

ASSIGNMENT STUDENT INFORMATION SYSTEM (SIS)

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write an SQL query to calculate the total payments made by a specific student. You will need to join the "Payments" table with the "Students" table based on the student's ID.

use sisdb;

```
SELECT s.student_id, s.first_name, s.last_name, SUM(p.amount) AS total_payment
FROM Students s
JOIN Payments p ON s.student_id = p.student_id
WHERE s.student_id = 5
GROUP BY s.student_id, s.first_name, s.last_name;
```

The screenshot shows a SQL query editor with the following query:

```
2
3 • SELECT s.student_id, s.first_name, s.last_name, SUM(p.amount) AS total_payment
4 FROM Students s
5 JOIN Payments p ON s.student_id = p.student_id
6 WHERE s.student_id = 5
7 GROUP BY s.student_id, s.first_name, s.last_name;
8
```

Below the query editor is the "Result Grid" showing the results of the query:

student_id	first_name	last_name	total_payment
5	AMAL	Davis	4600.00

The result grid also includes a "Filter Rows" section and an "Export" button. The "Read Only" status is indicated at the bottom right.

2. Write an SQL query to retrieve a list of courses along with the count of students enrolled in each course. Use a JOIN operation between the "Courses" table and the "Enrollments" table.

```
SELECT c.course_id, c.course_name, COUNT(e.student_id) AS student_count
FROM Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name
ORDER BY student_count DESC;
```

The screenshot shows a SQL query editor with the following query:

```
8
9 • SELECT c.course_id, c.course_name, COUNT(e.student_id) AS student_count
10 FROM Courses c
11 LEFT JOIN Enrollments e ON c.course_id = e.course_id
12 GROUP BY c.course_id, c.course_name
13 ORDER BY student_count DESC;
14
```

Below the query editor is the "Result Grid" showing the results of the query:

course_id	course_name	student_count
2	python	1
3	C#	1
4	C	1
5	C++	1
6	JAVASCRIPT	1
7	SQL	1
8	HTML	1
9	CSS	1
10	DATA SCIENCE	1
1	JAVA	0

The result grid also includes a "Filter Rows" section and an "Export" button. The "Read Only" status is indicated at the bottom right.

3. Write an SQL query to find the names of students who have not enrolled in any course. Use a LEFT JOIN between the "Students" table and the "Enrollments" table to identify students without enrollments.

```
SELECT s.student_id, s.first_name, s.last_name
FROM Students s
LEFT JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.course_id IS NULL;
```

The screenshot shows a SQL query editor with the following query:

```
15 • SELECT s.student_id, s.first_name, s.last_name
16 FROM Students s
17 LEFT JOIN Enrollments e ON s.student_id = e.student_id
18 WHERE e.course_id IS NULL;
19
```

Below the query editor is the "Result Grid" showing the results of the query:

student_id	first_name	last_name
11	JOHN	DOE

The interface includes a "Filter Rows" field, an "Export" button, and a "Wrap Cell Content" button. On the right side, there are buttons for "Result Grid", "Form Editor", and "Field Types". At the bottom, there is a "Read Only" indicator.

4. Write an SQL query to retrieve the first name, last name of students, and the names of the courses they are enrolled in. Use JOIN operations between the "Students" table and the "Enrollments" and "Courses" tables.

```
SELECT s.student_id, s.first_name, s.last_name, c.course_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
JOIN Courses c ON e.course_id = c.course_id
ORDER BY s.student_id;
```

The screenshot shows a SQL query editor with the following query:

```
20 • SELECT s.student_id, s.first_name, s.last_name, c.course_name
21 FROM Students s
22 JOIN Enrollments e ON s.student_id = e.student_id
23 JOIN Courses c ON e.course_id = c.course_id
24 ORDER BY s.student_id;
25
```

Below the query editor is the "Result Grid" showing the results of the query:

student_id	first_name	last_name	course_name
2	BOB	SMITH	python
3	RAJ	VEL	C#
4	DAVID	RAV	C
5	AMAL	Davis	C++
6	KISHORE	ROY	JAVASCRIPT
7	JASMINE	RAV	SQL
8	STELLA	MERRY	HTML
9	Isaac	newton	CSS
10	RANI	RAM	DATA SCIENCE

The interface includes a "Filter Rows" field, an "Export" button, and a "Wrap Cell Content" button. On the right side, there are buttons for "Result Grid", "Form Editor", and "Field Types". At the bottom, there is a "Read Only" indicator.

5. Create a query to list the names of teachers and the courses they are assigned to. Join the "Teacher" table with the "Courses" table.

```
SELECT t.teacher_id, t.first_name, t.last_name, c.course_name
FROM Teachers t
LEFT JOIN Courses c ON t.teacher_id = c.teacher_id
ORDER BY t.teacher_id;
```

The screenshot shows a SQL query editor with the following query:

```
26 • SELECT t.teacher_id, t.first_name, t.last_name, c.course_name
27 FROM Teachers t
28 LEFT JOIN Courses c ON t.teacher_id = c.teacher_id
29 ORDER BY t.teacher_id;
30
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has four columns: teacher_id, first_name, last_name, and course_name. The results are as follows:

teacher_id	first_name	last_name	course_name
1	SONA	KUMARI	JAVA
2	MERCY	PRINCE	python
3	MARTIN	RAJ	NULL
4	RANJITH	KUMAR	C
5	SANJAY	RAM	C#
5	SANJAY	RAM	C++
6	DHIVYA	PRIYA	JAVASCRIPT
7	AKALYA	MURUGESH	SQL
8	VARSHINI	MARTIN	HTML
9	KOWSI	LAKSHMI	CSS
10	PRIYA	RAV	DATA SCIENCE

The interface includes a "Filter Rows" field, an "Export" button, and a "Wrap Cell Content" button. The status bar at the bottom indicates "Result 5" and "Read Only".

6. Retrieve a list of students and their enrollment dates for a specific course. You'll need to join the "Students" table with the "Enrollments" and "Courses" tables.

```
SELECT s.student_id, s.first_name, s.last_name, e.enrollment_date
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
JOIN Courses c ON e.course_id = c.course_id
WHERE c.course_id = 3;
```

The screenshot shows a SQL query editor with the following query:

```
31 • SELECT s.student_id, s.first_name, s.last_name, e.enrollment_date
32 FROM Students s
33 JOIN Enrollments e ON s.student_id = e.student_id
34 JOIN Courses c ON e.course_id = c.course_id
35 WHERE c.course_id = 3;
36
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has four columns: student_id, first_name, last_name, and enrollment_date. The results are as follows:

student_id	first_name	last_name	enrollment_date
3	RAJ	VEL	2024-01-12

The interface includes a "Filter Rows" field, an "Export" button, and a "Wrap Cell Content" button. The status bar at the bottom indicates "Result 6" and "Read Only".

7. Find the names of students who have not made any payments. Use a LEFT JOIN between the "Students" table and the "Payments" table and filter for students with NULL payment records.

```
SELECT s.student_id, s.first_name, s.last_name
FROM Students s
LEFT JOIN Payments p ON s.student_id = p.student_id
WHERE p.payment_id IS NULL;
```

The screenshot shows a SQL IDE interface. The query editor at the top contains the following SQL code:

```
37 • SELECT s.student_id, s.first_name, s.last_name
38 FROM Students s
39 LEFT JOIN Payments p ON s.student_id = p.student_id
40 WHERE p.payment_id IS NULL;
41
```

Below the query editor is a toolbar with options: "Result Grid" (selected), "Filter Rows:", "Export:", and "Wrap Cell Content:". Below the toolbar is a table with the following data:

	student_id	first_name	last_name
▶	11	JOHN	DOE

On the right side of the IDE, there is a vertical toolbar with icons for "Result Grid", "Form Editor", and "Field Types". At the bottom of the IDE, there is a status bar that says "Result 7 x" and "Read Only".

8. Write a query to identify courses that have no enrollments. You'll need to use a LEFT JOIN between the "Courses" table and the "Enrollments" table and filter for courses with NULL enrollment records.

```
SELECT c.course_id, c.course_name
FROM Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
WHERE e.enrollment_id IS NULL;
```

The screenshot shows a SQL IDE interface. The query editor at the top contains the following SQL code:

```
42 • SELECT c.course_id, c.course_name
43 FROM Courses c
44 LEFT JOIN Enrollments e ON c.course_id = e.course_id
45 WHERE e.enrollment_id IS NULL;
46
```

Below the query editor is a toolbar with options: "Result Grid" (selected), "Filter Rows:", "Export:", and "Wrap Cell Content:". Below the toolbar is a table with the following data:

	course_id	course_name
▶	1	JAVA

On the right side of the IDE, there is a vertical toolbar with icons for "Result Grid", "Form Editor", and "Field Types". At the bottom of the IDE, there is a status bar that says "Result 8 x" and "Read Only".

9. Identify students who are enrolled in more than one course. Use a self-join on the "Enrollments" table to find students with multiple enrollment records.

```
SELECT DISTINCT e1.student_id, s.first_name, s.last_name
FROM Enrollments e1
JOIN Enrollments e2 ON e1.student_id = e2.student_id AND e1.course_id != e2.course_id
JOIN Students s ON e1.student_id = s.student_id;
```

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

```
328 • SELECT DISTINCT e1.student_id, s.first_name, s.last_name
329 FROM Enrollments e1
330 JOIN Enrollments e2 ON e1.student_id = e2.student_id AND e1.course_id != e2.course_id
331 JOIN Students s ON e1.student_id = s.student_id;
332
```

The bottom pane shows the 'Result Grid' with the following columns: student_id, first_name, last_name. The grid is currently empty. The interface includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox. On the right side, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'. The bottom status bar indicates 'Result 1 x' and 'Read Only'.

10. Find teachers who are not assigned to any courses. Use a LEFT JOIN between the "Teacher" table and the "Courses" table and filter for teachers with NULL course assignments.

```
SELECT t.teacher_id, t.first_name, t.last_name
FROM Teachers t
LEFT JOIN Courses c ON t.teacher_id = c.teacher_id
WHERE c.course_id IS NULL;
```

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

```
53 • SELECT t.teacher_id, t.first_name, t.last_name
54 FROM Teachers t
55 LEFT JOIN Courses c ON t.teacher_id = c.teacher_id
56 WHERE c.course_id IS NULL;
57
```

The bottom pane shows the 'Result Grid' with the following columns: teacher_id, first_name, last_name. The grid contains one row of data:

teacher_id	first_name	last_name
3	MARTIN	RAJ

The interface includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox. On the right side, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'. The bottom status bar indicates 'Result 10 x' and 'Read Only'. Below the status bar is an 'Output' pane.