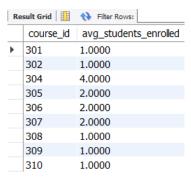
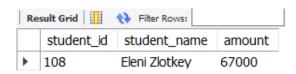
TASK 4 - Subquery and its type:-

1. Write an SQL query to calculate the average number of students enrolled in each course. Use aggregate functions and subqueries to achieve this.

```
SELECT c.course_id, AVG(student_count) AS avg_students_enrolled FROM courses c
INNER JOIN (
    SELECT course_id, COUNT(DISTINCT student_id) AS student_count FROM enrollments
    GROUP BY course_id
) AS enrolled_students
ON c.course_id = enrolled_students.course_id
GROUP BY c.course_id;
```



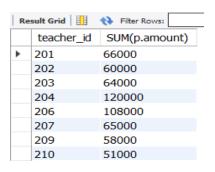
2. Identify the student(s) who made the highest payment. Use a subquery to find the maximum payment amount and then retrieve the student(s) associated with that amount.



3. Retrieve a list of courses with the highest number of enrollments. Use subqueries to find the course(s) with the maximum enrollment count.

```
SELECT c.course_id,
   c.course_name,
   COUNT(*) AS enrollment_count
FROM courses c
JOIN enrollments e
ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name
HAVING
 COUNT(*) = (
       SELECT MAX(enrollment_count)
       FROM
        (SELECT COUNT(*) AS enrollment_count
        FROM enrollments
        GROUP BY course_id) AS max_enrollment
       );
                 Result Grid | 🚻 ( Filter Rows:
                    course_id | course_name | enrollment_count
                 ▶ 304
                           EIE
```

4. Calculate the total payments made to courses taught by each teacher. Use subqueries to sum payments for each teacher's courses.



5. Identify students who are enrolled in all available courses. Use subqueries to compare a student's enrollments with the total number of courses.

6. Retrieve the names of teachers who have not been assigned to any courses. Use subqueries to find teachers with no course assignments.

```
SELECT teacher_id,

CONCAT(first_name, " ", last_name) AS Teacher_name
FROM teacher

WHERE teacher_id NOT IN (

SELECT c.teacher_id

FROM courses c
);

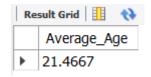
Result Grid | HE  Filter Rows: ______

teacher_id  Teacher_name

> 205  Regina Oleveria
```

7. Calculate the average age of all students. Use subqueries to calculate the age of each student based on their date of birth.

SELECT AVG(ages) AS Average_Age
FROM (SELECT TIMESTAMPDIFF(YEAR,date_of_birth,CURDATE()) AS ages
FROM students)AS student_ages;

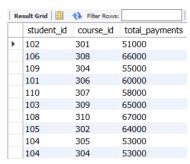


8. Identify courses with no enrollments. Use subqueries to find courses without enrollment records.

9. Calculate the total payments made by each student for each course they are enrolled in. Use subqueries and aggregate functions to sum payments.

```
SELECT
e.student_id,
e.course_id,
SUM(p.amount) AS total_payments
FROM Enrollments e
JOIN Payments p ON e.student_id = p.student_id
GROUP BY e.student_id, e.course_id;
```

WHERE payment_count >1;



10. Identify students who have made more than one payment. Use subqueries and aggregate functions to count payments per student and filter for those with counts greater than one.

```
SELECT s.student_id,
    CONCAT(s.first_name, " ", s.last_name) AS Student_Name

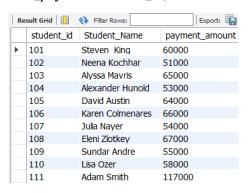
FROM students s

JOIN (SELECT
    p.student_id,
    COUNT(p.payment_id) AS payment_count
    FROM payments p
    GROUP BY p.student_id) AS Student_payment

ON s.student_id=Student_payment.student_id
```

11. Write an SQL query to calculate the total payments made by each student. Join the "Students" table with the "Payments" table and use GROUP BY to calculate the sum of payments for each student.

```
SELECT s.student_id,
    CONCAT(s.first_name, " ", s.last_name) AS Student_Name,
    payment_amount
FROM students s
JOIN (SELECT
    p.student_id,
    SUM(p.amount) AS payment_amount
    FROM payments p
    GROUP BY p.student_id) AS Student_payment
ON s.student_id=Student_payment.student_id;
```



12. Retrieve a list of course names along with the count of students enrolled in each course. Use JOIN operations between the "Courses" table and the "Enrollments" table and GROUP BY to count enrollments.

Result Grid		Name of the Filter Rows:	Export
	course_id	course_name	Student_Count
•	301	AERONAUTICAL	2
	302	ECE	2
	303	EEE	1
	304	EIE	4
	305	CES	2
	306	CSBS	3
	307	Π	3
	308	MECHANICAL	2
	309	AGRICULTURE	2
	310	TEXTILE	2

13. Calculate the average payment amount made by students. Use JOIN operations between the "Students" table and the "Payments" table and GROUP BY to calculate the average.

