Exercise 4: SQL JOIN EXERCISES

Q1: SELECT ~~student-id~~,
student-name,
grade
FROM students AS A
INNER JOIN grade AS B
ON A.student-id = B.student-id;

Student-id	Student-name	Grade
2	Bob	B
3	Charlie	A

② LEFT JOIN :
SELECT emp-id,
emp-name,
dept-name
FROM employees AS A
LEFT JOIN departments AS B
ON A.emp-id = B.emp-id;

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

③ SELECT product-id,
product_name
Quantity
FROM products AS A
FULL OUTER JOIN sales 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

④ SELECT order-id,
A.customer-id,
amount,
customer-name
CASE WHEN B.customer-id IS NOT NULL THEN 'Returning customer'
ELSE 'New customer'
END AS customer_type
MS2 ~~LEFT JOIN customers~~ FROM orders AS A
LEFT JOIN 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	Sarah	Returning Customer
3	105	0	NULL	New Customer

MS2

(Q5) SELECT region_id,
region_name,
SUM(amount) AS total_sales
FROM regions AS A
LEFT JOIN Sales AS B
ON A.region_id = B.region_id
GROUP BY A.region_id, region_name;

region_id	region_name	total_sales
1	North	2000
2	South	3500
3	East	NULL
4	NULL	1000

(6) SELECT student_id,
name,
days_present,
CASE WHEN days_present >15 THEN 'Excellent'
WHEN days_present BETWEEN 6 AND 14 THEN 'Needs Improvement'
MS2 ~~ELSE 'Poor Attendance'~~ WHEN days_present <= 5 THEN 'Poor Attendance'
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
4	NULL	20	Excellent

⑦
```
SELECT COUNT(task_id) AS task_count
       project_id,
       name
FROM projects AS A
INNER JOIN tasks AS B
ON A.project_id = B.project_id
GROUP BY project_id, name;
```

Project_id	name	task_count
1	AIChatBot	2
2	Website	1

⑧
```
SELECT coalesce(A.cust_id, B.cust_id) AS cust_id,
       order_total,
       return_total
CASE WHEN return_total IS NOT NULL THEN 'Returned'
     ELSE 'No Return'
  END AS return_status
FROM orders AS A
FULL OUTER JOIN returns AS B
ON A.cust_id = B.cust_id
WHERE orders > 100;
```

Cust_id	Order_total	return_total	return_status
11	120	20	Returned
12	250	NULL	No Return
13	180	NULL	No Return
14	NULL	100	No Return

⑨
```
SELECT user_id,
       name,
       COUNT(login_date) AS login_count
  FROM users AS A
LEFT JOIN logins AS B
ON A.user_id = B.user_id
GROUP BY user_id, name
ORDER BY login_count DESC;
```

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

(10) SELECT teacher-id,
teacher-name
CASE WHEN subject-name IS NULL THEN 'No Subject Assigned'
ELSE 'subject-name'
END AS subject-name
FROM teachers AS A
LEFT JOIN Subjects AS B
ON A. teacher-id = B. teacher-id
ORDER BY teacher-name ASC;

OR

SELECT A. teacher-id,
teacher-name
COALESCE (subject-name, 'No subject Assigned') AS subject-name
FROM teachers AS A
LEFT JOIN subject AS B
ON A. teacher-id = B. teacher-id
ORDER BY 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