnswers attributes. These attributes are used to calculate the grade when the exam grade is not entered. When adding the StudentExam procedure, the exam grade is calculated. Now, We keep these values as attributes between exam and Student.
- We started to get the gender and address values of instructors and students we entered.
- We started to keep the day the student completed the course and the main grade between the student and course tables.

- We will calculate the Maingrade by taking the average of the exams according to their weights.
- We have added a multivalued attribute named Instructor available time to our Instructor table. We have kept the times when instructors are available in the attribute of this value. We wrote a trigger that will update these values when Instructors sign up for a meeting. Inside this trigger, we added the instructor and the date to add to the meeting. After that, we deleted the date the Instructor was added to the meeting and the instructor from our table named InstructorAvailableTime.
- Inside the Instructor table, we have added an attribute named TotalHoursInLastWeek, which calculates how many hours the instructor has worked in the last week. We made these calculations in the view called InstructorInfo.

### **Processes supported by the system**

- Our system supports adding and deleting student data.
- Our system supports putting more than one topic in the courses.
- Our system supports adding exams.
- Our system supports adding and deleting Instructors.
- Our system supports assigning more than one available time to instructor.
- Our system supports multiple instructors to teach a course and one instructor to teach more than one course.
- Our system supports creating meetings between student and instructor.
- Our system supports adding students' parents.
- Our system supports multiple students in a course and a student in more than one course.
- Our system supports having more than one student in an exam and a student being able to take more than one exam.
- Our system supports the ability to delete instructor's available times.
- Our system supports the ability to delete the courses given by instructors.
- Our system supports being able to delete an existing meeting.
- Our system supports the ability to delete the exams the student has taken.
- Our system supports the ability to delete the topics of the courses.
- Our system supports updating the instructor's availableTime when the instructor is added to the meeting.
- Our system supports seeing the number of questions in the exams, the number of people taking the exam, course ID, course name, average grade, minimum grade and maximum grade in ExamInfo.
- Our system supports displaying the instructor's ID, firstName, LastName, mailAddress, PhoneNumber, zip, city, district, salary, TotalMinsInLast week in InstructorInfo.
- Our system supports displaying meeting's ID, studentID, studentName, instructor, instructor Name, course ID, courseName, Date, DurationInMins values in MeetingInfo.
- Our system supports seeing examID, studentID, studentName, numOfQuestions, correctAnswers, FalseAnswers, Grade, isAboveAvg values in studentExamInfo view in StudentExamInfo.
- Our system supports showing student ID, firstName, LastName, mailAddress, PhoneNumber, zip, city, district, Gender, Age, BirthDate, Payment, RegistrationDate, ParentName, ParentOccupation, ParentPhoneNumber in StudentInfo.

## **Processes not supported by the system**

- The system does not support more than one student, instructor to attend the meeting.
- The system does not allow more than one course to be processed in a meeting.

## **Business Rules and Constraints**

- People in the system must be either a student or an instructor
- Each meeting must have exactly one instructor and one student and one course.
- Each instructor must teach at least one course.
- Each meeting must provide exactly one course.
- Each student can have at most one parent.
- Each exam must have exactly one course.
- Each exam must have at least one student.

## **Data**

- Our client did not give us the data due to the privacy rights of the people, so we created the data ourselves.
- We used sites that generate random phone numbers and names while generating the data.

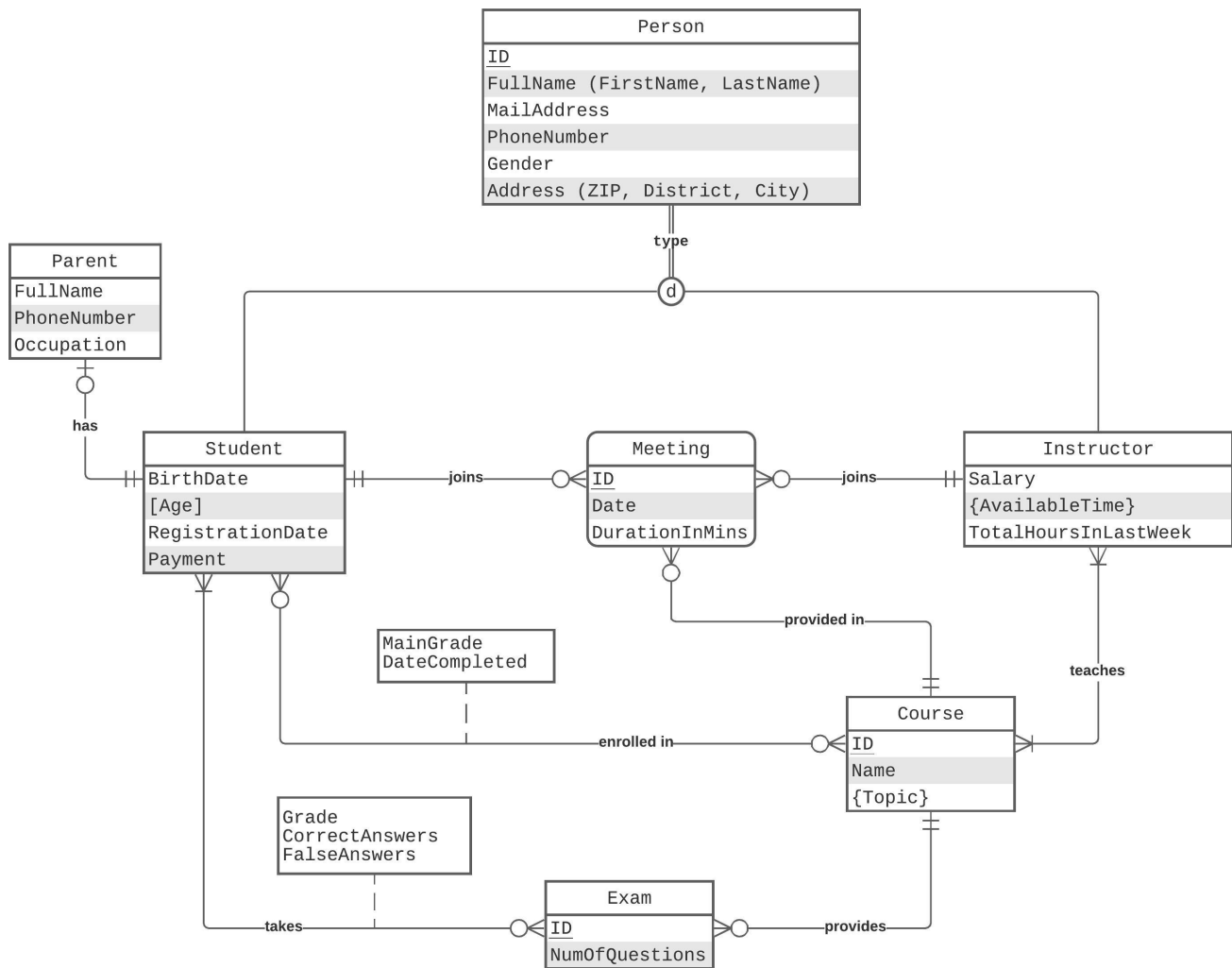
## **Functional Requirements**

- Users must be able to store course information in the system.
- Program allows users to create, delete or modify people in the system.
- Users can assign instructors and students to the meetings.
- Users can enter the grades of the students into the system.
- Program can list and show the important information in the database.

## **Non-functional Requirements**

- The system needs a server to keep the information in the database.
- The interface can only be accessed with the system's password.
- Users need to use an interface to interact with the database.
- The interface has a simple structure so that users can easily process data.
- New features can be easily integrated into the system.
- The database only keeps the necessary data init.

## Entity-Relationship Diagram



## TABLES

**Person:** We implemented disjoints for person and kept the common ones from instructor and student information in one place. They can access this information by personID.

1. ID SMALLINT
2. FirstName NVARCHAR(30)
3. LastName NVARCHAR(30)
4. MailAddress VARCHAR(50)
5. PhoneNumber VARCHAR(15)
6. Gender CHAR(1)
7. ZIP CHAR(5)
8. District NVARCHAR(15)
9. City NVARCHAR(15)

**Index:** The data in this table has been indexed according to firstName and LastName.

**Keys:** Each person created to be used in Instructor and student has an **ID** as a **primary key**.

**Student:** Specific information is kept about students who come to improvement themselves.

1. PersonID SMALLINT
2. BirthDate SMALLDATETIME
3. Age INT
4. RegistrationDate SMALLDATETIME
5. Payment SMALLMONEY

**Keys:** **PersonID** is the **primary key** for the student table, and also the **ID** and **foreign key** from the person table.

**Parent:** There is a parent information field for enrolling in a course for education, but it is at the student's initiative to provide this information but some must provide some information for this table

1. FullName NVARCHAR(50)
2. PhoneNumber VARCHAR(15)
3. Occupation NVARCHAR(30)
4. StudentID SMALLINT

**Keys:** **StudentID** is the **primary key** for the parent table, and also the **PersonID** and **foreign key** from the student table.

**Instructor:** Specific information about those who teach students at this training center is kept here.

1. PersonID SMALLINT
2. Salary SMALLMONEY

**Keys:** **PersonID** is the **primary key** for the student table, and also the **ID** and **foreign key** from the person table.

**InstructorAvailableTime:** Instructors are available for meetings on certain dates and times. Each time zone is entered into the system on an hourly basis.

1. InstructorID SMALLINT
2. StartTime SMALLDATETIME

**Keys:** InstructorID and StartTime are the **primary keys**. In addition, InstructorID is kept as the **personID foreign key** from the Instructor table.

**Course:** There are many courses in this training center. These courses are stored here with name.

1. ID SMALLINT
2. Name NVARCHAR(50)

**Index:** The data in this table has been indexed according to courseName.

**Keys:** Each course is created separately and the **courseID** is the **primary key**.

**CourseTopic:** Courses created can contain different topics for each meeting. These multiple values accessed from within the course are stored in the table named courseTopic.

1. CourseID SMALLINT
2. Name NVARCHAR(50)

**Keys:** CourseID and Name are the **primary keys**. In addition, CourseID is kept as the ID foreign key from the course table.

**StudentCourse :** Registered students can take different courses and these courses are kept here. The average of the exams in these courses is calculated as main grade. In addition, the date the student completes the course is kept datecompleted.

1. StudentID SMALLINT
2. CourseID SMALLINT
3. MainGrade FLOAT
4. DateCompleted SMALLDATETIME

**Index:** The data in this table has been indexed according to MainGrade.

**Keys:** StudentID and CourseID are the **primary keys**. StudentID is kept as the **ID foreign key** from the student table and CourseID is kept as the **ID foreign key** from the course table.

**InstructorCourse:** Instructors can teach multiple lectures, which we keep here for listing.

1. InstructorID SMALLINT
2. CourseID SMALLINT

**Keys:** InstructorID and CourseID are the **primary keys**. InstructorID is kept as the **PersonID foreign key** from the instructor table and CourseID is kept as the **ID foreign key** from the course table.

**Meeting:** The main demand of this education center is to create meetings and to list these meetings in the weekly . Therefore, this table created is between student, instructor and course. It has a specific date and time. All the meetings that occur are kept under this table.

1. ID SMALLINT
2. Date SMALLDATETIME
3. DurationInMins TINYINT

**Keys:** ID is the **primary keys**. StudentID is kept as the **PersonID foreign key** from the student table ,InstructorID is kept as the **PersonID foreign key** from the instuctor table and CourseID is kept as the **ID foreign key** from the course table.

**Exam:** Exams are held to measure whether the lessons are understood or not. The content of these exams is different, such as the number of questions.

1. ID SMALLINT
2. NumOfQuestions TINYINT
3. CourseID SMALLINT

**Keys:** ID is the **primary keys**. CourseID is kept as the **ID foreign key** from the course table.

**StudentExam:** We hold the exams students take. In this table, the number of correct and incorrect numbers made by the students in the exams is taken and the grade is calculated and all this information is kept.

1. StudentID SMALLINT
2. ExamID SMALLINT
3. Grade TINYINT
4. CorrectAnswers TINYINT
5. FalseAnswers TINYINT

**Keys:** ID is the **primary keys**. CourseID is kept as the **ID foreign key** from the course table.

**Keys:** StudentID is kept as the **PersonID foreign key** from the student table and ExamID is kept as the **ID foreign key** from the exam table.

**Uniques:** We did not want to have many restrictions by the company we work with, so we used less, but we can add it while preparing the user interface.

- In order not to re-register the person with the same **MailAddress** and **PhoneNumber** in the person table, both are unique.
- A course with the same name cannot be saved again, so the CourseName in the course table is unique.

**Identity:** When using identities, we set the data to start at a value.

- We kept that field as identity in **person, course, meeting and exam** tables that only keep data as ID, and we increased this value by 1 each.

**Check Constraints:** As I mentioned above, the company we work with did not want too many restrictions for data recording. Only necessary restrictions were made.

- When entering Person data **Email address** '@' followed by '.' must host.
- From the student course table, the **maingrade** must be a value between 0 or 100.
- Finally, the value of **durationMin** for the meeting can be entered as 50 at most.

**Default:** Some data do not have to be entered by the user and were asked not to appear as null.

- In the person table, the **zip** takes 34000 by default and the **city** takes Istanbul by default.
- If the **Student registration** date is not entered by the user, it is taken as the recorded date at that moment.

**Computed Columns:** For student age calculation, birth date data is taken from the student.



## Views

### StudentInfo

This view shows all the information about the student such as full name, mail address, phone number, gender, birth date, parent information, etc. by combining Person, Student and Parent tables registered in our database.

### Code

```
CREATE VIEW StudentInfo AS
SELECT p.ID, p.FirstName, p.LastName, p.MailAddress, p.PhoneNumber, COUNT(sc.CourseID) AS NumOfCourses,
       p.ZIP, p.City, p.District, p.Gender, s.Age, s.BirthDate, s.Payment, s.RegistrationDate,
       sp.FullName AS ParentName, sp.Occupation AS ParentOccupation, sp.PhoneNumber AS ParentPhoneNumber
FROM Person p,
     Student s LEFT JOIN Parent sp ON s.PersonID = sp.StudentID
LEFT JOIN
     (SELECT StudentID, CourseID FROM StudentCourse) AS sc ON sc.StudentID = s.PersonID
WHERE p.ID = s.PersonID
GROUP BY p.ID, p.FirstName, p.LastName, p.MailAddress, p.PhoneNumber,
       p.ZIP, p.City, p.District, p.Gender, s.Age, s.BirthDate, s.Payment, s.RegistrationDate,
       sp.FullName, sp.Occupation, sp.PhoneNumber
```

### Output

ID	FirstName	LastName	MailAddress	PhoneNumber	NumOfCourses	ZIP	City	District	Gender	Age	BirthDate	Payment	RegistrationDate	ParentName	ParentOccupation	ParentPhoneNumber
1	Fatma Özlem	Acar	fatma.acar@gmail.com	545-550-59880	2	34000	Kocaeli	NULL	F	22	1998-12-01 00:00:00	250.0000	2021-01-01 00:00:00	Ahmet Keskin	Teacher	505-550-3579
2	Gözde	Acarkan	gozde.acarkan@gmail.com	535-553-43084	3	34000	İstanbul	Kadıköy	F	20	2000-01-18 00:00:00	500.0000	2021-01-01 00:00:00	NULL	NULL	NULL
3	Halim	Salih	halim.salih@gmail.com	555-559-2898	2	34000	İstanbul	Üsküdar	M	22	1998-07-25 00:00:00	750.0000	2021-01-01 00:00:00	Mehmet Çelik	Attorney	535-550-9682
4	Bestami	Ağırbaş	bestami.agirbas@gmail.com	535-550-2731	0	41000	Kocaeli	NULL	M	22	1998-08-05 00:00:00	NULL	2021-01-01 00:00:00	Ayca Zilyagli	Astronaut	535-550-1122
5	Aykanat	Ağroğlu	aykanat.agiroglu@gmail.com	535-554-8892	2	41000	Kocaeli	NULL	M	21	1999-09-14 00:00:00	750.0000	2020-10-26 00:00:00	NULL	NULL	NULL
6	Şennur	Ağnar	sennur.agnar@gmail.com	505-550-1954	1	34000	İstanbul	Besikt.	F	24	1996-09-30 00:00:00	500.0000	2021-01-01 00:00:00	NULL	NULL	NULL
7	Tutkum	Yılmaz	tutkum.yilmaz@gmail.com	555-557-2723	1	34000	İstanbul	Üsküdar	F	23	1997-10-23 00:00:00	NULL	2021-01-01 00:00:00	Gül Karacao.	Thief	535-550-1345
8	Mügenur	Ahmet	mugenur.ahmet@gmail.com	545-556-0994	2	34000	İstanbul	NULL	F	19	2001-10-19 00:00:00	500.0000	2020-10-25 00:00:00	NULL	NULL	NULL
9	Sevinç	Ak	sevinc.ak@gmail.com	555-554-6195	1	34000	İstanbul	Kadıköy	F	18	2002-02-23 00:00:00	500.0000	2021-01-01 00:00:00	NULL	NULL	NULL
1.	Cihan	Akarpın.	cihan.akarpinar@gmail.com	545-555-2228	4	41000	Kocaeli	NULL	M	16	2004-03-19 00:00:00	NULL	2021-01-01 00:00:00	NULL	NULL	NULL
1.	Rafi	Akas	rafi.akas@gmail.com	505-551-5861	0	34000	İstanbul	Mecidi.	M	22	1998-06-25 00:00:00	750.0000	2021-01-01 00:00:00	NULL	NULL	NULL
1.	Mehmetcan	Akay	mehmetcan.akay@gmail.com	505-551-5061	2	41000	Kocaeli	NULL	M	22	1998-10-25 00:00:00	750.0000	2021-01-01 00:00:00	Fatma Turp	Painter	535-550-5346
1.	Nuhaydar	Akbilmez	nuhaydar.akbilmez@gmail.c..	505-551-5081	2	34000	İstanbul	Besikt.	F	22	1998-07-24 00:00:00	500.0000	2020-11-26 00:00:00	NULL	NULL	NULL
1.	Esine Müne.	Akca	esine.munever@gmail.com	535-553-5260	0	41000	Kocaeli	NULL	F	22	1998-07-11 00:00:00	500.0000	2021-01-01 00:00:00	NULL	NULL	NULL
1.	Servet	Akcagun.	servet.akcagunay@hotmail...	505-558-4638	3	34000	İstanbul	Mecidi.	M	22	1998-07-02 00:00:00	NULL	2021-01-01 00:00:00	NULL	NULL	NULL
1.	Çilem	Akçay	çilem.akçay@gmail.com	555-552-8187	0	34000	İstanbul	NULL	F	22	1998-07-25 00:00:00	NULL	2021-01-01 00:00:00	Ayşe Yavuz	Housewife	535-550-2365
1.	Recep Ali ..	Akdoğan	recepali.akdogan@gmail.com	535-552-3634	3	34000	İstanbul	Kadıköy	M	22	1998-07-14 00:00:00	NULL	2021-01-01 00:00:00	NULL	NULL	NULL
1.	Emre Ayberk	Akfarat	ayberk.akfarat@gmail.com	505-556-6147	0	34000	İstanbul	NULL	M	22	1998-12-05 00:00:00	NULL	2021-01-01 00:00:00	NULL	NULL	NULL
1.	Kerime Hac.	Akıllı	hacer.akilli@gmail.com	505-557-4463	0	34000	İstanbul	Kadıköy	F	22	1998-06-20 00:00:00	750.0000	2021-01-01 00:00:00	NULL	NULL	NULL
2.	Ercüment	..	ercument.akincilar@gmail...	555-554-6495	4	34000	Kocaeli	NULL	M	22	1998-07-25 00:00:00	750.0000	2021-01-01 00:00:00	NULL	NULL	NULL
2.	Sarper	Akış	sarper.akis@gmail.com	505-555-73029	1	34000	İstanbul	NULL	M	20	2000-07-25 00:00:00	500.0000	2021-01-01 00:00:00	NULL	NULL	NULL
2.	Berker	Akkiray	berker.akkiray@gmail.com	505-553-8020	2	34000	İstanbul	Besikt.	M	15	2005-07-25 00:00:00	500.0000	2021-01-01 00:00:00	NULL	NULL	NULL
2.	İcial	Akkoyun	icial.akkoyun@gmail.com	555-558-9865	1	34000	İstanbul	NULL	F	15	2005-06-18 00:00:00	500.0000	2021-01-01 00:00:00	NULL	NULL	NULL
2.	Lemis	Akküt	lemis.akkut@gmail.com	555-559-02946	0	34000	İstanbul	NULL	M	17	2003-07-25 00:00:00	250.0000	2021-01-01 00:00:00	NULL	NULL	NULL
2.	Ahmet Polat	Akılar C.	polat.corekci@gmail.com	505-558-1244	2	34000	İstanbul	NULL	M	18	2002-07-25 00:00:00	250.0000	2021-01-01 00:00:00	NULL	NULL	NULL
2.	Ata Kerem	Akman	kerem.akman@gmail.com	535-557-1730	0	34000	İstanbul	Besikt.	M	17	2003-01-17 00:00:00	250.0000	2021-01-01 00:00:00	NULL	NULL	NULL
2.	Ahmet Rasit	Akoğuz	rasit.akoгуz@gmail.com	535-550-9980	1	34000	İstanbul	Mecidi.	M	25	1995-10-25 00:00:00	500.0000	2021-01-01 00:00:00	Gaіp Sağlam	NULL	535-550-7243
2.	Ecem Hatice	Akova	hatice.akova@gmail.com	545-559-0960	1	34000	İstanbul	Kadıköy	F	25	1995-07-21 00:00:00	500.0000	2020-10-18 00:00:00	NULL	NULL	NULL
2.	Nuket	Aksan	nuket.aksan@gmail.com	535-559-5398	1	34000	İstanbul	Üsküdar	F	26	1994-07-14 00:00:00	500.0000	2021-01-01 00:00:00	NULL	NULL	NULL
3.	Senem	Akşeyim	senem.akşeyim@gmail.com	505-557-6464	1	34000	İstanbul	NULL	F	24	1996-09-28 00:00:00	750.0000	2021-01-01 00:00:00	NULL	NULL	NULL
3.	Aysen	Aksoy	aysen.aksoy@gmail.com	535-559-6791	1	34000	İstanbul	Kadıköy	F	17	2003-06-19 00:00:00	750.0000	2021-01-01 00:00:00	NULL	NULL	NULL
3.	Pekcan	Aksöz	pekcan.aksoz@gmail.com	555-552-8126	1	34000	İstanbul	NULL	M	23	1997-03-29 00:00:00	750.0000	2021-01-01 00:00:00	NULL	NULL	NULL

## InstructorInfo

This view shows all the information about the instructor such as full name, mail address, phone number, gender, salary etc. by combining Person and Instructor tables registered in our database. This view also shows the total minutes spent by the instructor in the meetings past week by summing up the durations of the meetings s/he attended in the last 7 days.

## Code

```
CREATE VIEW InstructorInfo AS
SELECT p.ID, p.FirstName, p.LastName, p.MailAddress, p.PhoneNumber, p.ZIP, p.City, p.District, p.Gender,
       i.Salary, SUM(DurationInMins) AS TotalMinsInLastWeek
FROM Person p, Instructor i LEFT JOIN
    (SELECT InstructorID, DurationInMins, [Date]
     FROM Meeting
     WHERE
        [Date] >= DATEADD(dd, ((DATEDIFF(dd, '01-01-1900', GETDATE()) / 7) * 7) - 7, '01-01-1900') AND
        [Date] <= DATEADD(dd, ((DATEDIFF(dd, '01-01-1900', GETDATE()) / 7) * 7) - 1, '01-01-1900')
    ) AS m ON i.PersonID = m.InstructorID
WHERE p.ID = i.PersonID
GROUP BY p.ID, p.FirstName, p.LastName, p.MailAddress, p.PhoneNumber, p.ZIP, p.City, p.District, p.Gender, i.Salary
```

## Output

ID	FirstName	LastName	MailAddress	PhoneNumber	ZIP	City	District	Gender	Salary	TotalMinsInLastWeek
51	Fatma Özlem	Acar	ftmacar@gmail.com	533-457-5858	34200	İstanbul	Bağcılar	F	5500.0000	NULL
52	Tunahan	Okatan	tok@gmail.com	588-789-5623	34000	İstanbul	Kadıköy	M	3700.0000	NULL
53	Murathan	Sakarya	mrsak@gmail.com	544-561-8947	34000	İstanbul	NULL	M	3300.0000	NULL
54	Ayçiçek	Saygın	say.aycicek@gmail.com	558-495-4878	34200	İstanbul	Bağcılar	F	3000.0000	NULL
55	Gencay	Ağaoğlu	gencay.agaoglu@gmail.com	507-925-0755	34180	İstanbul	Bahçelievler	M	NULL	NULL
56	Atakan	Yıldız	atakan@gmail.com	536-309-8325	34000	İstanbul	NULL	M	2400.0000	NULL
57	Dorukhan	Gürel	doruk58@gmail.com	544-415-9930	34180	İstanbul	Bahçelievler	NULL	3800.0000	NULL
58	Abbas	İncekara	abbas.ince@gmail.com	507-543-2777	34000	Kocaeli	Dilovası	M	5500.0000	NULL
59	Çiğdem	Bakkal	cigdem.bakkal.26@gmail.com	531-569-8270	41455	Kocaeli	Dilovası	F	4300.0000	NULL
60	Öğünç	Ayaz	ayaz34@hotmail.com	554-443-8291	41000	Kocaeli	NULL	M	5200.0000	NULL
61	Gürkan	Yıldırım	g.yldr@hotmail.com	545-371-4045	41400	Kocaeli	Gebze	M	6000.0000	NULL
62	Sezanur	İmamoğlu	seza@hotmail.com	504-559-8532	59850	Tekirdağ	Çorlu	F	NULL	NULL
63	Ahmet Utku	Kantarıcı	ahmetkntcr@hotmail.com	553-575-3817	59850	Tekirdağ	Çorlu	M	3000.0000	NULL
64	Farshad	Karacabey	farsahkaraca@hotmail.com	542-541-4702	34000	Tekirdağ	Çerkezköy	NULL	NULL	NULL
65	İsmail	Topcuoğlu	topcuoglu@hotmail.com	537-565-7156	59500	Tekirdağ	Çerkezköy	M	3800.0000	NULL
66	Dilşah	Pehlivan	dilsah.peh@hotmail.com	530-097-0914	34000	İstanbul	NULL	NULL	3800.0000	NULL
67	Perker	Sofyalıoğlu	perkersofya@hotmail.com	552-216-3596	NULL	Bursa	NULL	M	4300.0000	NULL
68	Mehmet	Taş	agalar@hotmail.com	538-081-5855	41400	Kocaeli	Gebze	M	5500.0000	NULL
69	Zeynep Doğa	Temizer	hesoes@hotmail.com	543-384-1426	34744	İstanbul	Kadıköy	F	NULL	NULL
70	Ramazan	Usluer	ramo@hotmail.com	541-889-2135	34180	İstanbul	Bahçelievler	M	NULL	NULL

## MeetingInfo

This view shows detailed informations of the meetings like meeting date and duration, student's and instructor's full names, their ID's, name and ID of the course the meeting belongs to.

## Code

```
CREATE VIEW MeetingInfo AS
SELECT m.ID,
       si.ID AS StudentID, si.FirstName + ' ' + si.LastName AS StudentName,
       ii.ID AS InstructorID, ii.FirstName + ' ' + ii.LastName AS InstructorName,
       c.ID AS CourseID, c.Name AS CourseName, m.[Date], m.DurationInMins
FROM Meeting m,
     (SELECT ID, FirstName, LastName FROM StudentInfo) AS si,
     (SELECT ID, FirstName, LastName FROM InstructorInfo) AS ii,
     (SELECT ID, Name FROM Course) AS c
WHERE m.StudentID = si.ID AND m.InstructorID = ii.ID AND m.CourseID = c.ID
```

## Output

ID	StudentID	StudentName	InstructorID	InstructorName	CourseID	CourseName	Date	DurationInMins
1	7	Tutkum Yılmaz	51	Fatma Özlem Acar	1	Math	2021-01-04 10:00:00	40
2	7	Tutkum Yılmaz	66	Dilşah Pehlivan	6	Cardistry	2021-01-04 14:00:00	30
3	2	Gözde Acarkan	53	Murathan Sakarya	3	Diction	2021-01-04 15:00:00	50
4	3	Halim Salih	52	Tunahan Okatan	2	Chess	2021-01-04 16:00:00	30
5	5	Aykanat Ağiroğlu	57	Dorukhan Gürel	8	Web Design	2021-01-05 16:00:00	40
6	6	Şennur Ağnar	60	Öğünç Ayaz	10	Spanish	2021-01-05 20:00:00	30
7	7	Tutkum Yılmaz	61	Gürkan Yıldırım	1	Math	2021-01-05 15:00:00	50
8	8	Mügenur Ahmet	63	Ahmet Utku Kantarcı	3	Diction	2021-01-05 16:00:00	30
9	10	Cihan Akarpınar	62	Sezanur İmamoğlu	5	Programming	2021-01-06 10:00:00	40
10	12	Mehmetcan Akay	59	Çiğdem Bakkal	9	English	2021-01-06 14:00:00	30
11	13	Nuhaydar Akbilmez	65	İsmail Topçuoğlu	1	Math	2021-01-07 17:00:00	50
12	13	Nuhaydar Akbilmez	52	Tunahan Okatan	2	Chess	2021-01-07 19:00:00	30
13	15	Servet Akçagunay	67	Perker Sofyalıoğlu	4	Basketball	2021-01-08 16:00:00	40
14	17	Recep Ali Samet Akdoğan	67	Perker Sofyalıoğlu	7	Video Editing	2021-01-09 20:00:00	30
15	20	Ercüment Akıncılar	70	Ramazan Usluer	9	English	2021-01-10 15:00:00	50
16	21	Sarper Akış	63	Ahmet Utku Kantarcı	3	Diction	2021-01-10 18:00:00	30
17	25	Ahmet Polat Aklar Çörekçi	57	Dorukhan Gürel	7	Video Editing	2021-01-11 18:00:00	40
18	25	Ahmet Polat Aklar Çörekçi	68	Mehmet Taş	8	Web Design	2021-01-11 20:00:00	30
19	27	Ahmet Raşit Akoğuz	62	Sezanur İmamoğlu	9	English	2021-01-12 17:00:00	50
20	28	Ecem Hatice Akova	69	Zeynep Doğa Temizer	10	Spanish	2021-01-12 19:00:00	30
21	29	Nüket Aksan	65	İsmail Topçuoğlu	1	Math	2021-01-12 16:00:00	40
22	30	Senem Aksevim	52	Tunahan Okatan	2	Chess	2021-01-13 20:00:00	30
23	31	Ayşen Aksoy	53	Murathan Sakarya	3	Diction	2021-01-13 15:00:00	50
24	32	Pekcan Aksöz	64	Farshad Karacabey	4	Basketball	2021-01-13 18:00:00	30
25	33	Bedirhan Lütfü Akşamoğlu	55	Gencay Ağaoğlu	5	Programming	2021-01-06 10:00:00	40
26	34	Semina Aktuna	66	Dilşah Pehlivan	6	Cardistry	2021-01-06 14:00:00	30
27	35	Eda Sena Akyıldız	67	Perker Sofyalıoğlu	7	Video Editing	2021-01-07 17:00:00	50
28	10	Cihan Akarpınar	57	Dorukhan Gürel	7	Video Editing	2021-01-07 19:00:00	30
29	10	Cihan Akarpınar	68	Mehmet Taş	8	Web Design	2021-01-08 16:00:00	40

## ExamInfo

This view shows the basic exam informations, name and ID of the exam's course, maximum and minimum grades by getting MAX and MIN from the StudentExam table's grade column. This view also calculates and shows the number of students taking the exam and average grade.

## Code

```
CREATE VIEW ExamInfo AS
SELECT
    e.ID, e.NumOfQuestions, COUNT(se.StudentID) AS NumOfStudents, e.CourseID, c.Name AS CourseName,
    CAST (AVG(se.Grade * 1.0) AS FLOAT) AS AvgGrade, MAX(se.Grade) AS MaxGrade, MIN(se.Grade) AS MinGrade
FROM Exam e
LEFT JOIN
    (SELECT StudentID, ExamID, Grade FROM StudentExam) AS se
    ON se.ExamID = e.ID,
    (SELECT ID, Name FROM Course) AS c
WHERE c.ID = e.CourseID
GROUP BY e.ID, e.NumOfQuestions, e.CourseID, c.Name
```

## Output

ID	NumOfQuestions	NumOfStudents	CourseID	CourseName	AvgGrade	MaxGrade	MinGrade
9	10	4	6	Cardistry	70	80	50
10	20	5	6	Cardistry	70	85	45
11	45	5	6	Cardistry	77.8	88	60
12	7	4	6	Cardistry	81.75	100	71
3	5	4	2	Chess	55	80	20
4	15	2	3	Diction	73	86	60
5	20	4	3	Diction	82.5	90	75
6	1	6	3	Diction	66.666666	100	0
17	12	6	9	English	72	100	25
1	10	7	1	Math	74.285714	100	50
2	20	7	1	Math	80	95	60
7	10	6	5	Programming	76.666666	90	60
8	30	4	5	Programming	84.5	96	76
18	15	5	10	Spanish	71.6	93	46
19	15	2	10	Spanish	46	86	6
20	20	3	10	Spanish	45	55	35
13	13	2	7	Video Editing	76.5	84	69
14	21	4	7	Video Editing	72.25	100	19
15	5	6	8	Web Design	73.333333	100	40

## StudentExamInfo

This view shows the ID and number of questions of the exam; correct and false answers, exam grade and also the name and ID of the student by combining StudentInfo view, ExamInfo view and StudentExam table.

## Code

```
CREATE VIEW StudentExamInfo AS
SELECT
    ei.ID AS ExamID, si.ID AS StudentID, si.FirstName + ' ' + si.LastName AS StudentName,
    ei.NumOfQuestions, se.CorrectAnswers, se.FalseAnswers, se.Grade,
    CAST(CASE WHEN se.Grade >= ei.AvgGrade THEN 1 ELSE 0 END AS BIT) AS IsAboveAvg
FROM StudentExam se,
    (SELECT ID, FirstName, LastName FROM StudentInfo) AS si,
    (SELECT * FROM ExamInfo) AS ei
WHERE se.StudentID = si.ID AND se.ExamID = ei.ID
GROUP BY ei.ID, si.ID, si.FirstName + ' ' + si.LastName, ei.NumOfQuestions,
    se.CorrectAnswers, se.FalseAnswers, se.Grade, ei.AvgGrade
```

## Output

ExamID	StudentID	StudentName	NumOfQuestions	CorrectAnswers	FalseAnswers	Grade	IsAboveAvg
1	1	Fatma Özlem Acar	10	8	2	80	1
1	7	Tutkum Yılmaz	10	7	1	70	0
1	13	Nuhaydar Akbilmez	10	7	3	70	0
1	20	Ercüment Akıncılar	10	10	0	100	1
1	29	Nüket Aksan	10	5	2	50	0
1	40	Kutlu Alibeyoğlu	10	6	2	60	0
1	49	Ahmet Kuru	10	9	1	90	1
2	1	Fatma Özlem Acar	20	12	0	60	0
2	7	Tutkum Yılmaz	20	15	2	75	0
2	13	Nuhaydar Akbilmez	20	14	4	70	0
2	20	Ercüment Akıncılar	20	17	1	85	1
2	29	Nüket Aksan	20	18	2	90	1
2	40	Kutlu Alibeyoğlu	20	17	3	85	1
2	49	Ahmet Kuru	20	19	1	95	1
3	1	Fatma Özlem Acar	5	3	1	60	1
3	20	Ercüment Akıncılar	5	4	1	80	1
3	30	Senem Aksevim	5	3	2	60	1
3	41	Nesibe Nurefşan Alkan	5	1	2	20	0
4	15	Servet Akçagunay	15	13	1	86	1
4	31	Ayşen Aksoy	15	9	4	60	0
5	2	Gözde Acarkan	20	18	2	90	1
5	8	Mügenur Ahmet	20	16	4	80	0

## Triggers

### TrgUpdateAvaliableTime

This trigger updates available times of an instructor when s/he joins a meeting. In this way, the database knows that s/he is no longer available for joining another meeting at that meeting's date and time.

### Code

```
CREATE TRIGGER TrgUpdateAvaliableTime
ON Meeting
AFTER INSERT

AS
BEGIN
    Declare @MeetingDate SMALLDATETIME ;
    Declare @InstructorID SMALLINT ;

    Select @MeetingDate=i.Date ,@InstructorID=i.InstructorID
    FROM inserted i

    Delete iat
    FROM InstructorAvailableTime iat
    WHERE @MeetingDate=iat.StartTime and @InstructorID=iat.InstructorID
END
```

## Before an Instructor Joining a Meeting

ID	StudentID	StudentName	InstructorID	InstructorName	CourseID	CourseName	Date	DurationInMins
1	7	Tutkum Yılmaz	51	Fatma Özlem Acar	1	Math	2021-01-04 10:00:00	40
33	40	Kutlu Alibeyoğlu	51	Fatma Özlem Acar	1	Math	2021-01-11 18:00:00	40

*Meetings of instructor 51*

InstructorID	StartTime
51	2021-01-04 09:00:00
51	2021-01-04 15:00:00
51	2021-01-04 19:00:00

*Available times of instructor 51*

## After an Instructor Joining a Meeting

ID	StudentID	StudentName	InstructorID	InstructorName	CourseID	CourseName	Date	DurationInMins
1	7	Tutkum Yılmaz	51	Fatma Özlem Acar	1	Math	2021-01-04 10:00:00	40
33	40	Kutlu Alibeyoğlu	51	Fatma Özlem Acar	1	Math	2021-01-11 18:00:00	40
41	27	Ahmet Raşit Akoğuz	51	Fatma Özlem Acar	1	Math	2021-01-04 15:00:00	35

*Meetings of instructor 51*

InstructorID	StartTime
51	2021-01-04 09:00:00
51	2021-01-04 19:00:00

*Available times of instructor 51*

## Procedures

### AddCourse

This procedure adds a new course with the given name.

### Code

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[AddCourse]
6  @CourseName NVARCHAR(50)
7  AS
8  BEGIN
9      SET NOCOUNT ON
10     INSERT INTO dbo.Course(Name)
11     VALUES (@CourseName);
12 END
13 GO
```

### Before

	ID	Name
1	1	Math
2	2	Chess
3	3	Diction
4	4	Basketball
5	5	Programming
6	6	Cardistry
7	7	Video Editing
8	8	Web Design
9	9	English
10	10	Spanish

### After

	ID	Name
1	1	Math
2	2	Chess
3	3	Diction
4	4	Basketball
5	5	Programming
6	6	Cardistry
7	7	Video Edit..
8	8	Web Design
9	9	English
10	10	Spanish
11	12	Literature



## AddCourseTopic

This procedure adds a new topic with given name to a course with given ID.

### Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[AddCourseTopic]
6     @CourseID SMALLINT,
7     @TopicName NVARCHAR(50)
8 AS
9 BEGIN
10     SET NOCOUNT ON
11     INSERT INTO dbo.CourseTopic(CourseID, Name)
12         VALUES (@CourseID, @TopicName)
13 END
14 GO
15
```

### Before

	CourseID	Name
1	1	Derivative
2	1	Geometry
3	1	Integral
4	1	Series

### After

	CourseID	Name
1	1	Derivative
2	1	Geometry
3	1	Integral
4	1	Limits
5	1	Series

## AddExam

This procedure adds a new exam to a course with the given number of questions.

## Code

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[AddExam]
6      @CourseID SMALLINT,
7      @NumOfQuestions TINYINT = NULL
8  AS
9  BEGIN
10     SET NOCOUNT ON
11     INSERT INTO dbo.Exam(CourseID,NumOfQuestions)
12         VALUES(@CourseID,@NumOfQuestions);
13 END
14 GO
15
```

## Before

	ID	NumOfQuestions	CourseID
1	1	10	1
2	2	20	1
3	3	5	2
4	4	15	3
5	5	20	3
6	6	1	3
7	7	10	5
8	8	30	5
9	9	10	6
10	10	20	6
11	11	45	6
12	12	7	6
13	13	13	7
14	14	21	7
15	15	5	8
16	16	4	8
17	17	12	9
18	18	15	10
19	19	15	10
20	20	20	10

## After

	ID	NumOfQuestions	CourseID
1	1	10	1
2	2	20	1
3	21	25	1
4	3	5	2
5	4	15	3
6	5	20	3
7	6	1	3
8	7	10	5
9	8	30	5
10	9	10	6
11	10	20	6
12	11	45	6
13	12	7	6
14	13	13	7
15	14	21	7
16	15	5	8
17	16	4	8
18	17	12	9
19	18	15	10
20	19	15	10
21	20	20	10

## AddInstructor

This procedure adds a new instructor to the system, creates a row in Person table and Instructor table.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[AddInstructor]
6     @FirstName NVARCHAR(30),
7     @LastName NVARCHAR(30),
8     @MailAddress VARCHAR(30),
9     @PhoneNumber VARCHAR(15),
10    @Gender CHAR(1),
11    @ZIP CHAR(5) = '34000',
12    @District NVARCHAR(15) = NULL,
13    @City NVARCHAR(15) = N'İstanbul',
14    @Salary SMALLINT = NULL
15 AS
16 BEGIN
17     SET NOCOUNT ON
18     INSERT INTO dbo.Person(FirstName, LastName, MailAddress, PhoneNumber, Gender, ZIP, District, City)
19     VALUES (@FirstName, @LastName, @MailAddress, @PhoneNumber, @Gender, @ZIP, @District, @City);
20     INSERT INTO dbo.Instructor(PersonID, Salary)
21     VALUES (SCOPE_IDENTITY(), CAST(@Salary AS SMALLMONEY));
22 END
23 GO
24
```

## Before

	ID	FirstName	LastName	MailAddress	PhoneNumber	ZIP	City	District	Gender	Salary	TotalMinsInLastWeek
1	51	Fatma Özlem	Acar	ftmacar@gmail.com	533-457-5858	34200	İstanbul	Bağcılar	F	5500.0000	NULL
2	52	Tunahan	Okatan	tok@gmail.com	588-789-5623	34000	İstanbul	Kadıköy	M	3700.0000	NULL
3	53	Murathan	Sakarya	mrtak@gmail.com	544-561-8947	34000	İstanbul	NULL	M	3300.0000	NULL
4	54	Ayçicek	Saygın	say.aycicek@gmail.com	558-495-4878	34200	İstanbul	Bağcılar	F	3000.0000	NULL
5	55	Gencay	Ağaoğlu	gencay.agaoglu@gmail.com	507-925-0755	34180	İstanbul	Bahçelievler	M	NULL	NULL
6	56	Atakan	Yıldız	atakan@gmail.com	536-309-8325	34000	İstanbul	NULL	M	2400.0000	NULL
7	57	Dorukhan	Gürel	doruk58@gmail.com	544-415-9930	34180	İstanbul	Bahçelievler	NULL	3800.0000	NULL
8	58	Abbas	İnceka...	abbas.ince@gmail.com	507-543-2777	34000	Kocaeli	Dilovası	M	5500.0000	NULL
9	59	Çiğdem	Bakkal	cigdem.bakkal.26@gmail...	531-569-8270	41455	Kocaeli	Dilovası	F	4300.0000	NULL
1..	60	Öğünç	Ayaz	ayaz34@hotmail.com	554-443-8291	41000	Kocaeli	NULL	M	5200.0000	NULL
1..	61	Gürkan	Yıldır...	g.yldr@hotmail.com	545-371-4045	41400	Kocaeli	Gebze	M	6000.0000	NULL
1..	62	Sezanur	İmamoğ...	seza@hotmail.com	504-559-8532	59850	Tekirdağ	Çorlu	F	NULL	NULL
1..	63	Ahmet Utku	Kantar...	ahmetkntcr@hotmail.com	553-575-3817	59850	Tekirdağ	Çorlu	M	3000.0000	NULL
1..	64	Farshad	Karaca...	farsahkaraca@hotmail.com	542-541-4702	34000	Tekirdağ	Çerkezköy	NULL	NULL	NULL
1..	65	İsmail	Topçuo...	topcuoglu@hotmail.com	537-565-7156	59500	Tekirdağ	Çerkezköy	M	3800.0000	NULL
1..	66	Dilşah	Pehliv...	dilsah.peh@hotmail.com	530-097-0914	34000	İstanbul	NULL	NULL	3800.0000	NULL
1..	67	Perker	Sofyal...	perkersofya@hotmail.com	552-216-3596	NULL	Bursa	NULL	M	4300.0000	NULL
1..	68	Mehmet	Taş	agalar@hotmail.com	538-081-5855	41400	Kocaeli	Gebze	M	5500.0000	NULL
1..	69	Zeynep Doğa	Temizer	hesoes@hotmail.com	543-384-1426	34744	İstanbul	Kadıköy	F	NULL	NULL
2..	70	Ramazan	Usiuer	ramo@hotmail.com	541-889-2135	34180	İstanbul	Bahçelievler	M	NULL	NULL

## After

	ID	FirstName	LastName	MailAddress	PhoneNumber	ZIP	City	District	Gender	Salary	TotalMinsInLastWeek
1	51	Fatma Özlem	Acar	ftmacar@gmail.com	533-457-5858	34200	İstanbul	Bağcılar	F	5500.0000	NULL
2	52	Tunahan	Okatan	tok@gmail.com	588-789-5623	34000	İstanbul	Kadıköy	M	3700.0000	NULL
3	53	Murathan	Sakarya	mrtsak@gmail.com	544-561-8947	34000	İstanbul	NULL	M	3300.0000	NULL
4	54	Ayçiçek	Saygın	say.aycicek@gmail.com	558-495-4878	34200	İstanbul	Bağcılar	F	3000.0000	NULL
5	55	Gencay	Ağaoğlu	gencay.agaoglu@gmail.com	507-925-0755	34180	İstanbul	Bahçelievler	M	NULL	NULL
6	56	Atakan	Yıldız	atakan@gmail.com	536-309-8325	34000	İstanbul	NULL	M	2400.0000	NULL
7	57	Dorukhan	Gürel	doruk58@gmail.com	544-415-9930	34180	İstanbul	Bahçelievler	NULL	3800.0000	NULL
8	58	Abbas	İncekara	abbas.ince@gmail.com	507-543-2777	34000	Kocaeli	Dilovası	M	5500.0000	NULL
9	59	Çiğdem	Bakkal	cigdem.bakkal.26@gmail.com	531-569-8270	41455	Kocaeli	Dilovası	F	4300.0000	NULL
10	60	Öğünç	Ayaz	ayaz34@hotmail.com	554-443-8291	41000	Kocaeli	NULL	M	5200.0000	NULL
11	61	Gürkan	Yıldırım	g.yldrm@hotmail.com	545-371-4045	41400	Kocaeli	Gebze	M	6000.0000	NULL
12	62	Sezanur	İmamoğlu	seza@hotmail.com	504-559-8532	59850	Tekirdağ	Çorlu	F	NULL	NULL
13	63	Ahmet Utku	Kantaracı	ahmetkntrc@hotmail.com	553-575-3817	59850	Tekirdağ	Çorlu	M	3000.0000	NULL
14	64	Farshad	Karacabey	farsahkaraca@hotmail.com	542-541-4702	34000	Tekirdağ	Çerkezköy	NULL	NULL	NULL
15	65	İsmail	Topçuoğlu	topcuoglu@hotmail.com	537-565-7156	59500	Tekirdağ	Çerkezköy	M	3800.0000	NULL
16	66	Dilşah	Pehlivan	dilsah.peh@hotmail.com	530-097-0914	34000	İstanbul	NULL	NULL	3800.0000	NULL
17	67	Perker	Sofyalıoğlu	perkersofya@hotmail.com	552-216-3596	NULL	Bursa	NULL	M	4300.0000	NULL
18	68	Mehmet	Taş	agalar@hotmail.com	538-081-5855	41400	Kocaeli	Gebze	M	5500.0000	NULL
19	69	Zeynep Doğa	Temizer	hesoes@hotmail.com	543-384-1426	34744	İstanbul	Kadıköy	F	NULL	NULL
20	70	Ramazan	Usluer	ramo@hotmail.com	541-889-2135	34180	İstanbul	Bahçelievler	M	NULL	NULL
21	71	Kerem	Akıncı	k.akinci@gmail.com	533-427-5812	34200	İstanbul	Bağcılar	M	6000.0000	NULL

## AddInstructorAvailableTime

This procedure adds an available time for the instructor.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[AddInstructorAvailableTime]
6     @InstructorID SMALLINT,
7     @StartTime NVARCHAR(50)
8 AS
9 BEGIN
10     SET NOCOUNT ON
11     INSERT INTO InstructorAvailableTime(InstructorID,StartTime)
12     VALUES ( @InstructorID, @StartTime);
13 END
14 GO
15
```

## Before

	InstructorID	StartTime
1	51	2021-01-04 09:00:00
2	51	2021-01-04 19:00:00
3	52	2021-01-05 09:00:00
4	52	2021-01-05 15:00:00
5	52	2021-01-05 19:00:00
6	53	2021-01-06 09:00:00
7	53	2021-01-06 15:00:00
8	53	2021-01-06 19:00:00
9	54	2021-01-07 09:00:00
10	54	2021-01-07 15:00:00
11	54	2021-01-07 19:00:00
12	55	2021-01-08 09:00:00
13	55	2021-01-08 15:00:00
14	55	2021-01-08 19:00:00
15	56	2021-01-09 09:00:00
16	56	2021-01-09 15:00:00
17	56	2021-01-09 19:00:00
18	57	2021-01-10 09:00:00
19	57	2021-01-10 15:00:00
20	57	2021-01-10 19:00:00
21	58	2021-01-11 12:00:00
22	58	2021-01-11 14:00:00
23	58	2021-01-11 18:00:00

## After

	InstructorID	StartTime
1	51	2020-01-05 14:00:00
2	51	2021-01-04 09:00:00
3	51	2021-01-04 19:00:00
4	52	2021-01-05 09:00:00
5	52	2021-01-05 15:00:00
6	52	2021-01-05 19:00:00
7	53	2021-01-06 09:00:00
8	53	2021-01-06 15:00:00
9	53	2021-01-06 19:00:00
10	54	2021-01-07 09:00:00
11	54	2021-01-07 15:00:00
12	54	2021-01-07 19:00:00
13	55	2021-01-08 09:00:00
14	55	2021-01-08 15:00:00
15	55	2021-01-08 19:00:00
16	56	2021-01-09 09:00:00
17	56	2021-01-09 15:00:00
18	56	2021-01-09 19:00:00
19	57	2021-01-10 09:00:00
20	57	2021-01-10 15:00:00
21	57	2021-01-10 19:00:00
22	58	2021-01-11 12:00:00
23	58	2021-01-11 14:00:00

## AddInstructorCourse

This procedure assigns an instructor to a course.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[AddInstructorCourse]
6     @InstructorID SMALLINT,
7     @CourseID SMALLINT
8 AS
9 BEGIN
10     SET NOCOUNT ON
11     INSERT INTO InstructorCourse(InstructorID,CourseID)
12     VALUES ( @InstructorID, @CourseID);
13 END
14 GO
15
```

## Before

	InstructorID	CourseID
1	51	1
2	51	2
3	51	9
4	52	2
5	53	3
6	54	4
7	55	1
8	55	5
9	56	6
10	57	7
11	57	8
12	59	9
13	60	9
14	60	10
15	61	1
16	62	1
17	62	2
18	62	5
19	62	7
20	62	8
21	62	9
22	63	3
23	64	4

## After

	InstructorID	CourseID
1	51	1
2	51	2
3	51	7
4	51	9
5	52	2
6	53	3
7	54	4
8	55	1
9	55	5
10	56	6
11	57	7
12	57	8
13	59	9
14	60	9
15	60	10
16	61	1
17	62	1
18	62	2
19	62	5
20	62	7
21	62	8
22	62	9
23	63	3

## AddMeeting

This procedure adds a new meeting with the given meeting information, ID's of the student, instructor and the course.

## Code

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[AddMeeting]
6  @Date VARCHAR(15),
7  @DurationInMins TINYINT,
8  @StudentID SMALLINT,
9  @InstructorID SMALLINT,
10 @CourseID SMALLINT
11 AS
12 BEGIN
13     SET NOCOUNT ON
14     INSERT INTO dbo.Meeting(Date, DurationInMins, StudentID, InstructorID, CourseID)
15     VALUES (CAST(@Date AS SMALLDATETIME), @DurationInMins, @StudentID, @InstructorID, @CourseID) ;
16 END
17 GO
18
```

Before

	ID	StudentID	StudentName	InstructorID	InstructorName	CourseID	CourseName	Date	DurationInMins
19	19	27	Ahmet Raşit Akoğuz	62	Sezanur İmamoğlu	9	English	2021-01-12 17:00:00	50
20	20	28	Ecem Hatice Akova	69	Zeynep Doğa Temizer	10	Spanish	2021-01-12 19:00:00	30
21	21	29	Nüket Aksan	65	İsmail Topçuoğlu	1	Math	2021-01-12 16:00:00	40
22	22	30	Senem Aksevim	52	Tunahan Okatan	2	Chess	2021-01-13 20:00:00	30
23	23	31	Ayşen Aksoy	53	Murathan Sakarya	3	Diction	2021-01-13 15:00:00	50
24	24	32	Pekcan Aksöz	64	Farshad Karacabey	4	Basketball	2021-01-13 18:00:00	30
25	25	33	Bedirhan Lütfü Akşamoğlu	55	Gencay Ağaoğlu	5	Programming	2021-01-06 10:00:00	40
26	26	34	Semina Aktuna	66	Dilşah Pehlivan	6	Cardistry	2021-01-06 14:00:00	30
27	27	35	Eda Sena Akyıldız	67	Perker Sofyalıoğlu	7	Video Editing	2021-01-07 17:00:00	50
28	28	10	Cihan Akarpınar	57	Dorukhan Gürel	7	Video Editing	2021-01-07 19:00:00	30
29	29	10	Cihan Akarpınar	68	Mehmet Taş	8	Web Design	2021-01-08 16:00:00	40
30	30	36	Müyesser Akyıldırım	68	Mehmet Taş	8	Web Design	2021-01-09 20:00:00	30
31	31	37	Selinti Al	59	Çiğdem Bakkal	9	English	2021-01-10 15:00:00	50
32	32	38	Bahar Özlem Albaş	69	Zeynep Doğa Temizer	10	Spanish	2021-01-10 18:00:00	30
33	33	40	Kutlu Alibeyoğlu	51	Fatma Özlem Acar	1	Math	2021-01-11 18:00:00	40
34	34	41	Nesibe Nurefşan Alkan	52	Tunahan Okatan	2	Chess	2021-01-11 20:00:00	30
35	35	42	Ömer Buğra Alparslan	53	Murathan Sakarya	3	Diction	2021-01-12 17:00:00	50
36	36	44	Mazlum Altan	65	İsmail Topçuoğlu	5	Programming	2021-01-12 19:00:00	30
37	37	47	Rana Altınoklu	62	Sezanur İmamoğlu	8	Web Design	2021-01-12 16:00:00	40
38	38	49	Ahmet Koru	60	Öğünç Ayaz	10	Spanish	2021-01-13 20:00:00	30
39	39	15	Servet Akçagunay	65	İsmail Topçuoğlu	5	Programming	2021-01-13 15:00:00	50
40	40	17	Recep Ali Samet Akdoğan	62	Sezanur İmamoğlu	8	Web Design	2021-01-13 18:00:00	30
41	41	27	Ahmet Raşit Akoğuz	51	Fatma Özlem Acar	1	Math	2021-01-04 15:00:00	35

After

	ID	StudentID	StudentName	InstructorID	InstructorName	CourseID	CourseName	Date	DurationInMins
20	20	28	Ecem Hatice Akova	69	Zeynep Doğa Temizer	10	Spanish	2021-01-12 19:00:00	30
21	21	29	Nüket Aksan	65	İsmail Topçuoğlu	1	Math	2021-01-12 16:00:00	40
22	22	30	Senem Aksevim	52	Tunahan Okatan	2	Chess	2021-01-13 20:00:00	30
23	23	31	Ayşen Aksoy	53	Murathan Sakarya	3	Diction	2021-01-13 15:00:00	50
24	24	32	Pekcan Aksöz	64	Farshad Karacabey	4	Basketball	2021-01-13 18:00:00	30
25	25	33	Bedirhan Lütfü Akşamoğlu	55	Gencay Ağaoğlu	5	Programming	2021-01-06 10:00:00	40
26	26	34	Semina Aktuna	66	Dilşah Pehlivan	6	Cardistry	2021-01-06 14:00:00	30
27	27	35	Eda Sena Akyıldız	67	Perker Sofyalıoğlu	7	Video Editing	2021-01-07 17:00:00	50
28	28	10	Cihan Akarpınar	57	Dorukhan Gürel	7	Video Editing	2021-01-07 19:00:00	30
29	29	10	Cihan Akarpınar	68	Mehmet Taş	8	Web Design	2021-01-08 16:00:00	40
30	30	36	Müyesser Akyıldırım	68	Mehmet Taş	8	Web Design	2021-01-09 20:00:00	30
31	31	37	Selinti Al	59	Çiğdem Bakkal	9	English	2021-01-10 15:00:00	50
32	32	38	Bahar Özlem Albaş	69	Zeynep Doğa Temizer	10	Spanish	2021-01-10 18:00:00	30
33	33	40	Kutlu Alibeyoğlu	51	Fatma Özlem Acar	1	Math	2021-01-11 18:00:00	40
34	34	41	Nesibe Nurefşan Alkan	52	Tunahan Okatan	2	Chess	2021-01-11 20:00:00	30
35	35	42	Ömer Buğra Alparslan	53	Murathan Sakarya	3	Diction	2021-01-12 17:00:00	50
36	36	44	Mazlum Altan	65	İsmail Topçuoğlu	5	Programming	2021-01-12 19:00:00	30
37	37	47	Rana Altınoklu	62	Sezanur İmamoğlu	8	Web Design	2021-01-12 16:00:00	40
38	38	49	Ahmet Koru	60	Öğünç Ayaz	10	Spanish	2021-01-13 20:00:00	30
39	39	15	Servet Akçagunay	65	İsmail Topçuoğlu	5	Programming	2021-01-13 15:00:00	50
40	40	17	Recep Ali Samet Akdoğan	62	Sezanur İmamoğlu	8	Web Design	2021-01-13 18:00:00	30
41	41	27	Ahmet Raşit Akoğuz	51	Fatma Özlem Acar	1	Math	2021-01-04 15:00:00	35
42	42	13	Nuhaydar Akbilmez	58	Abbas İncekara	1	Math	2021-01-02 12:00:00	45

## AddParent

This procedure adds a parent to the student with the given name, phone number and occupation.

## Code

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[AddParent]
6      @FullName NVARCHAR(50),
7      @PhoneNumber VARCHAR(15),
8      @Occupation NVARCHAR(30),
9      @StudentID SMALLINT
10 AS
11 BEGIN
12     SET NOCOUNT ON
13     INSERT INTO dbo.Parent(FullName, PhoneNumber, Occupation, StudentID)
14         VALUES ( @FullName, @PhoneNumber, @Occupation, @StudentID);
15 END
16 GO
17
```

## Before

	FullName	PhoneNumber	Occupation	StudentID
1	Ahmet Keskin	505-550-3579	Teacher	1
2	Mehmet Çelik	535-550-9682	Attorney	3
3	Ayça Ziyagil	535-550-1122	Astronaut	4
4	Gül Karacaoğlu	535-550-1345	Thief	7
5	Fatma Turp	535-550-5346	Painter	12
6	Ayşe Yavuz	535-550-2365	Housewife	16
7	Galip Sağlam	535-550-7243	NULL	27
8	Tarık Mengüç	535-550-2345	Actor	35
9	Kamil Korucu	535-550-7564	Doctor	49
10	Fırat Boru	535-550-5234	Chef	50

## After

	FullName	PhoneNumber	Occupation	StudentID
1	Ahmet Keskin	505-550-3579	Teacher	1
2	Murat Koyuncu	539-555-2345	NULL	2
3	Mehmet Çelik	535-550-9682	Attorney	3
4	Ayça Ziyagil	535-550-1122	Astronaut	4
5	Gül Karacaoğlu	535-550-1345	Thief	7
6	Fatma Turp	535-550-5346	Painter	12
7	Ayşe Yavuz	535-550-2365	Housewife	16
8	Galip Sağlam	535-550-7243	NULL	27
9	Tarık Mengüç	535-550-2345	Actor	35
10	Kamil Korucu	535-550-7564	Doctor	49
11	Fırat Boru	535-550-5234	Chef	50



## AddStudent

This procedure adds a new student to the system, creates a row in Person table and Student table.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[AddStudent]
6     @FirstName NVARCHAR(30),
7     @LastName NVARCHAR(30),
8     @MailAddress VARCHAR(30),
9     @PhoneNumber VARCHAR(15),
10    @Gender CHAR(1),
11    @ZIP CHAR(5) = '34000',
12    @District NVARCHAR(15) = NULL,
13    @City NVARCHAR(15) = N'İstanbul',
14    @BirthDate VARCHAR(15),
15    @RegistrationDate VARCHAR(15) = NULL,
16    @Payment SMALLINT
17 AS
18 IF @RegistrationDate IS NULL
19     SET @RegistrationDate = CONVERT(VARCHAR, GETDATE(), 1);
20 BEGIN
21     SET NOCOUNT ON
22     INSERT INTO dbo.Person(FirstName, LastName, MailAddress, PhoneNumber, Gender, ZIP, District, City)
23     VALUES (@FirstName, @LastName, @MailAddress, @PhoneNumber, @Gender, @ZIP, @District, @City);
24     INSERT INTO dbo.Student(PersonID, BirthDate, RegistrationDate, Payment)
25     VALUES (SCOPE_IDENTITY(), CAST(@BirthDate AS SMALLDATETIME), CAST(@RegistrationDate AS SMALLDATETIME), CAST(@Payment AS SMALLMONEY));
26 END
27 GO
28
```

## Before

	ID	FirstName	LastName
28	28	Ecem Hatice	Akova
29	29	Nüket	Aksan
30	30	Senem	Aksevim
31	31	Ayşen	Aksoy
32	32	Pekcan	Aksöz
33	33	Bedirhan Lütü	Akşamoğlu
34	34	Semina	Aktuna
35	35	Eda Sena	Akyıldız
36	36	Müyesser	Akyildir..
37	37	Selinti	Al
38	38	Bahar Özlem	Albaş
39	39	İlma	Aldağ
40	40	Kutlu	Alibeyoğ..
41	41	Nesibe Nurefşan	Alkan
42	42	Ömer Buğra	Alparslan
43	43	Hiba	Alpuğan
44	44	Mazlum	Altan
45	45	Elif Tuğçe	Altaş
46	46	Ahmet Ruken	Altay
47	47	Rana	Altınoklu
48	48	Erdi	Yol
49	49	Ahmet	Koru
50	50	Selinay	Akman

## After

	ID	FirstName	LastName
29	29	Nüket	Aksan
30	30	Senem	Aksevim
31	31	Ayşen	Aksoy
32	32	Pekcan	Aksöz
33	33	Bedirhan Lütü	Akşamoğlu
34	34	Semina	Aktuna
35	35	Eda Sena	Akyıldız
36	36	Müyesser	Akyildir..
37	37	Selinti	Al
38	38	Bahar Özlem	Albaş
39	39	İlma	Aldağ
40	40	Kutlu	Alibeyoğ..
41	41	Nesibe Nurefşan	Alkan
42	42	Ömer Buğra	Alparslan
43	43	Hiba	Alpuğan
44	44	Mazlum	Altan
45	45	Elif Tuğçe	Altaş
46	46	Ahmet Ruken	Altay
47	47	Rana	Altınoklu
48	48	Erdi	Yol
49	49	Ahmet	Koru
50	50	Selinay	Akman
51	72	Korkut	Çağlayan

## AddStudentCourse

This procedure registers student to a stored course with the given ID's.

### Code

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[AddStudentCourse]
6  |   @StudentID SMALLINT,
7  |   @CourseID SMALLINT
8  AS
9  BEGIN
10 |   SET NOCOUNT ON
11 |   INSERT INTO StudentCourse(StudentID,CourseID)
12 |   VALUES (@StudentID, @CourseID);
13 END
14 GO
15
```

### Before

	StudentID	CourseID	MainGrade	DateCompleted
1	1	1	NULL	NULL
2	1	2	NULL	NULL
3	2	3	NULL	NULL
4	2	4	NULL	NULL
5	2	5	NULL	NULL
6	3	6	NULL	NULL
7	3	7	NULL	NULL
8	5	8	NULL	NULL
9	5	9	NULL	NULL
10	6	10	NULL	NULL
11	7	1	NULL	NULL
12	8	2	NULL	NULL
13	8	3	NULL	NULL
14	9	4	NULL	NULL
15	10	5	NULL	NULL
16	10	6	NULL	NULL
17	10	7	NULL	NULL
18	10	8	NULL	NULL
19	12	9	NULL	NULL
20	12	10	NULL	NULL
21	13	1	NULL	NULL
22	13	2	NULL	NULL
23	15	3	NULL	NULL

### After

	StudentID	CourseID	MainGrade	DateCompleted
1	1	1	NULL	NULL
2	1	2	NULL	NULL
3	1	4	NULL	NULL
4	2	3	NULL	NULL
5	2	4	NULL	NULL
6	2	5	NULL	NULL
7	3	6	NULL	NULL
8	3	7	NULL	NULL
9	5	8	NULL	NULL
10	5	9	NULL	NULL
11	6	10	NULL	NULL
12	7	1	NULL	NULL
13	8	2	NULL	NULL
14	8	3	NULL	NULL
15	9	4	NULL	NULL
16	10	5	NULL	NULL
17	10	6	NULL	NULL
18	10	7	NULL	NULL
19	10	8	NULL	NULL
20	12	9	NULL	NULL
21	12	10	NULL	NULL
22	13	1	NULL	NULL
23	13	2	NULL	NULL

## AddStudentExam

This procedure adds an exam record for the student which contains student's grade, correct and false answers for the exam.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[AddStudentExam]
6     @StudentID SMALLINT,
7     @ExamID SMALLINT,
8     @Grade TINYINT,
9     @CorrectAnswers TINYINT,
10    @FalseAnswers TINYINT
11 AS
12 BEGIN
13     DECLARE @NumOfQuestions AS TINYINT;
14     SET @NumOfQuestions = (SELECT NumOfQuestions FROM Exam WHERE Exam.ID = @ExamID);
15     IF @CorrectAnswers + @FalseAnswers > @NumOfQuestions BEGIN
16         PRINT 'ERROR: CorrectAnswers + FalseAnswers can't be greater than NumOfQuestions.';
17         RETURN
18     END
19     IF @Grade IS NULL BEGIN
20         SET @Grade = @CorrectAnswers * 100.0 / @NumOfQuestions;
21     END
22     SET NOCOUNT ON
23     INSERT INTO StudentExam(StudentID, ExamID, Grade, CorrectAnswers, FalseAnswers)
24     VALUES (@StudentID, @ExamID, @Grade, @CorrectAnswers, @FalseAnswers);
25 END
26 GO
27
```

## Before

	StudentID	ExamID	Grade	CorrectAnswers	FalseAnswers
1	1	1	80	8	2
2	1	2	60	12	0
3	1	3	60	3	1
4	2	5	90	18	2
5	2	6	0	0	0
6	2	8	76	23	7
7	3	7	70	7	2
8	3	10	75	15	4
9	3	11	60	27	2
10	5	15	80	4	1
11	5	17	91	11	1
12	6	18	93	14	1
13	7	1	70	7	1
14	7	2	75	15	2
15	8	5	80	16	4
16	8	6	100	1	0
17	10	7	90	9	1
18	10	8	86	26	3
19	10	10	85	17	3
20	10	11	75	34	2
21	10	12	100	7	0
22	10	15	100	5	0
23	10	16	75	3	1

## After

	StudentID	ExamID	Grade	CorrectAnswers	FalseAnswers
1	1	1	80	8	2
2	1	2	60	12	0
3	1	3	60	3	1
4	1	8	83	25	0
5	2	5	90	18	2
6	2	6	0	0	0
7	2	8	76	23	7
8	3	7	70	7	2
9	3	10	75	15	4
10	3	11	60	27	2
11	5	15	80	4	1
12	5	17	91	11	1
13	6	18	93	14	1
14	7	1	70	7	1
15	7	2	75	15	2
16	8	5	80	16	4
17	8	6	100	1	0
18	10	7	90	9	1
19	10	8	86	26	3
20	10	10	85	17	3
21	10	11	75	34	2
22	10	12	100	7	0
23	10	15	100	5	0

## DeleteCourse

This procedure deletes the course with the given ID.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[DeleteCourse]
6     @Name nvarchar(50)
7 AS
8 BEGIN
9     SET NOCOUNT ON
10    DELETE FROM Course WHERE @Name=Course.Name;
11 END
12 GO
13
```

## Before

	ID	Name
1	4	Basketball
2	6	Cardistry
3	2	Chess
4	3	Diction
5	9	English
6	12	Literature
7	1	Math
8	5	Programming
9	10	Spanish
10	7	Video Editing
11	8	Web Design

## After

	ID	Name
1	4	Basketball
2	6	Cardistry
3	2	Chess
4	3	Diction
5	9	English
6	1	Math
7	5	Programming
8	10	Spanish
9	7	Video Editing
10	8	Web Design

## DeleteInstructorAvailableTime

This procedure deletes an available time from the instructor with the given ID.

### Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[DeleteInstructorAvailableTime]
6     @InstructorID SMALLINT,
7     @StartTime NVARCHAR(50)
8 AS
9 BEGIN
10     SET NOCOUNT ON
11     DELETE FROM InstructorAvailableTime
12         WHERE @InstructorID=InstructorAvailableTime.InstructorID AND @StartTime=InstructorAvailableTime.StartTime;
13 END
14 GO
15
```

### Before

	InstructorID	StartTime
1	51	2020-01-05 14:00:00
2	51	2021-01-04 09:00:00
3	51	2021-01-04 19:00:00
4	52	2021-01-05 09:00:00
5	52	2021-01-05 15:00:00
6	52	2021-01-05 19:00:00
7	53	2021-01-06 09:00:00
8	53	2021-01-06 15:00:00
9	53	2021-01-06 19:00:00
1..	54	2021-01-07 09:00:00
1..	54	2021-01-07 15:00:00
1..	54	2021-01-07 19:00:00
1..	55	2021-01-08 09:00:00
1..	55	2021-01-08 15:00:00
1..	55	2021-01-08 19:00:00
1..	56	2021-01-09 09:00:00
1..	56	2021-01-09 15:00:00
1..	56	2021-01-09 19:00:00
1..	57	2021-01-10 09:00:00
2..	57	2021-01-10 15:00:00
2..	57	2021-01-10 19:00:00
2..	58	2021-01-11 12:00:00
2..	58	2021-01-11 14:00:00

### After

	InstructorID	StartTime
1	51	2021-01-04 09:00:00
2	51	2021-01-04 19:00:00
3	52	2021-01-05 09:00:00
4	52	2021-01-05 15:00:00
5	52	2021-01-05 19:00:00
6	53	2021-01-06 09:00:00
7	53	2021-01-06 15:00:00
8	53	2021-01-06 19:00:00
9	54	2021-01-07 09:00:00
1..	54	2021-01-07 15:00:00
1..	54	2021-01-07 19:00:00
1..	55	2021-01-08 09:00:00
1..	55	2021-01-08 15:00:00
1..	55	2021-01-08 19:00:00
1..	56	2021-01-09 09:00:00
1..	56	2021-01-09 15:00:00
1..	56	2021-01-09 19:00:00
1..	57	2021-01-10 09:00:00
1..	57	2021-01-10 15:00:00
2..	57	2021-01-10 19:00:00
2..	58	2021-01-11 12:00:00
2..	58	2021-01-11 14:00:00
2..	58	2021-01-11 18:00:00

## DeletelInstructorCourse

This procedure removes an instructor from a course with the given IDs.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[DeletelInstructorCourse]
6     @InstructorID SMALLINT,
7     @CourseID SMALLINT
8 AS
9 BEGIN
10     SET NOCOUNT ON
11     DELETE FROM InstructorCourse
12     WHERE @InstructorID=InstructorCourse.InstructorID AND @CourseID=InstructorCourse.CourseID;
13 END
14 GO
15
```

## Before

	InstructorID	CourseID
1	51	1
2	51	2
3	51	7
4	51	9
5	52	2
6	53	3
7	54	4
8	55	1
9	55	5
10	56	6
11	57	7
12	57	8
13	59	9
14	60	9
15	60	10
16	61	1
17	62	1
18	62	2
19	62	5
20	62	7
21	62	8
22	62	9
23	63	3

## After

	InstructorID	CourseID
1	51	2
2	51	7
3	51	9
4	52	2
5	53	3
6	54	4
7	55	1
8	55	5
9	56	6
1_	57	7
1_	57	8
1_	59	9
1_	60	9
1_	60	10
1_	61	1
1_	62	1
1_	62	2
1_	62	5
1_	62	7
2_	62	8
2_	62	9
2_	63	3
2_	64	4

## DeleteMeeting

This procedure deletes the meeting with the given ID from the database.

## Code

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[DeleteMeeting]
6  |  @MeetingID SMALLINT
7  AS
8  BEGIN
9  |  SET NOCOUNT ON
10 |  DELETE FROM Meeting WHERE @MeetingID=Meeting.ID;
11 END
12 GO
13
```

## Before

	ID	StudentID	StudentName	InstructorID	InstructorName	CourseID	CourseName	Date	DurationInMins
1	1	7	Tutkum Yılmaz	51	Fatma Özlem Acar	1	Math	2021-01-04 10:00:00	40
2	2	7	Tutkum Yılmaz	66	Dilşah Pehlivan	6	Cardistry	2021-01-04 14:00:00	30
3	3	2	Gözde Acarkan	53	Murathan Sakarya	3	Diction	2021-01-04 15:00:00	50
4	4	3	Halim Salih	52	Tunahan Okatan	2	Chess	2021-01-04 16:00:00	30
5	5	5	Aykanat Ağıroğlu	57	Dorukhan Gürel	8	Web Design	2021-01-05 16:00:00	40
6	6	6	Şennur Ağnar	60	Öğünç Ayaz	10	Spanish	2021-01-05 20:00:00	30
7	7	7	Tutkum Yılmaz	61	Gürkan Yıldırım	1	Math	2021-01-05 15:00:00	50
8	8	8	Mügenur Ahmet	63	Ahmet Utku Kantarcı	3	Diction	2021-01-05 16:00:00	30
9	9	10	Cihan Akarpınar	62	Sezanur İmamoğlu	5	Programming	2021-01-06 10:00:00	40
1	1	12	Mehmetcan Akay	59	Çiğdem Bakkal	9	English	2021-01-06 14:00:00	30
1	1	13	Nuhaydar Akbilm.	65	İsmail Topçuoğlu	1	Math	2021-01-07 17:00:00	50
1	1	13	Nuhaydar Akbilm.	52	Tunahan Okatan	2	Chess	2021-01-07 19:00:00	30
1	1	15	Servet Akçagunay	67	Perker Sofyalıoğlu	4	Basketball	2021-01-08 16:00:00	40
1	1	17	Recep Ali Samet.	67	Perker Sofyalıoğlu	7	Video Edit.	2021-01-09 20:00:00	30
1	1	20	Ercüment Akıncı.	70	Ramazan Usluer	9	English	2021-01-10 15:00:00	50
1	1	21	Sarper Akış	63	Ahmet Utku Kantarcı	3	Diction	2021-01-10 18:00:00	30
1	1	25	Ahmet Polat Akı.	57	Dorukhan Gürel	7	Video Edit.	2021-01-11 18:00:00	40
1	1	25	Ahmet Polat Akı.	68	Mehmet Taş	8	Web Design	2021-01-11 20:00:00	30
1	1	27	Ahmet Raşit Ako.	62	Sezanur İmamoğlu	9	English	2021-01-12 17:00:00	50
2	2	28	Ecem Hatice Ako.	69	Zeynep Doğa Temizer	10	Spanish	2021-01-12 19:00:00	30
2	2	29	Nüket Aksan	65	İsmail Topçuoğlu	1	Math	2021-01-12 16:00:00	40
2	2	30	Senem Aksevim	52	Tunahan Okatan	2	Chess	2021-01-13 20:00:00	30
2	2	31	Ayşen Aksoy	53	Murathan Sakarya	3	Diction	2021-01-13 15:00:00	50

## After

	ID	StudentID	StudentName	InstructorID	InstructorName	CourseID	CourseName	Date	DurationInMins
1	2	7	Tutkum Yılmaz	66	Dilşah Pehlivan	6	Cardistry	2021-01-04 14:00:00	30
2	3	2	Gözde Acarkan	53	Murathan Sakarya	3	Diction	2021-01-04 15:00:00	50
3	4	3	Halim Salih	52	Tunahan Okatan	2	Chess	2021-01-04 16:00:00	30
4	5	5	Aykanat Ağiroğlu	57	Dorukhan Gürel	8	Web Design	2021-01-05 16:00:00	40
5	6	6	Şennur Ağnar	60	Öğünç Ayaz	10	Spanish	2021-01-05 20:00:00	30
6	7	7	Tutkum Yılmaz	61	Gürkan Yıldırım	1	Math	2021-01-05 15:00:00	50
7	8	8	Mügenur Ahmet	63	Ahmet Utku Kantarcı	3	Diction	2021-01-05 16:00:00	30
8	9	10	Cihan Akarpınar	62	Sezanur İmamoğlu	5	Programming	2021-01-06 10:00:00	40
9	10	12	Mehmetcan Akay	59	Çiğdem Bakkal	9	English	2021-01-06 14:00:00	30
1..	11	13	Nuhaydar Akbilm...	65	İsmail Topçuoğlu	1	Math	2021-01-07 17:00:00	50
1..	12	13	Nuhaydar Akbilm...	52	Tunahan Okatan	2	Chess	2021-01-07 19:00:00	30
1..	13	15	Servet Akçagunay	67	Perker Sofyalıoğlu	4	Basketball	2021-01-08 16:00:00	40
1..	14	17	Recep Ali Samet...	67	Perker Sofyalıoğlu	7	Video Edit...	2021-01-09 20:00:00	30
1..	15	20	Ercüment Akıncı...	70	Ramazan Usluer	9	English	2021-01-10 15:00:00	50
1..	16	21	Sarper Akış	63	Ahmet Utku Kantarcı	3	Diction	2021-01-10 18:00:00	30
1..	17	25	Ahmet Polat Akl...	57	Dorukhan Gürel	7	Video Edit...	2021-01-11 18:00:00	40
1..	18	25	Ahmet Polat Akl...	68	Mehmet Taş	8	Web Design	2021-01-11 20:00:00	30
1..	19	27	Ahmet Raşit Ako...	62	Sezanur İmamoğlu	9	English	2021-01-12 17:00:00	50
1..	20	28	Ecem Hatice Ako...	69	Zeynep Doğa Temizer	10	Spanish	2021-01-12 19:00:00	30
2..	21	29	Nüket Aksan	65	İsmail Topçuoğlu	1	Math	2021-01-12 16:00:00	40
2..	22	30	Senem Aksevim	52	Tunahan Okatan	2	Chess	2021-01-13 20:00:00	30
2..	23	31	Ayşen Aksoy	53	Murathan Sakarya	3	Diction	2021-01-13 15:00:00	50
2..	24	32	Pekcan Aksöz	64	Farshad Karacabey	4	Basketball	2021-01-13 18:00:00	30

## DeletePerson

This procedure deletes a student or an instructor with the given ID. In this way with cascade, every record related to the person to be deleted will also be deleted.

## Code

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[DeletePerson]
6  |   @PersonID SMALLINT
7  AS
8  BEGIN
9  |   SET NOCOUNT ON
10 |   DELETE FROM Person WHERE @PersonID=Person.ID;
11 END
12 GO
13
```



Before

	ID	FirstName	LastName	MailAddress	PhoneNumber	Gender	ZIP	District	City
1	1	Fatma Özlem	Acar	fatma.acar@gmail.com	545-550-59880	F	34000	NULL	Kocaeli
2	2	Gözde	Acarkan	gozde.acarkan@gmail.com	535-553-43084	F	34000	Kadıköy	İstanbul
3	3	Halim	Salih	halim.salih@gmail.com	555-559-2898	M	34000	Üsküdar	İstanbul
4	4	Bestami	Ağırağaç	bestami.agiragac@gmail.com	535-550-2731	M	41000	NULL	Kocaeli
5	5	Aykanat	Ağiroğlu	aykanat.agiroglu@gmail.com	535-554-8892	M	41000	NULL	Kocaeli
6	6	Şennur	Ağnar	sennur.agnar@gmail.com	505-550-1954	F	34000	Beşiktaş	İstanbul
7	7	Tutkum	Yılmaz	tutkum.yilmaz@gmail.com	555-557-2723	F	34000	Üsküdar	İstanbul
8	8	Mügenur	Ahmet	mugenur.ahmet@gmail.com	545-556-0994	F	34000	NULL	İstanbul
9	9	Sevinç	Ak	sevinc.ak@gmail.com	555-554-6195	F	34000	Kadıköy	İstanbul
10	10	Cihan	Akarpınar	cihan.akarpinar@gmail.com	545-555-2228	M	41000	NULL	Kocaeli
11	11	Rafi	Akaş	rafi.akas@gmail.com	505-551-5861	M	34000	Mecidiyeköy	İstanbul
12	12	Mehmetcan	Akay	mehmetcan.akay@gmail.com	505-551-5061	M	41000	NULL	Kocaeli
13	13	Nuhaydar	Akbilmez	nuhaydar.akbilmez@gmail.com	505-551-5081	F	34000	Beşiktaş	İstanbul
14	14	Emine Münevver	Akca	emine.munever@gmail.com	535-553-5260	F	41000	NULL	Kocaeli
15	15	Servet	Akçagunay	servet.akcagunay@hotmail.com	505-558-4638	M	34000	Mecidiyeköy	İstanbul
16	16	Çilem	Akçay	cilem.akcay@gmail.com	555-552-8187	F	34000	NULL	İstanbul
17	17	Recep Ali Samet	Akdoğan	recepali.akdogan@gmail.com	535-552-3634	M	34000	Kadıköy	İstanbul
18	18	Emre Ayberk	Akfırat	ayberk.akfirat@gmail.com	505-556-6147	M	34000	NULL	İstanbul
19	19	Kerime Hacer	Akıllı	hacer.akilli@gmail.com	505-557-4463	F	34000	Kadıköy	İstanbul
20	20	Ercüment	Akıncılar	ercument.akincilar@gmail.com	555-554-6495	M	34000	NULL	Kocaeli
21	21	Sarper	Akış	sarper.akis@gmail.com	505-555-73829	M	34000	NULL	İstanbul
22	22	Berker	Akkiray	berker.akkiray@gmail.com	505-553-8020	M	34000	Beşiktaş	İstanbul
23	23	İclal	Akkoyun	iclal.akkoyun@gmail.com	555-558-9865	F	34000	NULL	İstanbul

After

	ID	FirstName	LastName	MailAddress	PhoneNumber	Gender	ZIP	District	City
1	2	Gözde	Acarkan	gozde.acarkan@gmail.com	535-553-43084	F	34000	Kadıköy	İstanbul
2	3	Halim	Salih	halim.salih@gmail.com	555-559-2898	M	34000	Üsküdar	İstanbul
3	4	Bestami	Ağırağaç	bestami.agiragac@gmail.com	535-550-2731	M	41000	NULL	Kocaeli
4	5	Aykanat	Ağiroğlu	aykanat.agiroglu@gmail.com	535-554-8892	M	41000	NULL	Kocaeli
5	6	Şennur	Ağnar	sennur.agnar@gmail.com	505-550-1954	F	34000	Beşiktaş	İstanbul
6	7	Tutkum	Yılmaz	tutkum.yilmaz@gmail.com	555-557-2723	F	34000	Üsküdar	İstanbul
7	8	Mügenur	Ahmet	mugenur.ahmet@gmail.com	545-556-0994	F	34000	NULL	İstanbul
8	9	Sevinç	Ak	sevinc.ak@gmail.com	555-554-6195	F	34000	Kadıköy	İstanbul
9	10	Cihan	Akarpınar	cihan.akarpinar@gmail.com	545-555-2228	M	41000	NULL	Kocaeli
10	11	Rafi	Akaş	rafi.akas@gmail.com	505-551-5861	M	34000	Mecidiyeköy	İstanbul
11	12	Mehmetcan	Akay	mehmetcan.akay@gmail.com	505-551-5061	M	41000	NULL	Kocaeli
12	13	Nuhaydar	Akbilmez	nuhaydar.akbilmez@gmail.com	505-551-5081	F	34000	Beşiktaş	İstanbul
13	14	Emine Münevver	Akca	emine.munever@gmail.com	535-553-5260	F	41000	NULL	Kocaeli
14	15	Servet	Akçagunay	servet.akcagunay@hotmail.com	505-558-4638	M	34000	Mecidiyeköy	İstanbul
15	16	Çilem	Akçay	cilem.akcay@gmail.com	555-552-8187	F	34000	NULL	İstanbul
16	17	Recep Ali Samet	Akdoğan	recepali.akdogan@gmail.com	535-552-3634	M	34000	Kadıköy	İstanbul
17	18	Emre Ayberk	Akfırat	ayberk.akfirat@gmail.com	505-556-6147	M	34000	NULL	İstanbul
18	19	Kerime Hacer	Akıllı	hacer.akilli@gmail.com	505-557-4463	F	34000	Kadıköy	İstanbul
19	20	Ercüment	Akıncılar	ercument.akincilar@gmail.com	555-554-6495	M	34000	NULL	Kocaeli
20	21	Sarper	Akış	sarper.akis@gmail.com	505-555-73829	M	34000	NULL	İstanbul
21	22	Berker	Akkiray	berker.akkiray@gmail.com	505-553-8020	M	34000	Beşiktaş	İstanbul
22	23	İclal	Akkoyun	iclal.akkoyun@gmail.com	555-558-9865	F	34000	NULL	İstanbul
23	24	Lemis	Akküt	lemis.akkut@gmail.com	555-559-02946	M	34000	NULL	İstanbul

## DeleteStudentExam

This procedure deletes exam record of the student with the given IDs.

## Code

```
1 SET ANSI_WARNINGS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[DeleteStudentExam]
6     @StudentID SMALLINT,
7     @ExamID SMALLINT
8 AS
9 BEGIN
10     SET NOCOUNT ON
11     DELETE FROM StudentExam
12     WHERE @StudentID=StudentID AND @ExamID=StudentExam.ExamID;
13 END
14 GO
15
```

## Before

	StudentID	ExamID	Grade	CorrectAnswers	FalseAnswers
1	2	5	90	18	2
2	2	6	0	0	0
3	2	8	76	23	7
4	3	7	70	7	2
5	3	10	75	15	4
6	3	11	60	27	2
7	5	15	80	4	1
8	5	17	91	11	1
9	6	18	93	14	1
10	7	1	70	7	1
11	7	2	75	15	2
12	8	5	80	16	4
13	8	6	100	1	0
14	10	7	90	9	1
15	10	8	86	26	3
16	10	10	85	17	3
17	10	11	75	34	2
18	10	12	100	7	0
19	10	15	100	5	0
20	10	16	75	3	1
21	12	17	100	12	0
22	12	18	86	13	2
23	12	19	86	13	1

## After

	StudentID	ExamID	Grade	CorrectAnswers	FalseAnswers
1	2	6	0	0	0
2	2	8	76	23	7
3	3	7	70	7	2
4	3	10	75	15	4
5	3	11	60	27	2
6	5	15	80	4	1
7	5	17	91	11	1
8	6	18	93	14	1
9	7	1	70	7	1
10	7	2	75	15	2
11	8	5	80	16	4
12	8	6	100	1	0
13	10	7	90	9	1
14	10	8	86	26	3
15	10	10	85	17	3
16	10	11	75	34	2
17	10	12	100	7	0
18	10	15	100	5	0
19	10	16	75	3	1
20	12	17	100	12	0
21	12	18	86	13	2
22	12	19	86	13	1
23	13	1	70	7	3

## Delete Topic

This procedure deletes the topic with the given name from the course with the given ID.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[DeleteTopic]
6     @CourseID SMALLINT,
7     @TopicName NVARCHAR(50)
8 AS
9 BEGIN
10     SET NOCOUNT ON
11     DELETE FROM CourseTopic
12         WHERE @CourseID=CourseTopic.CourseID AND @TopicName=CourseTopic.Name;
13 END
14 GO
15
```

## Before

	CourseID	Name
1	1	Derivative
2	1	Geometry
3	1	Integral
4	1	Limits
5	1	Series
6	2	End Game
7	2	Fork
8	2	Middle Ga...
9	2	Openings
10	3	Letters
11	3	Patterns
12	3	Toning
13	4	Jumps
14	4	Positioni...
15	4	Rebound
16	4	Step
17	5	C
18	5	Data Stru...
19	5	Database
20	5	Introduct...
21	5	Java
22	6	Cards
23	6	Cuts

## After

	CourseID	Name
1	1	Derivative
2	1	Geometry
3	1	Integral
4	1	Series
5	2	End Game
6	2	Fork
7	2	Middle Ga...
8	2	Openings
9	3	Letters
10	3	Patterns
11	3	Toning
12	4	Jumps
13	4	Positioni...
14	4	Rebound
15	4	Step
16	5	C
17	5	Data Stru...
18	5	Database
19	5	Introduct...
20	5	Java
21	6	Cards
22	6	Cuts
23	6	Decks

## UpdateMeeting

This procedure updates meeting date and duration with the given values. If one of them is NULL, then the procedure doesn't update that field.

### Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[UpdateMeeting]
6     @MeetingID SMALLINT,
7     @Date VARCHAR(15),
8     @DurationInMins TINYINT
9 AS
10 BEGIN
11     SET NOCOUNT ON
12     UPDATE Meeting
13     SET Date = @Date
14     WHERE @Date IS NOT NULL and ID = @MeetingID
15     UPDATE Meeting
16     SET DurationInMins = @DurationInMins
17     WHERE @DurationInMins IS NOT NULL and ID = @MeetingID
18
19 END
20 GO
21
```

### Before

	ID	Date	DurationInMins	StudentID	InstructorID	CourseID
1	2	2021-01-04 14:00:00	30	7	66	6
2	3	2021-01-04 15:00:00	50	2	53	3
3	4	2021-01-04 16:00:00	30	3	52	2
4	5	2021-01-05 16:00:00	40	5	57	8
5	6	2021-01-05 20:00:00	30	6	60	10
6	7	2021-01-05 15:00:00	50	7	61	1
7	8	2021-01-05 16:00:00	30	8	63	3
8	9	2021-01-06 10:00:00	40	10	62	5
9	10	2021-01-06 14:00:00	30	12	59	9
1..	11	2021-01-07 17:00:00	50	13	65	1
1..	12	2021-01-07 19:00:00	30	13	52	2
1..	13	2021-01-08 16:00:00	40	15	67	4
1..	14	2021-01-09 20:00:00	30	17	67	7
1..	15	2021-01-10 15:00:00	50	20	70	9
1..	16	2021-01-10 18:00:00	30	21	63	3
1..	17	2021-01-11 18:00:00	40	25	57	7
1..	18	2021-01-11 20:00:00	30	25	68	8
1..	19	2021-01-12 17:00:00	50	27	62	9
1..	20	2021-01-12 19:00:00	30	28	69	10
2..	21	2021-01-12 16:00:00	40	29	65	1
2..	22	2021-01-13 20:00:00	30	30	52	2
2..	23	2021-01-13 15:00:00	50	31	53	3
2..	24	2021-01-13 18:00:00	30	32	64	4

## After

	ID	Date	DurationInMins	StudentID	InstructorID	CourseID
1	2	2021-01-10 18:00:00	40	7	66	6
2	3	2021-01-04 15:00:00	50	2	53	3
3	4	2021-01-04 16:00:00	30	3	52	2
4	5	2021-01-05 16:00:00	40	5	57	8
5	6	2021-01-05 20:00:00	30	6	60	10
6	7	2021-01-05 15:00:00	50	7	61	1
7	8	2021-01-05 16:00:00	30	8	63	3
8	9	2021-01-06 10:00:00	40	10	62	5
9	1..	2021-01-06 14:00:00	30	12	59	9
1..	1..	2021-01-07 17:00:00	50	13	65	1
1..	1..	2021-01-07 19:00:00	30	13	52	2
1..	1..	2021-01-08 16:00:00	40	15	67	4
1..	1..	2021-01-09 20:00:00	30	17	67	7
1..	1..	2021-01-10 15:00:00	50	20	70	9
1..	1..	2021-01-10 18:00:00	30	21	63	3
1..	1..	2021-01-11 18:00:00	40	25	57	7
1..	1..	2021-01-11 20:00:00	30	25	68	8
1..	1..	2021-01-12 17:00:00	50	27	62	9
1..	2..	2021-01-12 19:00:00	30	28	69	10
2..	2..	2021-01-12 16:00:00	40	29	65	1
2..	2..	2021-01-13 20:00:00	30	30	52	2
2..	2..	2021-01-13 15:00:00	50	31	53	3
2..	2..	2021-01-13 18:00:00	30	32	64	4

## UpdatePerson

This procedure updates mail address and phone number of the person with the given values. If one of them is NULL, then the procedure doesn't update that field.

## Code

```
1 SET ANSI_NULLS ON
2 GO
3 SET QUOTED_IDENTIFIER ON
4 GO
5 CREATE PROCEDURE [dbo].[UpdatePerson]
6     @PersonID SMALLINT,
7     @MailAddress VARCHAR(50),
8     @PhoneNumber VARCHAR(15)
9 AS
10 BEGIN
11     SET NOCOUNT ON
12     UPDATE Person
13     SET MailAddress = @MailAddress
14     WHERE @MailAddress IS NOT NULL and ID = @PersonID
15     UPDATE Person
16     SET PhoneNumber = @PhoneNumber
17     WHERE @PhoneNumber IS NOT NULL and ID = @PersonID
18 END
19 GO
20
```

## Before

	ID	FirstName	LastName	MailAddress	PhoneNumber	Gender	ZIP	District	City
1	2	Gözde	Acarkan	gozde.acarkan@gmail.com	535-553-43084	F	34000	Kadıköy	İstanbul
2	3	Halim	Salih	halim.salih@gmail.com	555-559-2898	M	34000	Üsküdar	İstanbul
3	4	Bestami	Ağırağaç	bestami.agiragac@gmail.com	535-550-2731	M	41000	NULL	Kocaeli
4	5	Aykanat	Ağiroğlu	aykanat.agiroglu@gmail.com	535-554-8892	M	41000	NULL	Kocaeli
5	6	Şennur	Ağnar	sennur.agnar@gmail.com	505-550-1954	F	34000	Beşiktaş	İstanbul
6	7	Tutkum	Yılmaz	tutkum.yilmaz@gmail.com	555-557-2723	F	34000	Üsküdar	İstanbul
7	8	Mügenur	Ahmet	mugenur.ahmet@gmail.com	545-556-0994	F	34000	NULL	İstanbul
8	9	Sevinç	Ak	sevinc.ak@gmail.com	555-554-6195	F	34000	Kadıköy	İstanbul
9	10	Cihan	Akarpınar	cihan.akarpinar@gmail.com	545-555-2228	M	41000	NULL	Kocaeli
10	11	Rafi	Akaş	rafi.akas@gmail.com	505-551-5861	M	34000	Mecidiyeköy	İstanbul
11	12	Mehmetcan	Akay	mehmetcan.akay@gmail.com	505-551-5061	M	41000	NULL	Kocaeli
12	13	Nuhaydar	Akbilmez	nuhaydar.akbilmez@gmail.com	505-551-5081	F	34000	Beşiktaş	İstanbul
13	14	Emine Münevver	Akca	emine.munever@gmail.com	535-553-5260	F	41000	NULL	Kocaeli
14	15	Servet	Akçagunay	servet.akcagunay@hotmail.com	505-558-4638	M	34000	Mecidiyeköy	İstanbul
15	16	Çilem	Akçay	cilem.akcay@gmail.com	555-552-8187	F	34000	NULL	İstanbul
16	17	Recep Ali Samet	Akdoğan	recepali.akdogan@gmail.com	535-552-3634	M	34000	Kadıköy	İstanbul
17	18	Emre Ayberk	Akfırat	ayberk.akfirat@gmail.com	505-556-6147	M	34000	NULL	İstanbul
18	19	Kerime Hacer	Akıllı	hacer.akilli@gmail.com	505-557-4463	F	34000	Kadıköy	İstanbul
19	20	Ercüment	Akıncılar	ercument.akincilar@gmail.com	555-554-6495	M	34000	NULL	Kocaeli
20	21	Sarper	Akış	sarper.akis@gmail.com	505-555-73829	M	34000	NULL	İstanbul
21	22	Berker	Akkiray	berker.akkiray@gmail.com	505-553-8020	M	34000	Beşiktaş	İstanbul
22	23	İclal	Akkoyun	iclal.akkoyun@gmail.com	555-558-9865	F	34000	NULL	İstanbul
23	24	Lemis	Akküt	lemis.akkut@gmail.com	555-559-02946	M	34000	NULL	İstanbul

## After

	ID	FirstName	LastName	MailAddress	PhoneNumber	Gender	ZIP	District	City
1	2	Gözde	Acarkan	acarkan58@gmail.com	540-521-1358	F	34000	Kadıköy	İstanbul
2	3	Halim	Salih	halim.salih@gmail.com	555-559-2898	M	34000	Üsküdar	İstanbul
3	4	Bestami	Ağırbaş	bestami.agirbas@gmail.com	535-550-2731	M	41000	NULL	Kocaeli
4	5	Aykanat	Ağırbaş	aykanat.agirbas@gmail.com	535-554-8892	M	41000	NULL	Kocaeli
5	6	Şennur	Ağnar	sennur.agnar@gmail.com	505-550-1954	F	34000	Beşiktaş	İstanbul
6	7	Tutkum	Yılmaz	tutkum.yilmaz@gmail.com	555-557-2723	F	34000	Üsküdar	İstanbul
7	8	Müenur	Ahmet	mugenur.ahmet@gmail.com	545-556-0994	F	34000	NULL	İstanbul
8	9	Sevinç	Ak	sevinc.ak@gmail.com	555-554-6195	F	34000	Kadıköy	İstanbul
9	10	Cihan	Akarpınar	cihan.akarpinar@gmail.com	545-555-2228	M	41000	NULL	Kocaeli
10	11	Rafi	Akaş	rafi.akas@gmail.com	505-551-5861	M	34000	Mecidiyeköy	İstanbul
11	12	Mehmetcan	Akay	mehmetcan.akay@gmail.com	505-551-5061	M	41000	NULL	Kocaeli
12	13	Nuhaydar	Akbilmez	nuhaydar.akbilmez@gmail.com	505-551-5081	F	34000	Beşiktaş	İstanbul
13	14	Emine Münevver	Akca	emine.munevver@gmail.com	535-553-5260	F	41000	NULL	Kocaeli
14	15	Servet	Akçagunay	servet.akcagunay@hotmail.com	505-558-4638	M	34000	Mecidiyeköy	İstanbul
15	16	Çilem	Akçay	cilem.akcay@gmail.com	555-552-8187	F	34000	NULL	İstanbul
16	17	Recep Ali Samet	Akdoğan	recepali.akdogan@gmail.com	535-552-3634	M	34000	Kadıköy	İstanbul
17	18	Emre Ayberk	Akfırat	ayberk.akfirat@gmail.com	505-556-6147	M	34000	NULL	İstanbul
18	19	Kerime Hacer	Akıllı	hacer.akilli@gmail.com	505-557-4463	F	34000	Kadıköy	İstanbul
19	20	Ercüment	Akıncılar	ercument.akincilar@gmail.com	555-554-6495	M	34000	NULL	Kocaeli
20	21	Sarper	Akış	sarper.akis@gmail.com	505-555-73829	M	34000	NULL	İstanbul
21	22	Berker	Akkiray	berker.akkiray@gmail.com	505-553-8020	M	34000	Beşiktaş	İstanbul
22	23	İclal	Akkoyun	iclal.akkoyun@gmail.com	555-558-9865	F	34000	NULL	İstanbul
23	24	Lemis	Akküt	lemis.akkut@gmail.com	555-559-02946	M	34000	NULL	İstanbul

**150116006 - Yunus Yıldırım (GrRep)**

**150117056 - Ahmet Emirhan Bakkal**

**150118016 - Muhammet Raşit Ayaz**