

Q1. 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.

SQLQuery4.sql - Z...ARSH SHARMA (53))* SQLQuery3.sql - Z...ARSH SHARMA (59))* SQLQuery1.sql - not connected*

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

WHERE s.student_id = 1

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

164 %

Results

Messages

	first_name	last_name	total_payments
1	adarsh	sharma	500.00

Q2. 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_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_name;
```

149 %

Results Messages

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

Q3. 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.first_name, s.last_name
FROM Students s
LEFT JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.enrollment_id IS NULL;
```

149 %

Results Messages

	first_name	last_name
1	arpit	gavshinde
2	sittu	gittu
3	sonali	rajguru
4	John	Doe

Q4. 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;
```

149 %

Results Messages

	first_name	last_name	course_name
1	vandana	sohani	Art
2	mohini	rathore	Computer Science
3	jessica	sharma	History
4	vaishali	sharma	Chemistry
5	khushbu	sharma	Biology
6	sunita	sharma	Mathematics
7	adarsh	sharma	English Literature
8	adarsh	sharma	Mathematics
9	khushbu	sharma	Physics

Q5. Create a query to list the names of teachers and the courses they are assigned to. Join the "Teacher" table with the "Courses" table.

SQLQuery21.sql - ...ARSH SHARMA (90))* SQLQuery20.sql - ...ARSH SHARMA (89))* SQLQuery19.sql - ...ARSH SHARMA (88))*

SELECT t.first_name, t.last_name, c.course_name
FROM Teacher t
JOIN Courses c ON t.teacher_id = c.teacher_id;

149 %

Results

Messages

	first_name	last_name	course_name
1	vibha	bhatore	Mathematics
2	sandhya	ggete	English Literature
3	sandhya	ggete	Biology
4	sudhir	sharma	Chemistry
5	smriti	irani	Physics
6	sandhya	ggete	History
7	sudhir	sharma	Computer Science
8	vibha	bhatore	Art
9	smriti	irani	Music
10	sandhya	ggete	Economics

Q6. 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.

SQLQuery23.sql - ...ARSH SHARMA (66))* SQLQuery21.sql - ...ARSH SHARMA (90))* SQLQuery20.sql - ...

SELECT 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 = 2;

149 %

Results

Messages

	first_name	last_name	enrollment_date
1	adarsh	sharma	2023-01-11

Q 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.

SQLQuery25.sql - ...ARSH SHARMA (75))* SQLQuery23.sql - ...ARSH SHARMA (66))* SQLQuery21.sql

SELECT 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;

149 %

Results

Messages

	first_name	last_name
1	John	Doe

Q8. 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.

SQLQuery26.sql - ...ARSH SHARMA (77))* SQLQuery25.sql - ...ARSH SHARMA (75))* SQLQuery23.sql - ...AR

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

149 %

Results Messages

	course_name
1	Music
2	Economics

Q9. 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.

SQLQuery1.sql - Z...ARSH SHARMA (51))*

```
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;
```

122 %

Results Messages

	student_id	first_name	last_name
1	1	adarsh	sharma
2	3	khushbu	sharma

Q10. 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.

SQLQuery3.sql - Z...ARSH SHARMA (59))* SQLQuery1.sql - not connected*

```
SELECT s.first_name, s.last_name, COUNT(e.course_id) AS course_count
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
GROUP BY s.student_id, s.first_name, s.last_name
HAVING COUNT(e.course_id) > 1;
```

164 %

Results Messages

	first_name	last_name	course_count
1	adarsh	sharma	2
2	khushbu	sharma	2