TASK 3 – Aggregate functions, Having, Order By, Group By 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.

```
SELECT SUM(AMOUNT)
FROM payments p
JOIN students s
ON p.student_id = s.student_id
WHERE s.student_id=108;
```

Result Grid | Filter Rows:

SUM(AMOUNT)
67000

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)
FROM courses c
LEFT JOIN enrollments e
ON c.course_id = e.course_id
GROUP BY c.course_id;
```

Result Grid | Filter Rows: | Export:

course_id	course_name	COUNT(e.student_id)
301	AERONAUTICAL	2
302	ECE	2
303	EEE	1
304	EIE	4
305	CES	2
306	CSBS	3
307	IT	3
308	MECHANICAL	2
309	AGRICULTURE	2
310	TEXTILE	2

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.

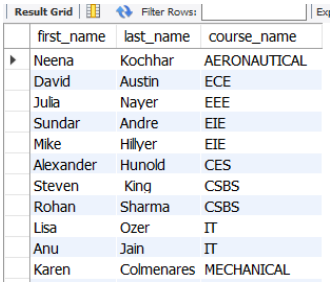
```
SELECT CONCAT(s.first_name," ",s.last_name) AS student_name
FROM students s
LEFT JOIN enrollments e
ON e.student_id = s.student_id
WHERE e.student_id IS NULL;
```

Result Grid | Filter Rows:

student_name
Adam Smith
Jhon Doe

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.

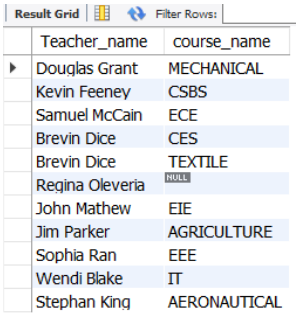
```
SELECT s.first_name, s.last_name,c.course_name
FROM students s
JOIN enrollments e
ON s.student_id = e. student_id
JOIN courses c
ON e.course_id=c.course_id;
```



first_name	last_name	course_name
Neena	Kochhar	AERONAUTICAL
David	Austin	ECE
Julia	Nayer	EEE
Sundar	Andre	EIE
Mike	Hillyer	EIE
Alexander	Hunold	CES
Steven	King	CSBS
Rohan	Sharma	CSBS
Lisa	Ozer	IT
Anu	Jain	IT
Karen	Colmenares	MECHANICAL

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.

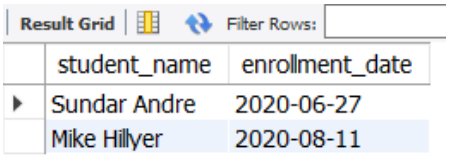
```
SELECT CONCAT (t.first_name," ", t.last_name) AS Teacher_name,
    c.course_name
FROM teacher t
LEFT JOIN courses c
ON c.teacher_id=t.teacher_id;
```



Teacher_name	course_name
Douglas Grant	MECHANICAL
Kevin Feeney	CSBS
Samuel McCain	ECE
Brevin Dice	CES
Brevin Dice	TEXTILE
Regina Oleveria	NULL
John Mathew	EIE
Jim Parker	AGRICULTURE
Sophia Ran	EEE
Wendi Blake	IT
Stephan King	AERONAUTICAL

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 CONCAT(s.first_name," ",s.last_name) AS student_name,
    e.enrollment_date
FROM students s
JOIN enrollments e
ON s.student_id=e.student_id
JOIN courses c
ON c.course_id=e.course_id
WHERE c.course_id=304;
```

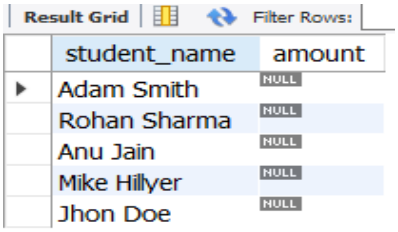


student_name	enrollment_date
Sundar Andre	2020-06-27
Mike Hillyer	2020-08-11

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 CONCAT(s.first_name," ",s.last_name) AS student_name,
```

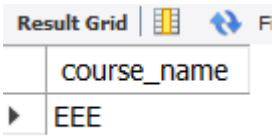
```
    p.amount
FROM students s
LEFT JOIN payments p
ON s.student_id=p.student_id
WHERE p.payment_id IS NULL;
```



student_name	amount
Adam Smith	NULL
Rohan Sharma	NULL
Anu Jain	NULL
Mike Hillyer	NULL
Jhon Doe	NULL

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.

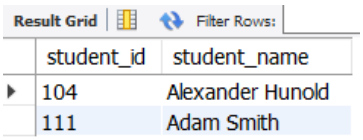
```
SELECT c.course_name
FROM courses c
LEFT JOIN enrollments e
ON c.course_id=e.course_id
WHERE e.course_id IS NULL;
```



course_name
EEE

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.

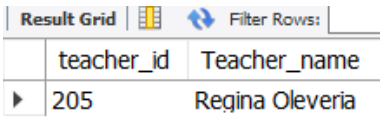
```
SELECT DISTINCT e1.student_id,
        CONCAT (s.first_name," ",s.last_name)AS student_name
FROM enrollments e1
JOIN enrollments e2
ON e1.student_id = e2.student_id AND e1.enrollment_id <> e2.enrollment_id
JOIN Students s
ON e1.student_id = s.student_id
ORDER BY
  e1.student_id;
```



student_id	student_name
104	Alexander Hunold
111	Adam Smith

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 DISTINCT (t.teacher_id),
        CONCAT (t.first_name," ", t.last_name) AS Teacher_name
FROM teacher t
LEFT JOIN courses c
ON t.teacher_id = c.teacher_id
WHERE c.teacher_id IS NULL;
```



teacher_id	Teacher_name
205	Regina Oleveria