DATABASE SYSTEM 1

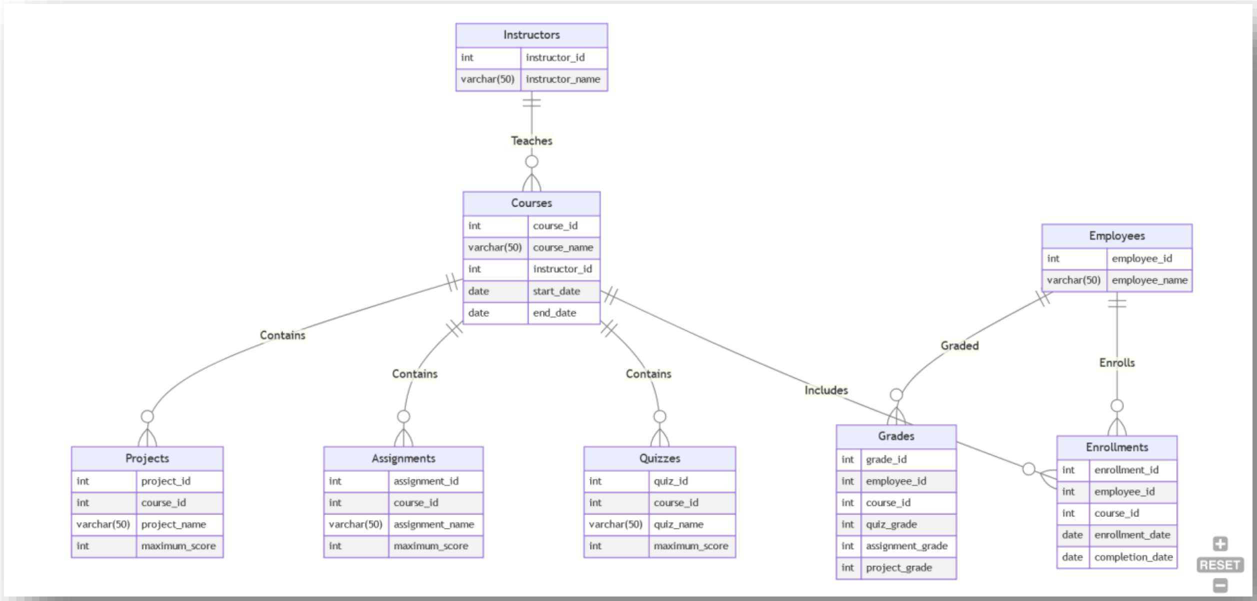
SYED SHAH HUSSIAN (FA21-BSE-172)
EISSA MASOOD (FA21-BSE-086)
EBADAT NISSA (FA21-BSE-085)

PROJECT

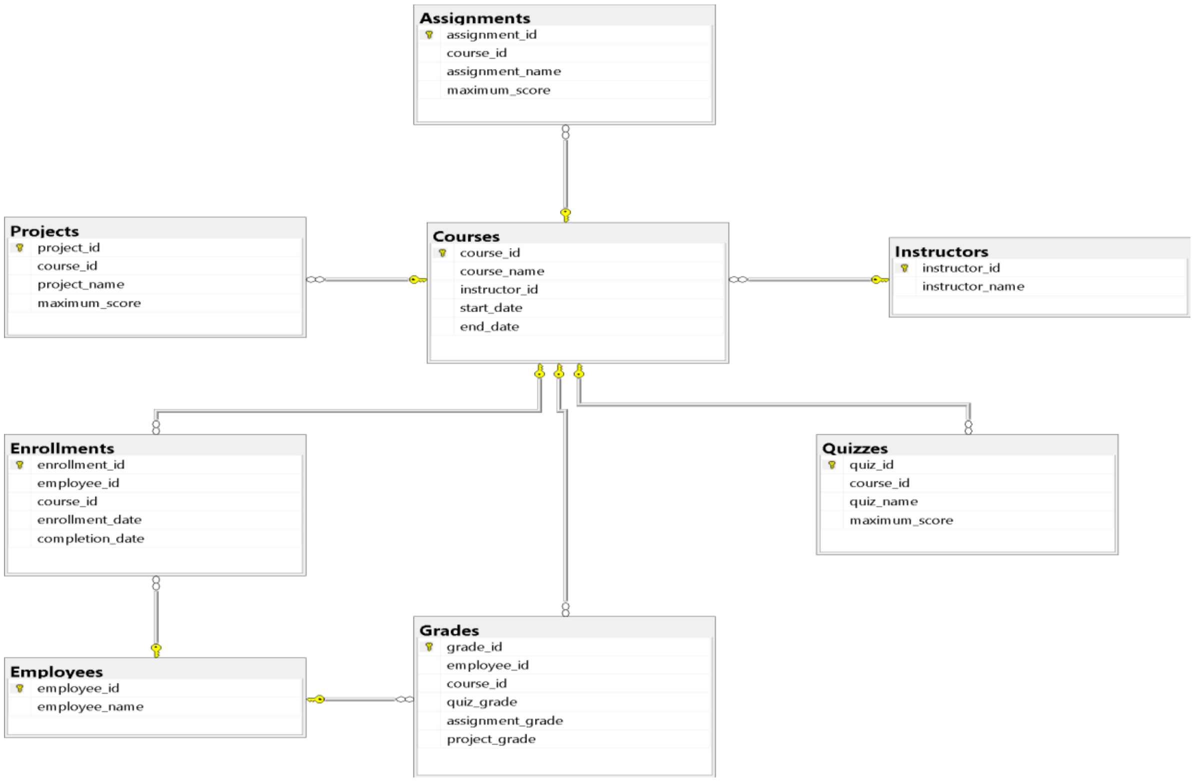
SUBMITTED TO: MA'AM HUMAIRA JABEEN

ERD

CARDINALITY:



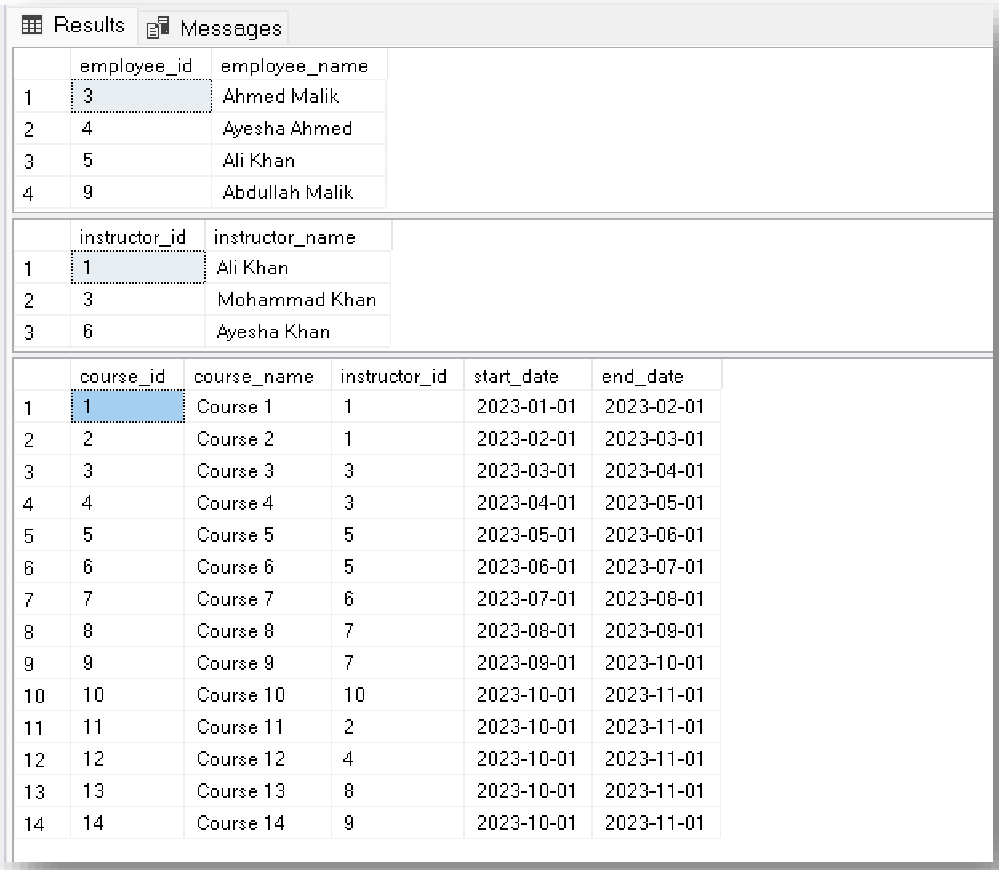
PRIMARY KEY + FOREIGY KEYS:



```sql
-- Create the Instructors table

CREATE TABLE Instructors (
  instructor_id INT PRIMARY KEY,
  instructor_name VARCHAR(50)
);


INSERT INTO Instructors (instructor_id, instructor_name)
VALUES
  (1, 'Ali Khan'),
  (2, 'Fatima Ahmed'),
  (3, 'Mohammad Khan'),
  (4, 'Sara Malik'),
  (5, 'Ahmed Ali'),
  (6, 'Ayesha Khan'),
  (7, 'Usman Ahmed'),
  (8, 'Farah Hussain'),
  (9, 'Abdullah Malik'),
  (10, 'Sadia Ahmed');


-- Create the Employees table
CREATE TABLE Employees (
  employee_id INT PRIMARY KEY,
  employee_name VARCHAR(50)
);

INSERT INTO Employees (employee_id, employee_name)
VALUES
  (1, 'Mohammad Khan'),
  (2, 'Fatima Ali'),
  (3, 'Ahmed Malik'),
  (4, 'Ayesha Ahmed'),
  (5, 'Ali Khan'),
  (6, 'Sara Malik'),
  (7, 'Usman Ahmed'),
  (8, 'Farah Khan'),
  (9, 'Abdullah Malik'),
  (10, 'Sadia Ahmed');
```

```sql
-- Create the Courses table
CREATE TABLE Courses (
  course_id INT PRIMARY KEY,
  course_name VARCHAR(50),
  instructor_id INT,
  start_date DATE,
  end_date DATE,
  FOREIGN KEY (instructor_id) REFERENCES Instructors(instructor_id)
);

INSERT INTO Courses (course_id, course_name, instructor_id,
start_date, end_date)
VALUES
  (1, 'Course 1', 1, '2023-01-01', '2023-02-01'),
  (2, 'Course 2', 1, '2023-02-01', '2023-03-01'),
  (3, 'Course 3', 3, '2023-03-01', '2023-04-01'),
  (4, 'Course 4', 3, '2023-04-01', '2023-05-01'),
  (5, 'Course 5', 5, '2023-05-01', '2023-06-01'),
  (6, 'Course 6', 5, '2023-06-01', '2023-07-01'),
  (7, 'Course 7', 6, '2023-07-01', '2023-08-01'),
  (8, 'Course 8', 7, '2023-08-01', '2023-09-01'),
  (9, 'Course 9', 7, '2023-09-01', '2023-10-01'),
  (10, 'Course 10', 10, '2023-10-01', '2023-11-01'),
  (11, 'Course 11', 2, '2023-10-01', '2023-11-01'),
  (12, 'Course 12', 4, '2023-10-01', '2023-11-01'),
  (13, 'Course 13', 8, '2023-10-01', '2023-11-01'),
  (14, 'Course 14', 9, '2023-10-01', '2023-11-01');

-- Create the Enrollments table
CREATE TABLE Enrollments (
  enrollment_id INT PRIMARY KEY,
  employee_id INT,
  course_id INT,
  enrollment_date DATE,
  completion_date DATE,
  FOREIGN KEY (employee_id) REFERENCES Employees(employee_id),
  FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);

INSERT INTO Enrollments (enrollment_id, employee_id, course_id,
enrollment_date, completion_date)
VALUES
  (1, 1, 1, '2023-01-01', '2023-01-15'),
  (2, 2, 2, '2023-02-01', '2023-02-15'),
  (3, 3, 3, '2023-03-01', '2023-03-15'),
  (4, 4, 4, '2023-04-01', '2023-04-15'),
  (5, 5, 5, '2023-05-01', '2023-05-15'),
  (6, 6, 6, '2023-06-01', '2023-06-15'),
  (7, 7, 7, '2023-07-01', '2023-07-15'),
  (8, 8, 8, '2023-08-01', '2023-08-15'),
  (9, 9, 9, '2023-09-01', '2023-09-15'),
  (10, 10, 10, '2023-10-01', '2023-10-15');


-- Create the Quizzes table
CREATE TABLE Quizzes (
  quiz_id INT PRIMARY KEY,
  course_id INT,
  quiz_name VARCHAR(50),
  maximum_score INT,
  FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);

INSERT INTO Quizzes (quiz_id, course_id, quiz_name, maximum_score)
VALUES
  (1, 1, 'Quiz 1', 10),
  (2, 2, 'Quiz 2', 10),
  (3, 3, 'Quiz 3', 10),
  (4, 4, 'Quiz 4', 10),
  (5, 5, 'Quiz 5', 10),
  (6, 6, 'Quiz 6', 10),
  (7, 7, 'Quiz 7', 10),
  (8, 8, 'Quiz 8', 10),
  (9, 9, 'Quiz 9', 10),
  (10, 10, 'Quiz 10', 10);

-- Create the Assignments table
CREATE TABLE Assignments (
  assignment_id INT PRIMARY KEY,
  course_id INT,
  assignment_name VARCHAR(50),
  maximum_score INT,
  FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);

INSERT INTO Assignments (assignment_id, course_id, assignment_name,
maximum_score)
VALUES
  (1, 1, 'Assignment 1', 10),
  (2, 2, 'Assignment 2', 10),
  (3, 3, 'Assignment 3', 10),
  (4, 4, 'Assignment 4', 10),
  (5, 5, 'Assignment 5', 10),
  (6, 6, 'Assignment 6', 10),
  (7, 7, 'Assignment 7', 10),
  (8, 8, 'Assignment 8', 10),
  (9, 9, 'Assignment 9', 10),
  (10, 10, 'Assignment 10', 10);



-- Create the Projects table
CREATE TABLE Projects (
  project_id INT PRIMARY KEY,
  course_id INT,
  project_name VARCHAR(50),
  maximum_score INT,
  FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);


INSERT INTO Projects (project_id, course_id, project_name,
maximum_score)
VALUES
  (1, 1, 'Project 1', 100),
  (2, 2, 'Project 2', 100),
  (3, 3, 'Project 3', 100),
  (4, 4, 'Project 4', 100),
  (5, 5, 'Project 5', 100),
  (6, 6, 'Project 6', 100),
  (7, 7, 'Project 7', 100),
  (8, 8, 'Project 8', 100),
  (9, 9, 'Project 9', 100),
  (10, 10, 'Project 10', 100);


-- Create the Grades table
CREATE TABLE Grades (
  grade_id INT PRIMARY KEY,
  employee_id INT,
  course_id INT,
  quiz_grade INT,
  assignment_grade INT,
  project_grade INT,
  FOREIGN KEY (employee_id) REFERENCES Employees(employee_id),
  FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);
```

```
INSERT INTO Grades (grade_id, employee_id, course_id, quiz_grade,
assignment_grade, project_grade)
VALUES
  (1, 1, 1, 8, 9, 95),
  (2, 2, 2, 9, 9, 85),
  (3, 3, 3, 10, 10, 90),
  (4, 4, 4, 7, 8, 80),
  (5, 5, 5, 8, 5, 75),
  (6, 6, 6, 9, 9, 70),
  (7, 7, 7, 10, 5, 85),
  (8, 8, 8, 7, 8, 90),
  (9, 9, 9, 8, 8, 95),
  (10, 10, 10, 9, 90, 100);

--a

SELECT * FROM Employees WHERE employee_name LIKE 'A%';
SELECT * FROM Instructors WHERE instructor_name LIKE '%Khan';
SELECT * FROM Courses WHERE course_name LIKE '%Course%';
```

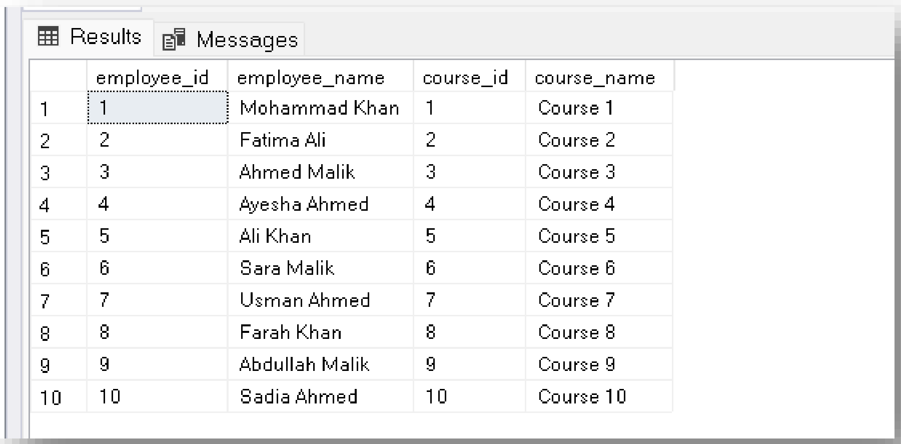

Results Messages

	employee_id	employee_name
1	3	Ahmed Malik
2	4	Ayesha Ahmed
3	5	Ali Khan
4	9	Abdullah Malik

	instructor_id	instructor_name
1	1	Ali Khan
2	3	Mohammad Khan
3	6	Ayesha Khan

	course_id	course_name	instructor_id	start_date	end_date
1	1	Course 1	1	2023-01-01	2023-02-01
2	2	Course 2	1	2023-02-01	2023-03-01
3	3	Course 3	3	2023-03-01	2023-04-01
4	4	Course 4	3	2023-04-01	2023-05-01
5	5	Course 5	5	2023-05-01	2023-06-01
6	6	Course 6	5	2023-06-01	2023-07-01
7	7	Course 7	6	2023-07-01	2023-08-01
8	8	Course 8	7	2023-08-01	2023-09-01
9	9	Course 9	7	2023-09-01	2023-10-01
10	10	Course 10	10	2023-10-01	2023-11-01
11	11	Course 11	2	2023-10-01	2023-11-01
12	12	Course 12	4	2023-10-01	2023-11-01
13	13	Course 13	8	2023-10-01	2023-11-01
14	14	Course 14	9	2023-10-01	2023-11-01

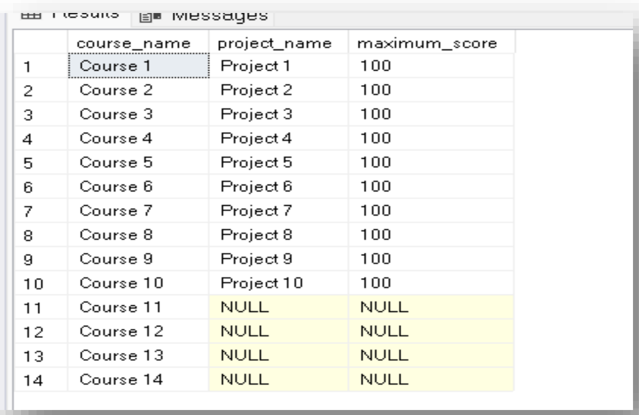
```
--b

--Query to retrieve the enrollment details of employees along with
their corresponding course information:

SELECT E.employee_id, E.employee_name, C.course_id, C.course_name
FROM Employees E
INNER JOIN Enrollments EN ON E.employee_id = EN.employee_id
INNER JOIN Courses C ON EN.course_id = C.course_id;
```



Results Messages

	employee_id	employee_name	course_id	course_name
1	1	Mohammad Khan	1	Course 1
2	2	Fatima Ali	2	Course 2
3	3	Ahmed Malik	3	Course 3
4	4	Ayesha Ahmed	4	Course 4
5	5	Ali Khan	5	Course 5
6	6	Sara Malik	6	Course 6
7	7	Usman Ahmed	7	Course 7
8	8	Farah Khan	8	Course 8
9	9	Abdullah Malik	9	Course 9
10	10	Sadia Ahmed	10	Course 10

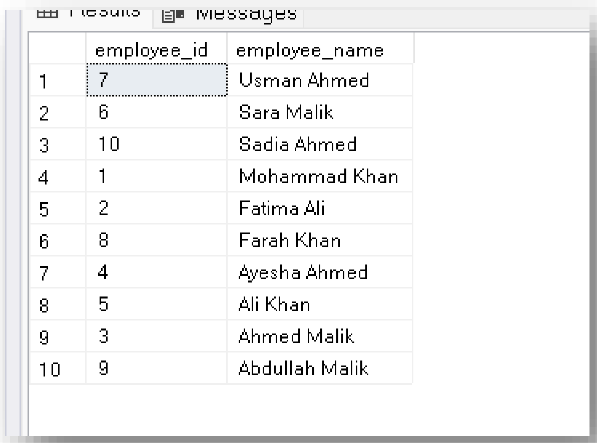
```
--Query to retrieve the project names and maximum scores of projects
associated with each course:

SELECT C.course_name, P.project_name, P.maximum_score
FROM Courses C
LEFT JOIN Projects P ON C.course_id = P.course_id;
```



Results Messages

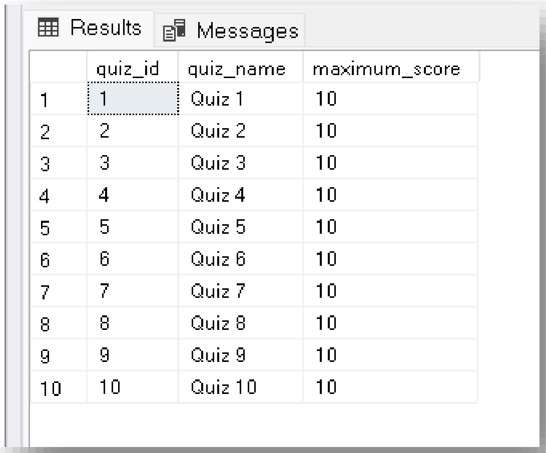
	course_name	project_name	maximum_score
1	Course 1	Project 1	100
2	Course 2	Project 2	100
3	Course 3	Project 3	100
4	Course 4	Project 4	100
5	Course 5	Project 5	100
6	Course 6	Project 6	100
7	Course 7	Project 7	100
8	Course 8	Project 8	100
9	Course 9	Project 9	100
10	Course 10	Project 10	100
11	Course 11	NULL	NULL
12	Course 12	NULL	NULL
13	Course 13	NULL	NULL
14	Course 14	NULL	NULL

```
--c

--Get employees sorted by their names in descending order:

SELECT employee_id, employee_name
FROM Employees
ORDER BY employee_name DESC;
```



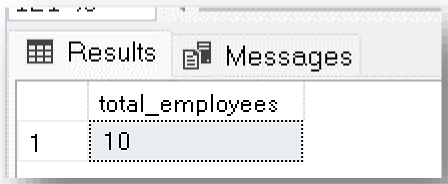
	employee_id	employee_name
1	7	Usman Ahmed
2	6	Sara Malik
3	10	Sadia Ahmed
4	1	Mohammad Khan
5	2	Fatima Ali
6	8	Farah Khan
7	4	Ayesha Ahmed
8	5	Ali Khan
9	3	Ahmed Malik
10	9	Abdullah Malik

```
--Find quizzes sorted by the maximum score in descending order:

SELECT quiz_id, quiz_name, maximum_score
FROM Quizzes
ORDER BY maximum_score DESC;
```



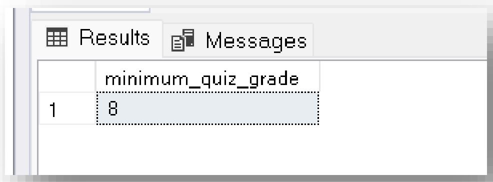
	quiz_id	quiz_name	maximum_score
1	1	Quiz 1	10
2	2	Quiz 2	10
3	3	Quiz 3	10
4	4	Quiz 4	10
5	5	Quiz 5	10
6	6	Quiz 6	10
7	7	Quiz 7	10
8	8	Quiz 8	10
9	9	Quiz 9	10
10	10	Quiz 10	10

```
--d

--Query to calculate the total number of employees:

SELECT COUNT(*) AS total_employees
FROM Employees;
```


Results Messages

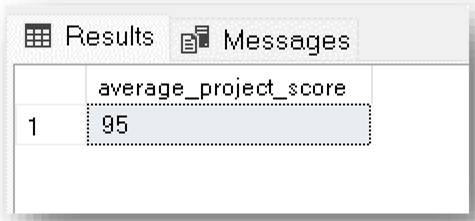
```
   total_employees
1  10
```

```
--Query to find the minimum quiz grade in a specific course:

SELECT MIN(quiz_grade) AS minimum_quiz_grade
FROM Grades
WHERE course_id = 1;
```


Results Messages

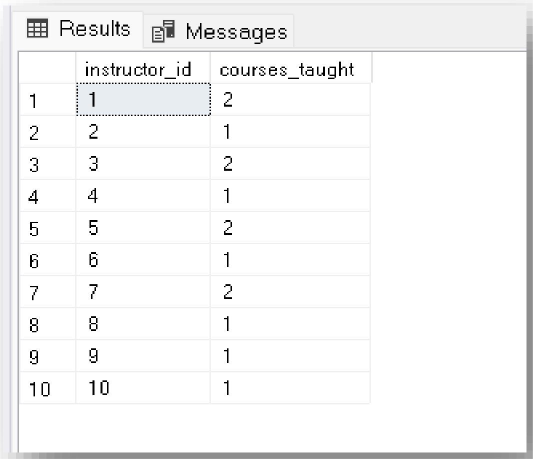
```
   minimum_quiz_grade
1  8
```

```
--Query to calculate the average project score for a given employee:

SELECT AVG(project_grade) AS average_project_score
FROM Grades
WHERE employee_id = 1;
```


Results Messages

```
   average_project_score
1  95
```

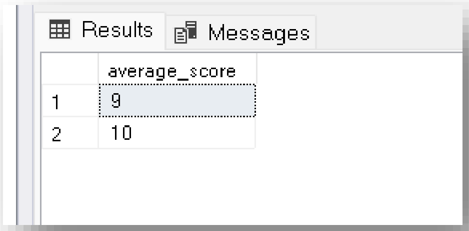
```
--e

--Query to find instructors who have taught at least 1 courses:

SELECT instructor_id, COUNT(*) AS courses_taught
FROM Courses
GROUP BY instructor_id
HAVING COUNT(*) >= 1;
```



Results | Messages

	instructor_id	courses_taught
1	1	2
2	2	1
3	3	2
4	4	1
5	5	2
6	6	1
7	7	2
8	8	1
9	9	1
10	10	1

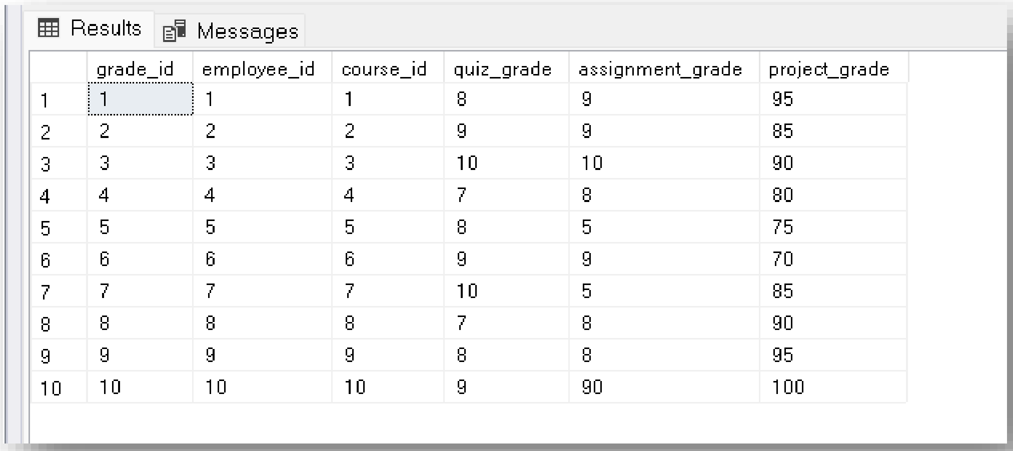
```
--Query to find quizzes with an average score higher than 8:

SELECT  AVG(quiz_grade) AS average_score
FROM Grades
GROUP BY quiz_grade
HAVING AVG(quiz_grade) > 8;
```



Results | Messages

	average_score
1	9
2	10

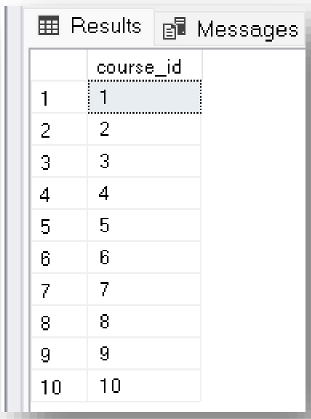
```
--f

--Query to retrieve employees who have a Project grade greater than or
equal to all quiz grades:
SELECT *
FROM Grades
WHERE project_grade >= ALL (SELECT quiz_grade FROM Grades);
```



Results | Messages

	grade_id	employee_id	course_id	quiz_grade	assignment_grade	project_grade
1	1	1	1	8	9	95
2	2	2	2	9	9	85
3	3	3	3	10	10	90
4	4	4	4	7	8	80
5	5	5	5	8	5	75
6	6	6	6	9	9	70
7	7	7	7	10	5	85
8	8	8	8	7	8	90
9	9	9	9	8	8	95
10	10	10	10	9	90	100

```
--Query to retrieve courses where the maximum project score is lesser
than any quiz maximum score:

SELECT course_id
FROM Quizzes
WHERE maximum_score < ANY (SELECT maximum_score FROM Projects);
```

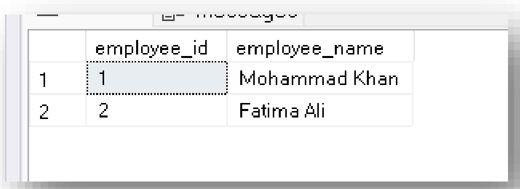


Results | Messages

	course_id
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

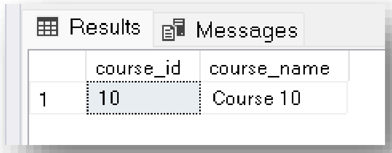
--g

```sql
--Query to retrieve employees who are enrolled in a course taught by
instructor 'Ali Khan':
SELECT employee_id, employee_name
FROM Employees
WHERE employee_id IN (
  SELECT employee_id
  FROM Enrollments
  WHERE course_id IN (
    SELECT course_id
    FROM Courses
    WHERE instructor_id = (
      SELECT instructor_id
      FROM Instructors
      WHERE instructor_name = 'Ali Khan'
    )
  )
);
```



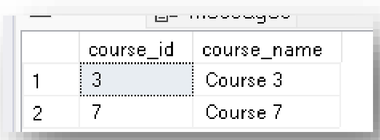
```
   employee_id  employee_name
1  1            Mohammad Khan
2  2            Fatima Ali
```

```sql
--Query to retrieve courses in which an employee  achieved a perfect
score (100) in the project:
SELECT course_id, course_name
FROM Courses
WHERE course_id IN (
  SELECT course_id
  FROM Grades
  WHERE project_grade = 100
);
```



```
Results  Messages
   course_id  course_name
1  10         Course 10
```

```
--Query to retrieve courses that have at least one employee with a
grade higher than 9 in any quiz:

SELECT course_id, course_name
FROM Courses
WHERE course_id IN (
  SELECT course_id
  FROM Grades
  WHERE quiz_grade > 9
);
```

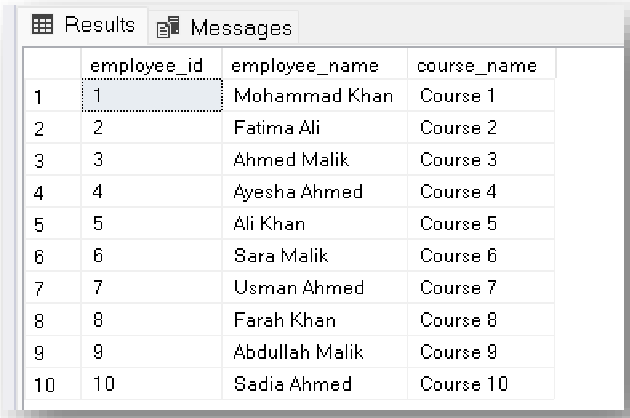


	course_id	course_name
1	3	Course 3
2	7	Course 7

--h

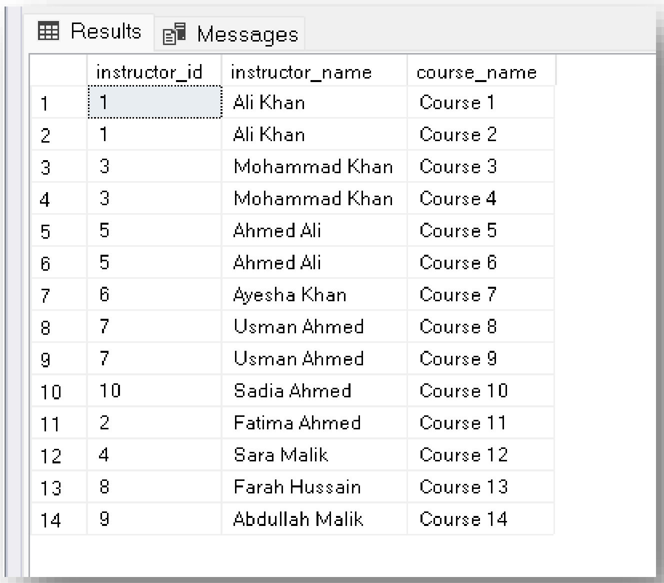
--SIMPLE JOIN

```
--Retrieve the employees and their corresponding course names:

SELECT E.employee_id, E.employee_name, C.course_name
FROM Employees E
JOIN Enrollments EN ON E.employee_id = EN.employee_id
JOIN Courses C ON EN.course_id = C.course_id;
```



	employee_id	employee_name	course_name
1	1	Mohammad Khan	Course 1
2	2	Fatima Ali	Course 2
3	3	Ahmed Malik	Course 3
4	4	Ayesha Ahmed	Course 4
5	5	Ali Khan	Course 5
6	6	Sara Malik	Course 6
7	7	Usman Ahmed	Course 7
8	8	Farah Khan	Course 8
9	9	Abdullah Malik	Course 9
10	10	Sadia Ahmed	Course 10

```sql
--Retrieve the instructors and the courses they teach:

SELECT I.instructor_id, I.instructor_name, C.course_name
FROM Instructors I
JOIN Courses C ON I.instructor_id = C.instructor_id;
```

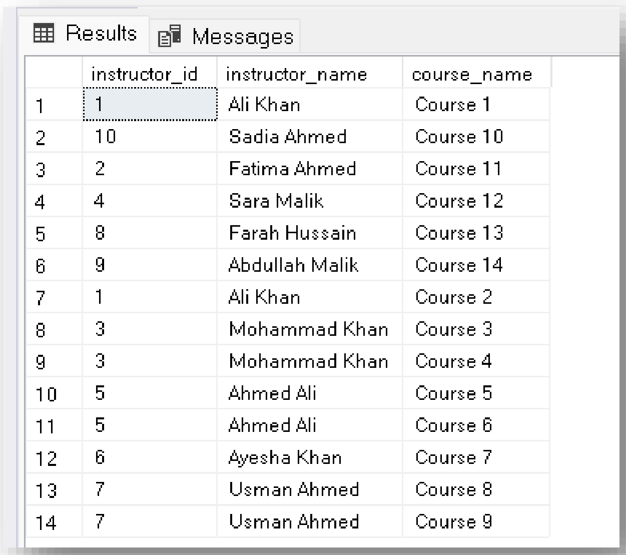


Results | Messages

	instructor_id	instructor_name	course_name
1	1	Ali Khan	Course 1
2	1	Ali Khan	Course 2
3	3	Mohammad Khan	Course 3
4	3	Mohammad Khan	Course 4
5	5	Ahmed Ali	Course 5
6	5	Ahmed Ali	Course 6
7	6	Ayesha Khan	Course 7
8	7	Usman Ahmed	Course 8
9	7	Usman Ahmed	Course 9
10	10	Sadia Ahmed	Course 10
11	2	Fatima Ahmed	Course 11
12	4	Sara Malik	Course 12
13	8	Farah Hussain	Course 13
14	9	Abdullah Malik	Course 14

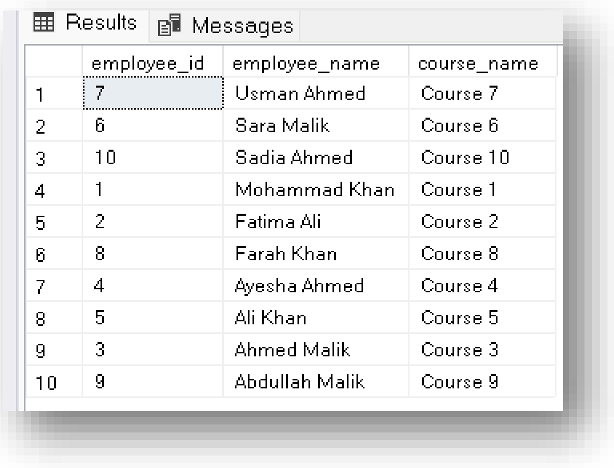
--SORTING A JOIN

```
--Retrieve the instructors and the courses they teach, sorted by the
course names in ascending order:

SELECT I.instructor_id, I.instructor_name, C.course_name
FROM Instructors I
JOIN Courses C ON I.instructor_id = C.instructor_id
ORDER BY C.course_name ASC;
```



	instructor_id	instructor_name	course_name
1	1	Ali Khan	Course 1
2	10	Sadia Ahmed	Course 10
3	2	Fatima Ahmed	Course 11
4	4	Sara Malik	Course 12
5	8	Farah Hussain	Course 13
6	9	Abdullah Malik	Course 14
7	1	Ali Khan	Course 2
8	3	Mohammad Khan	Course 3
9	3	Mohammad Khan	Course 4
10	5	Ahmed Ali	Course 5
11	5	Ahmed Ali	Course 6
12	6	Ayesha Khan	Course 7
13	7	Usman Ahmed	Course 8
14	7	Usman Ahmed	Course 9

```
--Retrieve the employees and their corresponding course names, sorted
by the employee names in descending order:

SELECT E.employee_id, E.employee_name, C.course_name
FROM Employees E
JOIN Enrollments EN ON E.employee_id = EN.employee_id
JOIN Courses C ON EN.course_id = C.course_id
ORDER BY E.employee_name DESC;
```

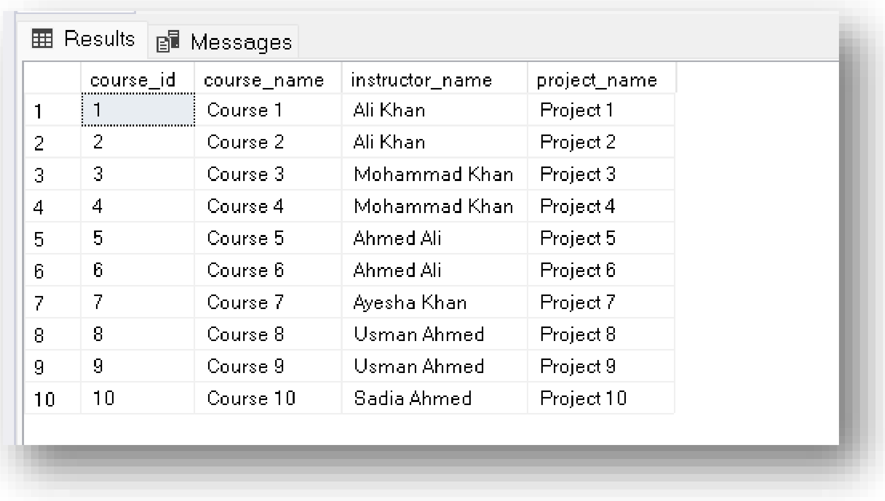


Results | Messages

	employee_id	employee_name	course_name
1	7	Usman Ahmed	Course 7
2	6	Sara Malik	Course 6
3	10	Sadia Ahmed	Course 10
4	1	Mohammad Khan	Course 1
5	2	Fatima Ali	Course 2
6	8	Farah Khan	Course 8
7	4	Ayesha Ahmed	Course 4
8	5	Ali Khan	Course 5
9	3	Ahmed Malik	Course 3
10	9	Abdullah Malik	Course 9

--THREE TABLE JOIN

```
--Retrieve the courses, their corresponding instructor names, and the
project names associated with each course:
SELECT C.course_id, C.course_name, I.instructor_name, P.project_name
FROM Courses C
JOIN Instructors I ON C.instructor_id = I.instructor_id
JOIN Projects P ON C.course_id = P.course_id;
```

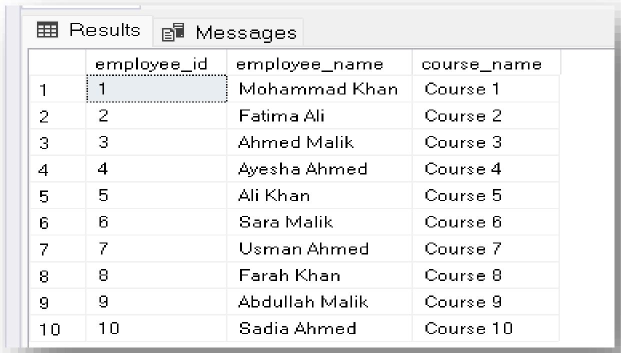


Results | Messages

	course_id	course_name	instructor_name	project_name
1	1	Course 1	Ali Khan	Project 1
2	2	Course 2	Ali Khan	Project 2
3	3	Course 3	Mohammad Khan	Project 3
4	4	Course 4	Mohammad Khan	Project 4
5	5	Course 5	Ahmed Ali	Project 5
6	6	Course 6	Ahmed Ali	Project 6
7	7	Course 7	Ayesha Khan	Project 7
8	8	Course 8	Usman Ahmed	Project 8
9	9	Course 9	Usman Ahmed	Project 9
10	10	Course 10	Sadia Ahmed	Project 10

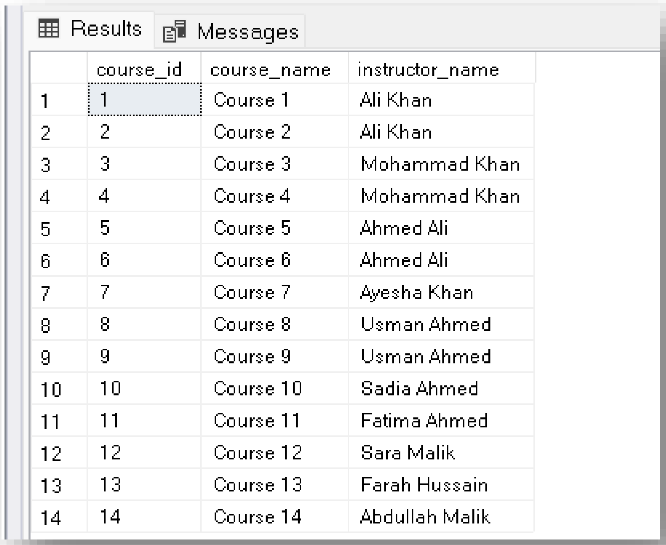
--OUTER JOIN

```
--Retrieve all employees and their corresponding course names,
including employees who haven't enrolled in any course:
SELECT E.employee_id, E.employee_name, C.course_name
FROM Employees E
LEFT JOIN Enrollments EN ON E.employee_id = EN.employee_id
LEFT JOIN Courses C ON EN.course_id = C.course_id;
```


Results | Messages

	employee_id	employee_name	course_name
1	1	Mohammad Khan	Course 1
2	2	Fatima Ali	Course 2
3	3	Ahmed Malik	Course 3
4	4	Ayesha Ahmed	Course 4
5	5	Ali Khan	Course 5
6	6	Sara Malik	Course 6
7	7	Usman Ahmed	Course 7
8	8	Farah Khan	Course 8
9	9	Abdullah Malik	Course 9
10	10	Sadia Ahmed	Course 10

```
--Retrieve all courses and their corresponding instructor names,
including courses without assigned instructors:


SELECT C.course_id, C.course_name, I.instructor_name
FROM Courses C
LEFT JOIN Instructors I ON C.instructor_id = I.instructor_id;
```


Results | Messages

	course_id	course_name	instructor_name
1	1	Course 1	Ali Khan
2	2	Course 2	Ali Khan
3	3	Course 3	Mohammad Khan
4	4	Course 4	Mohammad Khan
5	5	Course 5	Ahmed Ali
6	6	Course 6	Ahmed Ali
7	7	Course 7	Ayesha Khan
8	8	Course 8	Usman Ahmed
9	9	Course 9	Usman Ahmed
10	10	Course 10	Sadia Ahmed
11	11	Course 11	Fatima Ahmed
12	12	Course 12	Sara Malik
13	13	Course 13	Farah Hussain
14	14	Course 14	Abdullah Malik

--i CREATING UPDATABLE VIEWS

```
CREATE VIEW UpdatableCoursesView AS
SELECT course_id, course_name
FROM Courses;
```

--j MODIFYING VIEWS

```
ALTER VIEW UpdatableCoursesView AS
SELECT course_id, course_name, instructor_id, start_date, end_date
FROM Courses;
```

SQL Server does not support directly modifying the structure of a view using ALTER VIEW instead, we need to drop and recreate the view with the desired changes.

================================== **END ☺** ===================================