#### Name - Ishaan Gupta

Batch - c# batch 2

<u>Assignment - Student Information System (SIS)</u> <u>Task - 4</u>

(question+SQL query+output added below)

# Q1) Write an SQL query to calculate the average number of students enrolled in each course. Use aggregate functions and subqueries to achieve this

```
task4.sql - DESKTOP...ESS.SISDB (sa (63))* 😕 🗶
   □ SELECT AVG(student_count) AS avg_enrollments
     FROM (
         SELECT COUNT(e.enrollment_id) AS student_count
         FROM Courses c
         LEFT JOIN Enrollments e ON c.course_id = e.course_i
         GROUP BY c.course_id
     ) AS subquery;
100 % ▼ 4
■ Results  Messages
     avg_enrollments
```

0

1

# Q2) 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

### Q3) 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_name

     FROM Courses c
     JOIN Enrollments e ON c.course id = e.course id
     GROUP BY c.course_name
     HAVING COUNT(e.enrollment_id) = (
          SELECT MAX(enrollment count)
          FROM (
              SELECT COUNT(enrollment_id) AS enrollment_count
              FROM Enrollments
              GROUP BY course_id
          ) AS subquery
     );
100 %
■ Results Messages
     course_name
     Biology
1
     Chemistry
2
3
     Computer Science
     Economics
4
5
     English
     History
6
     Mathematics
7
     Political Science
     Statistics
```

### Q4) Calculate the total payments made to courses taught by each teacher. Use subqueries to sum payments for each teacher's courses

```
FROM Teacher t

JOIN Courses c ON t.teacher_id = c.teacher_id

JOIN Enrollments e ON c.course_id = e.course_id

JOIN Payments p ON e.student_id = p.student_id

GROUP BY t.first_name, t.last_name;
```

100 % ▼ ◀ ■ Results ■ Messages

	first_name	last_name	total_payments
1	Anjali	Nair	2200.00
2	Karan	Shah	1100.00
3	Megha	Ghosh	1300.00
4	Seema	Joshi	3000.00
5	Suresh	Yadav	1400.00
6	Varsha	Patil	3500.00
7	Vijay	Thakur	1600.00

### Q5) Identify students who are enrolled in all available courses. Use subqueries to compare a student's enrollments with the total number of courses

### Q6) Retrieve the names of teachers who have not been assigned to any courses. Use subqueries to find teachers with no course assignments.

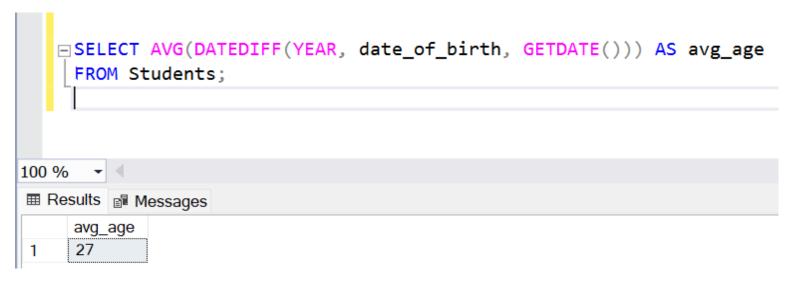
```
FROM Teacher t
WHERE NOT EXISTS (
SELECT c.course_id
FROM Courses c
WHERE c.teacher_id = t.teacher_id
);

100 %

Results Messages

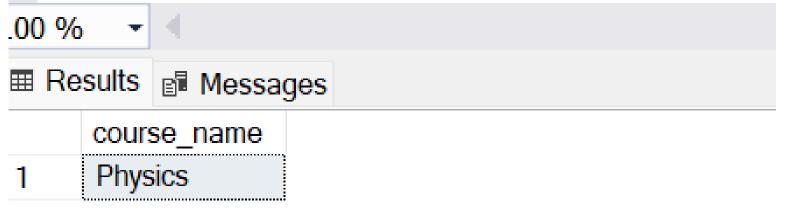
first_name last_name
1 Preeti Reddy
2 Nisha Chauhan
```

### Q7) Calculate the average age of all students. Use subqueries to calculate the age of each student based on their date of birth.



### Q8) Identify courses with no enrollments. Use subqueries to find courses without enrollment records.

```
FROM Courses c
WHERE NOT EXISTS (
SELECT e.course_id
FROM Enrollments e
WHERE e.course_id = c.course_id
);
```



### Q9) Calculate the total payments made by each student for each course they are enrolled in. Use subqueries and aggregate functions to sum payments.

```
SELECT s.first_name, s.last_name, c.course_name, SUM(p.amount) AS total_payment
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
JOIN Courses c ON e.course_id = c.course_id
JOIN Payments p ON s.student_id = p.student_id
GROUP BY s.first_name, s.last_name, c.course_name;

### Results ### Messages
| first_name | last_name | course_name | total_payment |
| Anita | Nair | Political Science | 1600.00 |
| Ishaan | Gupta | Mathematics | 1000.00 |
| John | Doe | Biology | 2200.00
```

	first_name	last_name	course_name	total_payment
1	Anita	Nair	Political Science	1600.00
2	Ishaan	Gupta	Mathematics	1000.00
3	John	Doe	Biology	2200.00
4	Manish	Rao	Statistics	1400.00
5	Rajesh	Patel	History	1200.00
6	Ravi	Mehta	Chemistry	1800.00
7	Rohan	Joshi	Computer Science	1300.00
8	Sonal	Gupta	Economics	1100.00
9	Sunita	Singh	English	2500.00

# Q10). 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.first_name, s.last_name, COUNT(p.payment_id) AS payment_count
FROM Students s
JOIN Payments p ON s.student_id = p.student_id
GROUP BY s.first_name, s.last_name
HAVING COUNT(p.payment_id) > 1;

Results Messages

first_name last_name payment_count
```

# Q11) 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.first_name, s.last_name, SUM(p.amount) AS total_payments
FROM Students s
JOIN Payments p ON s.student_id = p.student_id
GROUP BY s.first_name, s.last_name;
```

			100 %
	ssages	sults 🕫 Mes	⊞ Re
total_payments	last_name	first_name	
2200.00	Doe	John	1
1000.00	Gupta	Ishaan	2
1100.00	Gupta	Sonal	3
1300.00	Joshi	Rohan	4
1800.00	Mehta	Ravi	5
1600.00	Nair	Anita	6
1200.00	Patel	Rajesh	7
2200.00 1000.00 1100.00 1300.00 1800.00 1600.00	Doe Gupta Gupta Joshi Mehta Nair	John Ishaan Sonal Rohan Ravi Anita	2 3 4 5 6

1400.00

2500.00

Rao

Singh

Manish

Sunita

# Q12) 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

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

100 % 
Results Messages

course_name student_count
```

	course_name	student_count
1	Biology	1
2	Chemistry	1
3	Computer Science	1
4	Economics	1
5	English	1
6	History	1
7	Mathematics	1
8	Political Science	1
9	Statistics	1

# Q13) 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

```
SELECT s.first_name, s.last_name, AVG(p.amount) AS average_payment FROM Students s

JOIN Payments p ON s.student_id = p.student_id

GROUP BY s.first_name, s.last_name;
```

100 %	6 ▼ ◀		
⊞ Re	■ Results		
	first_name	last_name	average_payment
1	John	Doe	2200.000000
2	Ishaan	Gupta	1000.000000
3	Sonal	Gupta	1100.000000
4	Rohan	Joshi	1300.000000
5	Ravi	Mehta	1800.000000
6	Anita	Nair	1600.000000
7	Rajesh	Patel	1200.000000
8	Manish	Rao	1400.000000

2500.000000

Sunita

Singh