

Assignment on Database (MySQL)

1. Write a query to find the students' first name, last name, phone number, and address for those who have the blood groups O+ and A-

```
SELECT firstname, lastname, fulladdress
FROM studentpersonal
WHERE bloodgroup IN ('O+', 'A-');
```

The screenshot shows a MySQL IDE interface. The top pane displays the following SQL query:

```
SELECT firstname, lastname, fulladdress
FROM studentpersonal
WHERE bloodgroup IN ('O+', 'A-');
```

The bottom pane shows the results of the query in a table format. The table has three columns: **firstname**, **lastname**, and **fulladdress**. There are three rows of data:

	firstname	lastname	fulladdress
1	John	Doe	123 5th Avenue, Manhattan, New York, NY 10001
2	Emily	Smith	456 Sunset Boulevard, Downtown, Los Angeles, CA 90001
3	Yuki	Tanaka	1212 Jackson St, Japantown, San Jose, CA 95112

The status bar at the bottom indicates that 3 row(s) were fetched in 0.005s (0.001s fetch) on 2025-07-23 at 11:06:10. The system tray at the bottom right shows the time as 11:06:15 AM on 23/07/2025.

2. Write a query to find the department name, department code, and subject title for the subject with the subject code "EE201."

```
SELECT d.departmentName , d.departmentCode, s.subjectTitle
FROM subjects s
JOIN departments d on d.departmentCode = s.departmentCode
WHERE s.subjectCode = 'EE201'
```

The screenshot shows a database management tool interface. The top pane displays a SQL query: `SELECT d.departmentName , d.departmentCode, s.subjectTitle FROM subjects s JOIN departments d on d.departmentCode = s.departmentCode WHERE s.subjectCode = 'EE201'`. The bottom pane shows the results of the query in a table grid. The table has three columns: departmentName, departmentCode, and subjectTitle. The first row shows 'Electrical Engineering', 'EE102', and 'Circuit Analysis'. The status bar at the bottom indicates '1 row(s) fetched - 0.003s, on 2025-07-23 at 11:08:52'.

	departmentName	departmentCode	subjectTitle
1	Electrical Engineering	EE102	Circuit Analysis

3. Show the count of students for each blood group that has at least 2 students

```
SELECT s.bloodgroup, COUNT(*)  
FROM studentpersonal s  
GROUP BY s.bloodgroup  
HAVING COUNT(*) >= 2;
```

The screenshot shows a database management tool interface. The top pane displays the following SQL query:

```
SELECT s.bloodgroup, COUNT(*)  
FROM studentpersonal s  
GROUP BY s.bloodgroup  
HAVING COUNT(*) >= 2;
```

The bottom pane shows the results of the query in a table format. The table has two columns: 'bloodgroup' and 'COUNT(*)'. There are three rows of data:

	bloodgroup	COUNT(*)
1	A-	2
2	AB+	2
3	O-	2

The status bar at the bottom indicates that 3 rows were fetched in 0.006s (0.001s fetch) on 2025-07-23 at 11:09:51. The interface also includes a toolbar with various icons for navigation and data manipulation, and a sidebar with a 'Panels' section.

4. Write a query to find the students' first name, last name, and subject title for those enrolled in the department CS101.

```
SELECT sp.firstname, sp.lastname, sub.subjectTitle
FROM studentpersonal sp
JOIN courses c ON sp.studentId = c.studentId
JOIN subjects sub ON c.subjectCode = sub.subjectCode
WHERE sub.departmentCode = 'CS101';
```

The screenshot shows a database management tool interface. The top panel displays a SQL query: `SELECT sp.firstname, sp.lastname, sub.subjectTitle FROM studentpersonal sp JOIN courses c ON sp.studentId = c.studentId JOIN subjects sub ON c.subjectCode = sub.subjectCode WHERE sub.departmentCode = 'CS101';`. The bottom panel shows the results of the query in a grid view. The grid has three columns: 'A-Z firstname', 'A-Z lastname', and 'A-Z subjectTitle'. There are three rows of data: John Doe (Introduction to Programming), John Doe (Data Structures), and Emily Smith (Algorithms). A tooltip is visible over the first row, stating: 'Column: firstname varchar(50) Read-only: No valid row identifier found'. The status bar at the bottom indicates '3 row(s) fetched - 0.002s, on 2025-07-23 at 11:03:15'.

	A-Z firstname	A-Z lastname	A-Z subjectTitle
1	John	Doe	Introduction to Programming
2	John	Doe	Data Structures
3	Emily	Smith	Algorithms

5. Find the total semester fee collected for the Computer Science department.

```
SELECT SUM(totalSemesterFees), COUNT(*)  
FROM studentacademic sa  
JOIN departments d ON sa.departmentCode = d.departmentCode  
WHERE d.departmentName = 'Computer Science'
```

The screenshot shows a database management tool interface. The top pane displays a SQL query: `SELECT SUM(totalSemesterFees), COUNT(*) FROM studentacademic sa JOIN departments d ON sa.departmentCode = d.departmentCode WHERE d.departmentName = 'Computer Science'`. The bottom pane shows the results of the query in a grid format. The first row contains the values 4,500 and 3, corresponding to the SUM and COUNT functions respectively. A tooltip is visible over the first row, showing the value 4500.00. The interface includes a toolbar with various icons for editing and viewing data, and a status bar at the bottom showing the current database (BDT), language (en), and other settings.

	123 SUM(totalSemesterFees)	123 COUNT(*)
1	4,500	3

6. Write a query to update the semester fee by 10% for students in the CS101 department.

```
SELECT
    studentId, totalSemesterFees,
    totalSemesterFees * 1.10 AS updated_fee
FROM studentacademic
WHERE departmentCode = 'CS101';
```

The screenshot shows a database management tool interface. The top pane displays a SQL query: `SELECT studentId, totalSemesterFees, totalSemesterFees * 1.10 AS updated_fee FROM studentacademic WHERE departmentCode = 'CS101';`. The bottom pane shows the results of this query in a grid view. The grid has four columns: `studentId`, `totalSemesterFees`, and `updated_fee`. The first three rows of data are visible, showing student IDs 1, 5, and 6, all with a `totalSemesterFees` of 1,500 and an `updated_fee` of 1,650. To the right of the grid, there is a 'Dictionary' panel for the `studentpersonal` table, showing a list of student names and their descriptions. The status bar at the bottom indicates that 3 rows were fetched.

studentId	totalSemesterFees	updated_fee
1	1,500	1,650
5	1,500	1,650
6	1,500	1,650

3 row(s) fetched - 0.082s (0.001s fetch), on 2025-07-23 at 14:33:17

7. Write a query to find the students' first name, last name, teacher's first name, and teacher's designation who are enrolled in the course CS101.

```
SELECT
    sp.firstname AS student_firstname,
    sp.lastname AS student_lastname,
    tp.firstname AS teacher_firstname,
    ta.designation AS teacher_designation
FROM studentacademic sa
JOIN studentpersonal sp ON sa.studentId = sp.studentId
JOIN teacheracademic ta ON sa.departmentCode = ta.departmentCode
JOIN teacherpersonal tp ON ta.teacherId = tp.teacherId
WHERE sa.departmentCode = 'CS101';
```

The screenshot shows a database management tool interface. The top pane displays a SQL query in a script editor. The query is a SELECT statement that joins four tables: studentacademic (sa), studentpersonal (sp), teacheracademic (ta), and teacherpersonal (tp). The query filters for students enrolled in the course CS101. The bottom pane shows the results of the query in a grid view. The grid has four columns: student_firstname, student_lastname, teacher_firstname, and teacher_designation. There are three rows of data, all showing 'Sr. Lecturer' as the teacher designation.

	AZ student_firstname	AZ student_lastname	AZ teacher_firstname	AZ teacher_designation
1	John	Doe	David	Sr. Lecturer
2	Chen	Lee	David	Sr. Lecturer
3	Alice	Green	David	Sr. Lecturer

8. Write a query to show the student's first name and the teacher's last name where both the teacher and student are from the same city.

SELECT

sp.firstname **AS** *student_firstname*,
tp.lastname **AS** *teacher_lastname*, *sp*.city

FROM studentpersonal *sp*

JOIN teacherpersonal *tp* **ON** *sp*.city = *tp*.city;

The screenshot shows a database management tool interface. The top panel displays a SQL query in a script editor. The query is as follows:

```
SELECT
    sp.firstname AS student_firstname,
    tp.lastname AS teacher_lastname,
    sp.city
FROM studentpersonal sp
JOIN teacherpersonal tp ON sp.city = tp.city;
```

The bottom panel shows the results of the query in a table view. The table has three columns: *student_firstname*, *teacher_lastname*, and *city*. There are 7 rows of data.

	<i>student_firstname</i>	<i>teacher_lastname</i>	<i>city</i>
1	John	Johnson	New York
2	Emily	Brown	Los Angeles
3	Mohammed	Ali	Houston
4	Priya	Mehta	Chicago
5	Chen	Zhang	San Francisco
6	Ibrahim	Johnson	New York
7	Yuki	Mehta	Chicago

The interface also includes a toolbar with various icons for navigation and execution, and a status bar at the bottom showing the number of rows fetched (7 rows) and the execution time (0.001s).

9. Write a query to show the student's first name, last name, email, subject code, and subject title for the students enrolled in the subject "Algorithms."

```
SELECT sp.firstname, sp.lastname, c.subjectCode, s.subjectTitle
FROM courses c
JOIN studentpersonal sp ON c.studentId = sp.studentId
JOIN subjects s ON c.subjectCode = s.subjectCode
WHERE s.subjectTitle = 'Algorithms';
```

The screenshot shows a database management tool interface. The top pane displays a SQL query in a script editor. The query is as follows:

```
SELECT sp.firstname, sp.lastname, c.subjectCode, s.subjectTitle
FROM courses c
JOIN studentpersonal sp ON c.studentId = sp.studentId
JOIN subjects s ON c.subjectCode = s.subjectCode
WHERE s.subjectTitle = 'Algorithms';
```

The bottom pane shows the results of the query in a grid view. The grid has four columns: 'AZ firstname', 'AZ lastname', 'AZ subjectCode', and 'AZ subjectTitle'. There is one row of data:

	AZ firstname	AZ lastname	AZ subjectCode	AZ subjectTitle
1	Emily	Smith	CS203	Algorithms

The status bar at the bottom indicates '1 row(s) fetched - 0.003s, on 2025-07-24 at 11:32:28'. A red circle with the number 4 is visible in the bottom right corner of the grid area.

10. Write a query to promote teachers who are currently Lecturers to Sr.Lecturers, and promote Sr. Lecturers to Associate Professors.

```
SELECT ta.teacherId, tp.firstname, ta.Designation,
CASE
    WHEN Designation = 'Lecturer' THEN 'Sr. Lecturer'
    WHEN Designation = 'Sr. Lecturer' THEN 'Associate Professor'
    ELSE Designation
END AS promoted_to
FROM teacheracademic ta
JOIN teacherpersonal tp ON ta.teacherId = tp.teacherId;
```

The screenshot shows a database management tool interface. The top pane displays a SQL query that promotes teachers based on their current designation. The bottom pane shows the results of the query in a table format.

SQL Query:

```
SELECT ta.teacherId, tp.firstname, ta.Designation,
CASE
    WHEN Designation = 'Lecturer' THEN 'Sr. Lecturer'
    WHEN Designation = 'Sr. Lecturer' THEN 'Associate Professor'
    ELSE Designation
END AS promoted_to
FROM teacheracademic ta
JOIN teacherpersonal tp ON ta.teacherId = tp.teacherId;
```

Query Results:

teacherId	firstname	Designation	promoted_to
1	David	Sr. Lecturer	Associate Professor
2	Sarah	Sr. Lecturer	Associate Professor
3	Ahmed	Lecturer	Sr. Lecturer
4	Sanjay	Lecturer	Sr. Lecturer
5	Wei	Associate Professor	Associate Professor

The interface also shows a status bar at the bottom indicating "5 row(s) fetched - 0.002s, on 2025-07-24 at 15:42:36".

11. Write a query to find the teacher's first name and last name who has the second highest salary.

```
SELECT tp.firstname, tp.lastname
FROM teacherpersonal tp
JOIN teacheracademic ta ON tp.teacherId = ta.teacherId
WHERE ta.salary = (
    SELECT MAX(salary)
    FROM teacheracademic
    WHERE salary < (
        SELECT MAX(salary)
        FROM teacheracademic
    )
);
```

The screenshot shows a database management tool interface. The top pane displays a SQL query in a script editor. The query is designed to find the teacher's first name and last name who has the second highest salary. The bottom pane shows the results of the query in a grid view. The results table has two columns: 'AZ firstname' and 'AZ lastname'. The first row of data shows 'David' and 'Johnson'. The status bar at the bottom indicates that 1 row(s) were fetched.

```
SELECT tp.firstname, tp.lastname
FROM teacherpersonal tp
JOIN teacheracademic ta ON tp.teacherId = ta.teacherId
WHERE ta.salary = (
    SELECT MAX(salary)
    FROM teacheracademic
    WHERE salary < (
        SELECT MAX(salary)
        FROM teacheracademic
    )
);
```

	AZ firstname	AZ lastname
1	David	Johnson

1 row(s) fetched - 0.012s, on 2025-07-24 at 15:54:55

-);

The screenshot displays the Microsoft Access interface with a SQL query in the SQL View and its results in the Datasheet View.

SQL View:

```

SELECT
    tp.firstname AS teacher_firstname,
    tp.lastname AS teacher_lastname,
    sp.firstname AS student_firstname,
    sp.lastname AS student_lastname,
    sp.city,
    sa.departmentCode
FROM teacheracademic ta
JOIN teacherpersonal tp ON ta.teacherId = tp.teacherId
JOIN studentacademic sa ON ta.departmentCode = sa.departmentCode
JOIN studentpersonal sp ON sa.studentId = sp.studentId
WHERE ta.salary = (
    SELECT MAX(salary)
    FROM teacheracademic
    WHERE salary < (
        SELECT MAX(salary)
        FROM teacheracademic
    )
);

```

Datasheet View:

The query results are displayed in a table with the following columns: teacher_firstname, teacher_lastname, student_firstname, student_lastname, city, and departmentCode. The results show three records for teachers named David Johnson.

	teacher_firstname	teacher_lastname	student_firstname	student_lastname	city	departmentCode
1	David	Johnson	John	Doe	New York	CS101
2	David	Johnson	Chen	Lee	San Francisco	CS101
3	David	Johnson	Alice	Green	Boston	CS101

05:01:06 PM
24/07/2025