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

SELECT T.teacher_id, T.first_name, T.last_name,

(SELECT SUM(P.amount)

FROM Payments P

INNER JOIN Enrollments E ON P.student_id = E.student_id

INNER JOIN Courses C ON E.course_id = C.course_id

WHERE C.teacher_id = T.teacher_id

AS total_payments FROM Teacher T;
```

111 % 🔻 🖣

| | teacher_id | first_name | last_name | total_payments |
|----|------------|------------|-----------|----------------|
| 1 | 1 | vibha | bhatore | 1300.00 |
| 2 | 2 | sandhya | ggete | 1350.00 |
| 3 | 3 | sudhir | sharma | 800.00 |
| 4 | 4 | smriti | irani | 400.00 |
| 5 | 5 | gaduda | hariye | NULL |
| 6 | 6 | vandana | sohani | NULL |
| 7 | 7 | mukesh | prajapati | NULL |
| 8 | 8 | shailendra | khede | NULL |
| 9 | 9 | sarthak | sharma | NULL |
| 10 | 10 | gangadhar | sharma | NULL |

```
SQLQuery9.sql - Z...ARSH SHARMA (56))* 

SQLQuery8.sql - Z...ARSH SHARMA (65))*
                                                                             SQLQuery7.sql - Z...ARSI
   □/* 5. Identify students who are enrolled in all available courses.
     Use subqueries to compare a student's enrollments with the total number of courses.*/
   ■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;
101 % ▼ <
first_name
             last_name
                      course_name
     vandana
             sohani
                      Art
```

2

mohini

jessica

vaishali

khushbu

sunita

adarsh

adarsh

khushbu

rathore

sharma

sharma

sharma

sharma

sharma

sharma

sharma

Computer Science

History

Biology

Physics

Chemistry

Mathematics English Literature

Mathematics

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

SELECT T.first_name, T.last_name
FROM Teacher T
WHERE T.teacher_id NOT IN (SELECT C.teacher_id FROM Courses C);

| | first_name | last_name |
|---|------------|-----------|
| 1 | gaduda | hariye |
| 2 | vandana | sohani |
| 3 | mukesh | prajapati |
| 4 | shailendra | khede |
| 5 | sarthak | sharma |
| 6 | gangadhar | sharma |
| | | |

SQLQuery11.sql -...ARSH SHARMA (51))* * X SQLQuery10.sql -...ARSH SHARMA (61))* SQLQuery9.sql - Z...ARSH SHARMA (56))* SQLQuery8.sql - Z...ARSH SHARMA (65))*

-- 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(age) AS average_age
FROM (SELECT FLOOR(DATEDIFF(YEAR, date_of_birth, GETDATE())) AS age FROM Students) AS AgeCalculation;

101 %

Results Messages

average_age
1 21

```
SQLQuery13.sql -...ARSH SHARMA (62))*  

SQLQuery12.sql -...ARSH SHARMA (72))*  

SQLQuery11.sql -...ARSH SHARMA (51))*  

SQLQuery10.sql -...ARSH SHARMA (61))*  

SQLQuery10.sql -...ARSH SHARMA (61))*
```


| | student_id | first_name | last_name | course_name | total_payments |
|---|------------|------------|-----------|--------------------|----------------|
| 1 | 1 | adarsh | sharma | English Literature | 500.00 |
| 2 | 1 | adarsh | sharma | Mathematics | 500.00 |
| 3 | 2 | sunita | sharma | Mathematics | 300.00 |
| 4 | 3 | khushbu | sharma | Biology | 400.00 |
| 5 | 3 | khushbu | sharma | Physics | 400.00 |
| 6 | 4 | vaishali | sharma | Chemistry | 200.00 |
| 7 | 6 | jessica | sharma | History | 450.00 |
| 8 | 7 | mohini | rathore | Computer Science | 600.00 |
| 9 | 8 | vandana | sohani | Art | 500.00 |

```
SQLQuery14.sql -...ARSH SHARMA (54))* 

□ × SQLQuery13.sql -...ARSH SHARMA (62))*
                                                                               SQL
   □/*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 t.first_name, t.last_name

     FROM Teacher t
     LEFT JOIN Courses c ON t.teacher_id = c.teacher_id
    WHERE c.course_id IS NULL;
101 %
last_name
    first_name
     gaduda
              hariye
2
     vandana
              sohani
     mukesh
              prajapati
     shailendra
              khede
     sarthak
              sharma
5
```

gangadhar

sharma

SQLQuery15.sql -...ARSH SHARMA (59))* ** X SQLQuery13.sql -...ARSH SHARMA (62))* SQLQuery14.sql -...ARSH SHARMA (54))* SQLQuery12.sql -...ARSH SHA

300.00

400.00

200.00

350.00

450.00 600.00

500.00

700.00

550.00

NULL

sharma

sharma

sharma

sharma

rathore

sohani

rajguru

gittu

gavshinde

2

3

5 5

6 6

7

8

10

11

4

9

10

sunita

khushbu

vaishali

arpit

jessica

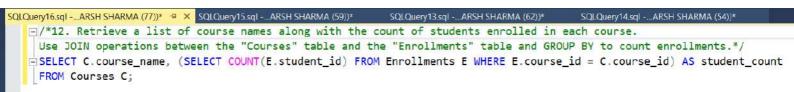
mohini

sittu

sonali

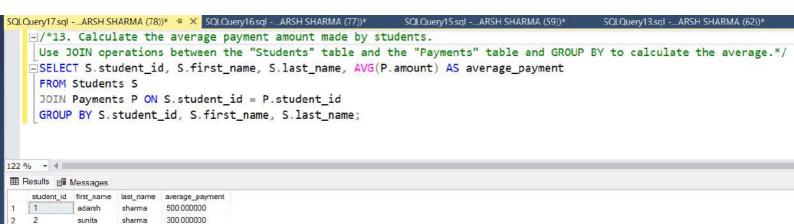
John

vandana



122 % ▼ ◀ ■ Results № Messages

| | course_name | student_count |
|----|--------------------|---------------|
| 1 | Mathematics | 2 |
| 2 | English Literature | 1 |
| 3 | Biology | 1: |
| 4 | Chemistry | 1 |
| 5 | Physics | 1 |
| б | History | 1 |
| 7 | Computer Science | 1 |
| 8 | Art | 1 |
| 9 | Music | 0 |
| 10 | Economics | 0 |



3

5

5

6

8

10 10

khushbu

vaishali

mohini

sonali

vandana

arpit jessica sharma

sharma

sharma

rathore

sohani

gittu

rajquru

400.000000

200.000000

450 000000

600.000000

500.000000

700.000000 550.000000

gavshinde 350,000000