

## Exercise 4: SQL Joins

① SELECT student-id,  
student-name,  
grade  
from students  
INNER JOIN grades  
ON A.student-id = B.student-id;

② 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  
 FULL OUTER JOIN sales ON A.product-id =  
                      B.product-id;
```

product-id	product-name	quantity
1	Laptop	NULL
2	Mouse	20
3	Keyboard	NULL
4	NULL	30

④ ~~SELECT order-id,  
 customer-id,  
 amount  
 customer-name  
 CASE WHEN customer-name is NULL~~

④ ~~SELECT order\_id,  
customer\_id,  
amount,  
customer\_name  
FROM orders AS A  
LEFT JOIN customers AS B  
ON A.customer\_id = B.customer\_id;~~

~~order\_id~~

④ ~~SELECT order\_id,  
customer\_id,  
amount,  
customer\_name  
CASE WHEN customer\_name IS NULL THEN  
'new'  
ELSE 'returning'  
END AS customer\_type  
FROM orders AS A  
LEFT JOIN customers ON A.customer\_id = B.customer\_id~~

order_id	customer_id	amount
1	101	506
2	102	306
3	105	0

```
SELECT region - id,  
      region - name,
```

```
TOTAL - SALES = COALESCE (sum (amount), 0)
```

```
FROM regions
```

```
LEFT JOIN sales ON A.region - id = B.region - id  
GROUP BY region - id,  
        region - name;
```

region - id	region - name	total - Sales
1	North	2000
2	South	3500
3	East	0

```
) SELECT student - id,
```

```
name,
```

```
days - present
```

```
CASE WHEN days - present >= 15 THEN 'Excellent'  
      WHEN days - present BETWEEN 10 AND 14
```

```
      THEN 'Needs Improvement'
```

```
      ELSE 'Poor Attendance'
```

```
END AS attendance - status
```

```
FROM students
```

```
LEFT JOIN attendance ON A.student - id = B.
```

```
= B.student - id;
```

student - id	name	days - present
1	Alice	18
2	Bob	5
3	Charlie	Null

⑦ `Select project_id,`

`name`

`COUNT(task_id) AS task_count`

`from projects`

`INNER JOIN tasks ON A.project_id = B.project_id`

`Group By project_id,`

`name,`

project_id	name	task_count
1	AI Chatbot	2
2	Website	1

⑧ `SELECT COALESCE(A.cust_id, B.cust_id) AS`

`cust_id`

`order_total`

`return_total`

`CASE WHEN return_id IS NOT NULL THEN 'Returned'`

`ELSE 'No Return'`

`END AS return_status`

`From orders`

`full outer JOIN returns ON A.cust_id = B.cust_id`

`WHERE COALESCE(order_total, 0) > 100;`

cust_id	order_total	return_total	return_status
11	120	20	Returned
12	250	NULL	No Return
14	100	100	Returned

```

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

```

User_id	Name	Login_Count
2	Gloria	1
3	Steve	1
1	Nelson	0

```

⑩ SELECT teacher_id,
        teacher_name,
        COALESCE(subject_name, 'No subject Assigned') AS 'subject_name'
      FROM teachers
      LEFT JOIN subjects ON A.teacher_id = B.teacher_id
    ORDER BY teacher_name ASC;

```

teacher_id	teacher_name	subject_name
3	Mr. Alainini	No Subject Assigned
1	Mr. Hongnare	Math
1	Mr. Hongnare	Science
2	Mr. Ndaba	No Subject Assigned