CSE 204 Database Systems Assignment 2 Solutions

Yahya Efe Kuruçay - 20220808005 - docs.efekurucay.com|dbms|assignment2-efekurucay.com

1. Create the ve (5) tables including appropriate column types and necessary constraints.- 20 points

Database Schema

Table: Teacher

```
CREATE TABLE `dbassignment2`.`teacher` (
  `TeacherID` INT NOT NULL,
  `Name` VARCHAR(50) NOT NULL,
  `Surname` VARCHAR(50) NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `CV` TEXT NULL,
  PRIMARY KEY (`TeacherID`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE);
```

Table: Course

```
CREATE TABLE Course (
    Department VARCHAR(50) NOT NULL,
    Code VARCHAR(10) NOT NULL,
    Title VARCHAR(100) NOT NULL,
    AKTS INT NOT NULL,
    PRIMARY KEY (Department, Code)
);
```

.

.

Table: CourseSemester

```
CREATE TABLE CourseSemester (
    Department VARCHAR(50) NOT NULL,
    Code VARCHAR(10) NOT NULL,
    Year INT NOT NULL,
    TeacherID INT ,
    PRIMARY KEY (Department, Code, Year),
    FOREIGN KEY (Department, Code) REFERENCES Course(Department, Code) ON
DELETE CASCADE,
    FOREIGN KEY (TeacherID) REFERENCES Teacher(TeacherID) ON DELETE SET NULL
);
```

Table: Student

```
CREATE TABLE Student (
    StudentID INT AUTO_INCREMENT,
    Name VARCHAR(50) NOT NULL,
    Surname VARCHAR(50) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    PRIMARY KEY (StudentID)
);
```

Table: CourseRegistration

```
CREATE TABLE CourseRegistration (
    Department VARCHAR(50) NOT NULL,
    Code VARCHAR(10) NOT NULL,
    Year INT NOT NULL,
    StudentID INT NOT NULL,
    Score INT CHECK (Score BETWEEN 0 AND 100), -- Notlar 0-100 aralığında
olmalı
    PRIMARY KEY (Department, Code, Year, StudentID),
    FOREIGN KEY (Department, Code, Year) REFERENCES
CourseSemester(Department, Code, Year) ON DELETE CASCADE,
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID) ON DELETE CASCADE
);
```

.
.
.
.
.
.
.

2. Assume only ve (5) SQL statements were used to create the tables (one for each table). Give two (2) examples of one of the tables that must be created before one of the other ones. Explain why.- **15 points**

Table Creation Order:

1. `Teacher` table must be created before `CourseSemester` because `CourseSemester` has a foreign key referencing `Teacher`.
2. `Student` table must be created before `CourseRegistration` because `CourseRegistration` has a foreign key referencing `Student`.

3. Choose one of your responses for question 2 and explain how you could create table 2 before table 1 by using an extra SQL statement. Explain how the SQL statements would be changed.- 15 points

Creating CourseSemester before Teacher

We can create `CourseSemester` first without the foreign key constraint, then alter the table to add the constraint later:

```
CREATE TABLE CourseSemester (
    Department VARCHAR(50) NOT NULL,
    Code VARCHAR(10) NOT NULL,
    Year INT NOT NULL,
    TeacherID INT ,
    PRIMARY KEY (Department, Code, Year),
);

ALTER TABLE CourseSemester
ADD CONSTRAINT fk_Teacher
FOREIGN KEY (TeacherID) REFERENCES Teacher(TeacherID)
ON DELETE SET NULL;
```

.
.
.
.
.
.
.

Example Data for each Table

EXAMPLE DATA WAS CREATED BY AN LLM

Note: Requested SQL statements is below after the sample data insertion section.

Teachers

```
INSERT INTO Teacher (TeacherID, Name, Surname, email, CV) VALUES
(1, 'Dr. Joseph', 'Ledet', 'joseph.ledet@university.com', 'Database
Research'),
(2, 'Prof. Dr. Yahya Efe', 'Kuruçay', 'contactprof@efekurucay.com', 'AI and
ML'),
(3, 'Dr. Ayşe', 'Yılmaz', 'ayse.yilmaz@university.com', 'Software
Engineering'),
(4, 'Prof. Dr. Mehmet', 'Demir', 'mehmet.demir@university.com', 'Cyber
Security'),
(5, 'Dr. Can', 'Ersoy', 'can.ersoy@university.com', 'Data Science'),
(6, 'Dr. Zeynep', 'Koç', 'zeynep.koc@university.com', 'Cloud Computing');
```

Courses

```
INSERT INTO Course (Department, Code, Title, AKTS) VALUES
('CSE', '204', 'Database Systems', 6),
('CSE', '101', 'Introduction to Programming', 5),
('CSE', '303', 'Machine Learning', 6),
('CSE', '307', 'Big Data Analytics', 6),
('CSE', '405', 'Cybersecurity Fundamentals', 5),
('MATH', '201', 'Discrete Mathematics', 4),
('EEE', '301', 'Digital Signal Processing', 5),
('EEE', '305', 'Microcontrollers', 5),
('MECH', '401', 'Thermodynamics', 5),
('PHYS', '101', 'General Physics I', 4);
```

CourseSemester

```
INSERT INTO CourseSemester (Department, Code, Year, TeacherID) VALUES
('CSE', '204', 2025, 1),
('CSE', '101', 2025, 2),
('CSE', '303', 2025, 3),
```

```
('CSE', '307', 2025, 5),
('CSE', '405', 2025, 4),
('MATH', '201', 2025, 6),
('EEE', '301', 2025, 2),
('EEE', '305', 2025, 3),
('MECH', '401', 2025, 4),
('PHYS', '101', 2025, 6);
```

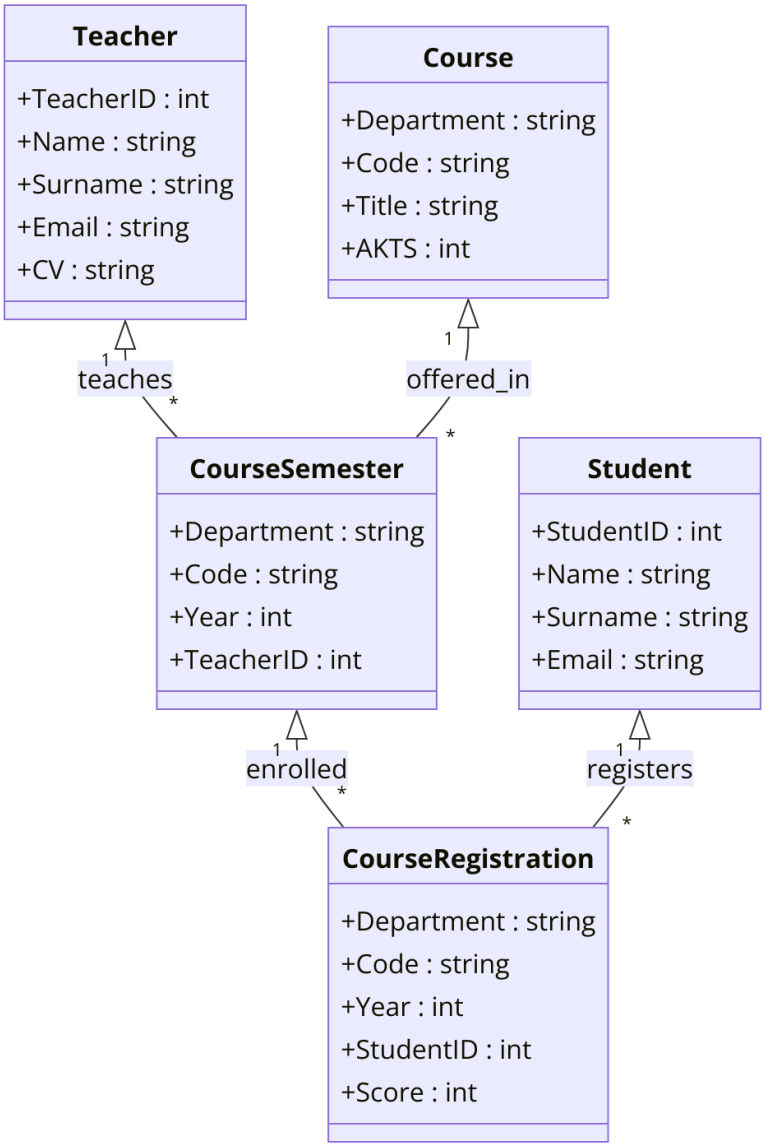
Students

```
INSERT INTO Student (StudentID, Name, Surname, Email) VALUES
(1, 'Yahya Efe', 'Kuruçay', 'contact@efekurucay.com'),
(2, 'Cem', 'Yılmaz', 'cemyilmaz@email.com'),
(3, 'Elif', 'Arslan', 'elif.arslan@university.com'),
(4, 'Mehmet', 'Öztürk', 'mehmet.ozturk@university.com'),
(5, 'Ayşe', 'Kara', 'ayse.kara@university.com'),
(6, 'Murat', 'Tuna', 'murat.tuna@university.com'),
(7, 'Zeynep', 'Aydın', 'zeynep.aydin@university.com'),
(8, 'Ahmet', 'Demir', 'ahmet.demir@university.com'),
(9, 'Fatma', 'Çelik', 'fatma.celik@university.com'),
(10, 'Emre', 'Şahin', 'emre.sahin@university.com');
```

CourseRegistration

```
INSERT INTO CourseRegistration (Department, Code, Year, StudentID, Score)
VALUES
('CSE', '204', 2025, 1, 85),
('CSE', '101', 2025, 2, 90),
('CSE', '303', 2025, 3, 88),
('CSE', '307', 2025, 4, 92),
('CSE', '405', 2025, 5, 78),
('MATH', '201', 2025, 6, 70),
('EEE', '301', 2025, 7, 95),
('EEE', '305', 2025, 8, 82),
('MECH', '401', 2025, 9, 76),
('PHYS', '101', 2025, 10, 89),
('CSE', '204', 2025, 2, 60),
('CSE', '204', 2025, 3, 75),
('CSE', '101', 2025, 4, 80),
('MATH', '201', 2025, 5, 85),
('CSE', '303', 2025, 6, 79),
('EEE', '301', 2025, 8, 67),
```

```
('EEE', '305', 2025, 9, 88),
('MECH', '401', 2025, 10, 74),
('PHYS', '101', 2025, 1, 81),
('CSE', '405', 2025, 7, 92);
```



Write SQL statements for the following questions. Each question has a num ber in parentheses. For full credit, you must provide that many substantially di erent SQL statements that would have the same result.

4. List all student details. (2)- 10 points

```
SELECT * FROM Student;
SELECT StudentID, Name, Surname, Email FROM Student;
```

StudentID	Name	Surname	Email
1	Yahya Efe	Kuruçay	contact@efekurucay.com
2	Cem	Yılmaz	cemyilmaz@email.com
3	Elif	Arslan	elif.arslan@university.com
4	Mehmet	Öztürk	mehmet.ozturk@university.com
5	Ayşe	Kara	ayse.kara@university.com
6	Murat	Tuna	murat.tuna@university.com
7	Zeynep	Aydın	zeynep.aydin@university.com
8	Ahmet	Demir	ahmet.demir@university.com
9	Fatma	Çelik	fatma.celik@university.com
10	Emre	Şahin	emre.sahin@university.com

5. List all teacher names and surnames. (1)- 5 points

```sql
SELECT Name, Surname FROM Teacher;
```

Name	Surname
Dr. Joseph	Ledet
Prof. Dr. Yahya Efe	Kuruçay
Dr. Ayşe	Yılmaz
Prof. Dr. Mehmet	Demir
Dr. Can	Ersoy
Dr. Zeynep	Koç

6. List the department and codes of all courses offered in 2025. (2)- 10 points

```sql
SELECT Department, Code FROM CourseSemester WHERE Year = 2025;
SELECT DISTINCT Department, Code FROM CourseSemester WHERE Year = 2025;
```

	Department	Code
▶	CSE	204
	CSE	101
	EEE	301
	CSE	303
	EEE	305
	CSE	405
	MECH	401
	CSE	307
	MATH	201
	PHYS	101

	Department	Code
▶	CSE	101
	CSE	204
	CSE	303
	CSE	307
	CSE	405
	EEE	301
	EEE	305
	MATH	201
	MECH	401
	PHYS	101

7. List all course titles that contain the word database and are o ered in 2025. (2)- 10 points

```
SELECT Title FROM Course WHERE Title LIKE '%database%' AND Code IN (SELECT
Code FROM CourseSemester WHERE Year = 2025);
SELECT c.Title FROM Course c JOIN CourseSemester cs ON c.Code = cs.Code AND
cs.Year = 2025 WHERE c.Title LIKE '%database%';
```

	Title
▶	Database Systems

8. List all book courses (department, code, title) that have no students registered. (1)- 5 points

```
SELECT c.Department, c.Code, c.Title FROM Course c LEFT JOIN
CourseRegistration cr ON c.Code = cr.Code WHERE cr.StudentID IS NULL;
```

	Department	Code	Title
▶	NSR	645	No Student Registered This Course

9. Remove all courses that have not been offered since 2020 from teh database. (1)- 5 points

```
DELETE FROM Course WHERE Code NOT IN (SELECT Code FROM CourseSemester WHERE Year >= 2020);
```

10. List course department and code along with how many students area in each course? (1)- 5 points

```
SELECT Department, Code, COUNT(StudentID) AS StudentCount FROM CourseRegistration GROUP BY Department, Code;
```

Department	Code	StudentCount
CSE	101	2
CSE	204	3
CSE	303	2
CSE	307	1
CSE	405	2
EEE	301	2
EEE	305	2
MATH	201	2
MECH	401	2
PHYS	101	2

11. How many courses are being taught by Joseph Ledet in 2025? (1)- 5 points

```
SELECT COUNT(*) FROM CourseSemester WHERE TeacherID = (SELECT TeacherID FROM Teacher WHERE Name = 'Dr. Joseph' AND Surname = 'Ledet') AND Year = 2025;
```

COUNT(*)
1

12. List the names of students who are not registered for any courses. (1)- 5 points

```
SELECT Name FROM Student WHERE StudentID NOT IN (SELECT StudentID FROM CourseRegistration);
```

Name
Free

13. Give a list of students who are currently taking more than 30 AKTS and also list which courses they are registered for. (1)- 5 points

```sql
SELECT s.StudentID, s.Name, s.Surname, SUM(c.AKTS) AS Total_AKTS
FROM Student s
JOIN CourseRegistration cr ON s.StudentID = cr.StudentID
JOIN Course c ON cr.Code = c.Code
GROUP BY s.StudentID
HAVING Total_AKTS > 30;
```

StudentID	Name	Surname	Total_AKTS
1	Yahya Efe	Kuruçay	47

14. Assume you are registered for CSE 204 in 2025. Change your score for that course to be 100? (1)- 5 points

```sql
UPDATE CourseRegistration SET Score = 100 WHERE Code = 'CSE204' AND Year =
2025 AND StudentID = (SELECT StudentID FROM Student WHERE Email =
'contact@efekurucay.com');
```

Department	Code	Year	StudentID	Score
CSE	101	2025	2	90
CSE	101	2025	4	80
CSE	204	2025	1	85
CSE	204	2025	2	60
CSE	204	2025	3	75

Department	Code	Year	StudentID	Score
CSE	101	2025	2	90
CSE	101	2025	4	80
CSE	204	2025	1	100
CSE	204	2025	2	60
CSE	204	2025	3	75
CSE	303	2025	3	88
CSE	303	2025	6	79

15. List the courses that are being taken by a student with email address cemyilmaz@email.com (1)- 5 points

```sql
SELECT c.Department, c.Code, c.Title FROM Course c JOIN CourseRegistration
cr ON c.Code = cr.Code JOIN Student s ON cr.StudentID = s.StudentID WHERE
s.Email = 'cemyilmaz@email.com';
```

	Department	Code	Title
▶	CSE	101	Introduction to Programming
	CSE	204	Database Systems
	PHYS	101	General Physics I

16. For each course that has been offered at least 5 times, list the teachers who have taught them. (1)- 10 points

```
SELECT t.Name, t.Surname
FROM Teacher t
JOIN CourseSemester cs ON t.TeacherID = cs.TeacherID
GROUP BY cs.Code, t.TeacherID
HAVING COUNT(cs.Year) >= 5;
```

	Name	Surname

17. List student names, surnames and total AKTS for 2025 for each student. (1)- 5 points

```
SELECT s.Name, s.Surname, SUM(c.AKTS) AS Total_AKTS FROM Student s JOIN
CourseRegistration cr ON s.StudentID = cr.StudentID JOIN Course c ON c.Code
= cr.Code WHERE cr.Year = 2025 GROUP BY s.StudentID;
```

	Name	Surname	Total_AKTS
▶	Emre	Şahin	14
	Mehmet	Öztürk	15
	Cem	Yılmaz	15
	Yahya Efe	Kuruçay	47
	Elif	Arslan	12
	Murat	Tuna	10
	Zeynep	Aydın	10
	Ayşe	Kara	9
	Ahmet	Demir	10
	Fatma	Çelik	10

18. There could be additional columns on these tables. Use SQL to add columns to two di erent tables that would seem reasonable. (2)- 10 points

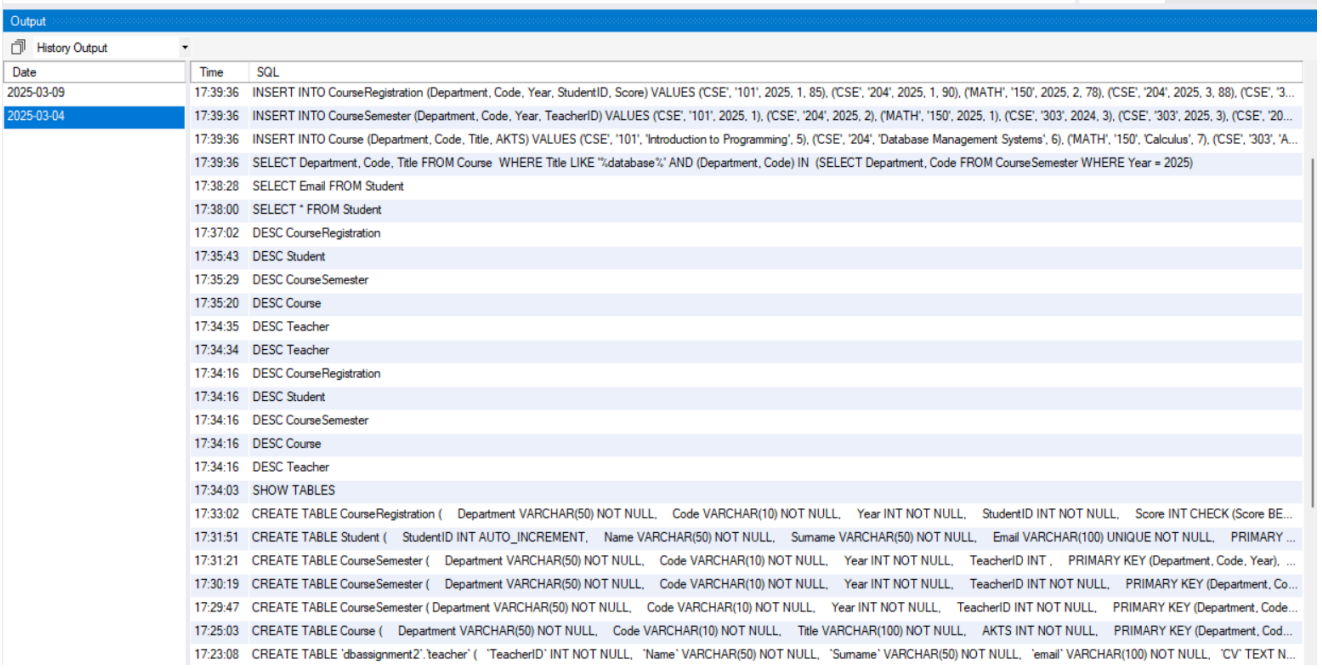
```
ALTER TABLE Teacher ADD COLUMN Office VARCHAR(50);
```

```sql
ALTER TABLE Student ADD COLUMN DateOfBirth DATE;
```

Field	Type	Null	Key	Default	Extra
TeacherID	int	NO	PRI	NULL	
Name	varchar(50)	NO		NULL	
Surname	varchar(50)	NO		NULL	
email	varchar(100)	NO	UNI	NULL	
CV	text	YES		NULL	
Office	varchar(50)	YES		NULL	

Field	Type	Null	Key	Default	Extra
StudentID	int	NO	PRI	NULL	auto_increment
Name	varchar(50)	NO		NULL	
Surname	varchar(50)	NO		NULL	
Email	varchar(100)	NO	UNI	NULL	
DateOfBirth	date	YES		NULL	

I also have included the MySQL Workbench history output for your review



Output

History Output

Date	Time	SQL
2025-03-09	17:39:36	INSERT INTO CourseRegistration (Department, Code, Year, StudentID, Score) VALUES ('CSE', '101', 2025, 1, 85), ('CSE', '204', 2025, 1, 90), ('MATH', '150', 2025, 2, 78), ('CSE', '204', 2025, 3, 88), ('CSE', '3...
2025-03-04	17:39:36	INSERT INTO CourseSemester (Department, Code, Year, TeacherID) VALUES ('CSE', '101', 2025, 1), ('CSE', '204', 2025, 2), ('MATH', '150', 2025, 1), ('CSE', '303', 2024, 3), ('CSE', '303', 2025, 3), ('CSE', '20...
	17:39:36	INSERT INTO Course (Department, Code, Title, AKTS) VALUES ('CSE', '101', 'Introduction to Programming', 5), ('CSE', '204', 'Database Management Systems', 6), ('MATH', '150', 'Calculus', 7), ('CSE', '303', 'A...
	17:39:36	SELECT Department, Code, Title FROM Course WHERE Title LIKE '%database%' AND (Department, Code) IN (SELECT Department, Code FROM CourseSemester WHERE Year = 2025)
	17:38:28	SELECT Email FROM Student
	17:38:00	SELECT * FROM Student
	17:37:02	DESC CourseRegistration
	17:35:43	DESC Student
	17:35:29	DESC CourseSemester
	17:35:20	DESC Course
	17:34:35	DESC Teacher
	17:34:34	DESC Teacher
	17:34:16	DESC CourseRegistration
	17:34:16	DESC Student
	17:34:16	DESC CourseSemester
	17:34:16	DESC Course
	17:34:16	DESC Teacher
	17:34:03	SHOW TABLES
	17:33:02	CREATE TABLE CourseRegistration (Department VARCHAR(50) NOT NULL, Code VARCHAR(10) NOT NULL, Year INT NOT NULL, StudentID INT NOT NULL, Score INT CHECK (Score BE...
	17:31:51	CREATE TABLE Student (StudentID INT AUTO_INCREMENT, Name VARCHAR(50) NOT NULL, Surname VARCHAR(50) NOT NULL, Email VARCHAR(100) UNIQUE NOT NULL, PRIMARY ...
	17:31:21	CREATE TABLE CourseSemester (Department VARCHAR(50) NOT NULL, Code VARCHAR(10) NOT NULL, Year INT NOT NULL, TeacherID INT , PRIMARY KEY (Department, Code, Year), ...
	17:30:19	CREATE TABLE CourseSemester (Department VARCHAR(50) NOT NULL, Code VARCHAR(10) NOT NULL, Year INT NOT NULL, TeacherID INT NOT NULL, PRIMARY KEY (Department, Co...
	17:29:47	CREATE TABLE CourseSemester (Department VARCHAR(50) NOT NULL, Code VARCHAR(10) NOT NULL, Year INT NOT NULL, TeacherID INT NOT NULL, PRIMARY KEY (Department, Code...
	17:25:03	CREATE TABLE Course (Department VARCHAR(50) NOT NULL, Code VARCHAR(10) NOT NULL, Title VARCHAR(100) NOT NULL, AKTS INT NOT NULL, PRIMARY KEY (Department, Cod...
	17:23:08	CREATE TABLE `dbassignment2`.`teacher` (`TeacherID` INT NOT NULL, `Name` VARCHAR(50) NOT NULL, `Surname` VARCHAR(50) NOT NULL, `email` VARCHAR(100) NOT NULL, `CV` TEXT N...