SQL JOIN Exercise

Question 1 -> INNER JOIN

Table: students
Table: grades

```
SELECT student_id,
       student_name,
       grade
FROM students AS A
INNER JOIN grades AS B
ON A.student_id = B.student_id;
```

Output:

student_id	student_name	grade
2	Bob	B
3	Charlie	A

QUESTION 2 → LEFT JOIN

Table. employees
Table. departments

```
SELECT emp_id,
       emp_name,
       dept_name

FROM employees AS A
LEFT JOIN AS B
ON A. emp_id = B. emp_id;
```

output:

emp_id	emp_name	dept_name
1	John	NULL
2	Lisa	HR
3	Mike	NULL

Question 3 → FULL OUTER JOIN

Table : products
Table : sales

```
SELECT product_id,
       product_name,
       quantity
FROM products AS A
FULL OUTER JOIN AS B
ON A.product_id = B.product_id;
```

product_id	product_name	quantity
1	Laptop	NULL
2	Mouse	50
3	keyboard	NULL
4	Null	30

Advanced JOIN Practice (Logic + Filtering + Aggregates)

Question 4 → LEFT JOIN + CASE

Table. orders
Table. customers

```
SELECT order_id,
       customer_id,
       amount,
       customer_name
      CASE
       WHEN B.customer_id IS NOT NULL THEN
       ('Returning Customer'
       ELSE 'New Customer'
       END AS customer_type

FROM orders AS A
LEFT JOIN ~~B~~ customers AS B
ON A.customer_id = B.customer_id;
```

order_id	customer_id	amount	customer_name	customer_type
1	101	500	paul	Returning customer
2	102	300	Sasah	Returning customers
3	105	0	NULL	New customers

Question 5 : LEFT JOIN + GROUP BY + SUM

Table : sales
Table : regions

```
SELECT region_id,
       region_name,
       SUM(amount) AS total_sales
FROM sales AS A
LEFT JOIN AS B
ON A.region_id = B.region_id
GROUP BY B.region_id, region_name;
```

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

Question 6 → LEFT JOIN + CASE

Table. students
Table. attendance

```
SELECT student_id,
       name,
       days_present
       CASE
        WHEN days_present >= 15 THEN 'Excellent'
        WHEN days_present BETWEEN 6 AND 14
        THEN 'Needs Improvement'
        WHEN days_present <= 5 THEN 'Poor Attendance'
        ELSE 'No Record'
       END AS attendance_status

FROM students AS A
LEFT JOIN attendance AS B
ON A.student_id = B.student_id;
```

student_id	name	days_present	attendance_status
1	Alice	18	Excellent
2	Bob	5	Poor Attendance
3	charlie	NULL	No Record

QUESTION 7 → INNER JOIN + COUNT + GROUP BY

Table: projects
Table: tasks

```
SELECT project_id,
       name,
       COUNT(task_id) AS task_count
FROM projects AS A
INNER JOIN tasks AS B
ON A.project_id = B.project_id
GROUP BY A.project_id, name;
```

project_id	name	task_count
1	AI chatbot	2
2	website	1

QUESTION 8 -> FULL OUTER JOIN + CASE + WHERE

Table: orders

Table: returns

```
SELECT cust_id,
       order_total,
       return_total
       CASE
       WHEN return_total IS NOT NULL THEN 'Returned'
       ECSE ('NO return')
    END AS return_status
FROM orders AS A
FULL OUTER JOIN returns AS B
ON A.cust_id = B.cust_id
WHERE order_total > 100;
```

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 + ~~DIS~~ ORDER BY

Table: users
Table: logins

```
SELECT user_id,
       name,
       ~~login_count~~
       COUNT(login_date) AS login_count
FROM users AS A
LEFT JOIN logins AS B
ON A.user_id = B.user_id
ORDER BY login_count DESC;
```

user_id	name	login_count
2	~~Nelson~~ gloria	2
3	steve	1
~~8~~ 1	~~gloria~~ Nelson	0
	~~Steve~~	

user_id	name	login_count
2	gloria	2
3	steve	1
1	Nelson	0

QUESTION 10 - D LEFT JOIN + CASE + ORDER BY

Table: teachers
Table: subjects

```
SELECT teacher_id,
       teacher_name,
       ~~subject_name~~
       COALESCE(subject_name, 'No subject Assigned')
       AS subject_name
FROM teachers AS A
LEFT JOIN subjects AS B
ON A.teacher_id = B.teacher_id
ORDER BY teacher_name ASC;
```

teacher_id	teacher_name	subject_name
1	Mr. Hlongwane	Math
2	Ms. Ndaba	
3	Mr. Dlamini	

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