

SQL JOIN Exercises: 4Question 01 - inner joining

Select A.student_id, student_name, grade,
 From student As A
 INNER JOIN grades As B
 ON A.student_id = B.student_id;

student_id	student_name	grade
2	Bob	B
3	charlie	A

Question 2 - left join

SELECT A.emp-id, emp-name, depart_name
 From employees As A
 Left JOIN department As B
 ON A.emp-id = B.emp-id;

emp-id	emp-name	dept_name
1	John	NULL
2	Lisa	HR
3	Mike	NULL

Question 3 - Full OUTER JOIN

SELECT COALESCE(p.product-id, s.product-id) As product_id, product_name,
 s.quantity)
 FROM product p
 FULL OUTER JOIN sales ON p.product-id = s.product-id;

product_id	Product_name	quantity
1	Laptop	NULL
2	Mouse	50
3	Keyboard	NULL

Question 4: Left Join + case

```

SELECT o.order_id, o.customer_id, o.amount, c.customer_name,
CASE
    WHEN c.customer_id IS NULL THEN 'New customer'
    ELSE 'Returning customer'
END AS customer_type

```

FROM orders o

Left Join customers c ON o.customer_id = c.customer_id;

order_id	customer_id	amount	customer_name	customer_type
1	101	500	Paul	Returning customer
2	102	300	Sarah	Returning customer
3	105	0	NULL	New customer

Question 5: LEFT + GROUP BY + SUM

```

SELECT r.region_id, r.region_name, COALESCE(SUM(s.amount), 0)
    AS total_sales

```

FROM regions r

Left JOIN sales s ON r.region_id = s.region_id

GROUP BY r.region_id, r.region_name;

region_id	region_name	total_sales
1	North	2000
2	South	3500
3	East	0

Question 6 - LEFT JOIN + CASE

```
SELECT s.student_id, s.name, a.days-present,  
CASE  
    WHEN a.days-present IS NULL THEN 'Poor Attendance'  
    WHEN a.days-present >= 15 THEN 'Excellent'  
    WHEN a.days-present >= 10 THEN 'Needs improvement'  
    ELSE 'Poor Attendance'
```

END AS attendance-status

FROM students s

LEFT JOIN attendance a ON s.student_id = a.student_id;

student_id	name	day-present	attendance-status
1	Alice	18	Excellent
2	Bob	5	Poor Attendance
3	Charlie	NULL	Poor Attendance

Question 7 - INNER JOIN + COUNT + GROUP BY

```
SELECT p.product_id, p.name, COUNT(t.task_id) AS task-count  
FROM projects p
```

```
INNER JOIN task t ON p.project_id = t.project_id  
GROUP BY p.product_id, p.name;
```

Project-id	name	task-count
1	AI chatbot	2
2	website	1

Question 8 - FULL OUTER JOIN + CASE + WHERE

```
SELECT COALESCE(o.cust_id, r.cust_id) AS cust_id, o.order-total,  
r.return-total;
```

CASE

```
    WHEN r.return-total IS NOT NULL THEN 'Returned'  
    ELSE 'No Return'
```

END AS return-status

FROM order o

cust_id	order_total	return_total	return_status
11	120	20	Returned
12	250	NULL	No return
13	180	NULL	No return

Question 9 - Left JOIN + COUNT + ORDER BY

SELECT u.user_id, u.name, COUNT(l.login_date) AS login_count
FROM user u

LEFT JOIN logins l ON u.user_id = l.user_id

GROUP BY u.user_id, u.name

ORDER BY login_count DESC;

user_id	name	login_count
2	Gloria	2
3	Steve	1
1	Nelson	0

Question 10 - Left JOIN + CASE + ORDER BY

SELECT t.teacher_id, t, teacher_name,
CASE
WHEN s.subject_name IS NULL THEN 'No subject Assigned'
ELSE s.subject_name
END AS subject_name

FROM teachers t
LEFT JOIN subject s ON t.teacher_id = s.teacher_id
ORDER BY t.teacher_name ASC;

teacher-id	teacher-name	Subject-name
3	Mr. Dlamini	No Subject Assigned
1	Mr Hlongwane	Math
1	Mr Hlongwane	Science
2	Ms. Ndaba	No Subject Assigned

