

## Exercise 4: SQL JOIN Exercise

1. SELECT A. student\_id,  
student\_name,  
grade  
FROM students AS A  
INNER JOIN grades AS B  
ON A.student\_id = B.student\_id;

Student_id	Student_name	grade
2 ✓	Bob Alice	N/A - B
3 ✓	Charlie	B - A
3	Charlie	

~~Base table~~ → ~~Focus table~~

2. SELECT A.emp-id,  
emp name,  
dept-name

FROM employees AS A

LEFT JOIN departments AS B

ON A.emp-id = A.emp-id;

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

3. SELECT COALESCE(A.product-id, B.product-id)  
product-id, AS product-id,  
product-name,  
quantity

FROM products AS A

PULL 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,  
customer\_id,  
amount,  
customer\_name

CASE WHEN B. customer\_id IS NOT NULL THEN  
'Returning Customer'  
ELSE 'New customer'  
FROM orders AS A

LEFT JOIN customers AS B ←

ON A. customer\_id = B. customer\_id

CASE WHEN amount ≥ 300 THEN 'Returning'  
ELSE 'New'

END AS 'customer-type'

FROM orders AS A —————

order_id	customer_id	customer_name	Amount	customer-type
1 ✓	101 ✓	Paul ✓	500 ✓	Returning ✓
2 ✓	102 ✓	Sarah ✓	300 ✓	Returning ✓
3 ✓	105 ✓	Matt ✓	0 ✓	New ✓

5. SELECT region\_id,  
region\_name,  
sum(amount) AS total\_sales

FROM sales AS A

LEFT JOIN regions AS B

ON A. region\_id = B. region\_id

GROUP BY region\_name.

~~HAVING SUM(Amount)=0;~~

region_id	region_name	total_sales
1 ✓	North ✓	2000
2 ✓	South ✓	3500
3 ✓	East ✓	0 NULL
4	NULL ✗	1000

6. SELECT student\_id,

name,

days-present

FROM Students AS A

LEFT JOIN attendance AS B

ON A.student\_id = B.student\_id;

CASE WHEN days-present = 0 THEN 'Excellent'

WHEN days-present BETWEEN 6 AND 14 THEN 'Need 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

Needs Improvement

2 ✓

Bob

5 ✓

Poor

Needs Improvement

3 ✓

Charlie

NULL ✗

No Record

Poor attendance

7. ~~SELECT project-id,~~  
~~name,~~  
~~count(task-id) AS task-count~~  
~~COUNT(task-id) AS task-count~~  
~~FROM projects AS A~~  
~~INNER JOIN tasks AS B~~  
~~ON A.project-id = B.project-id;~~  
  
~~COUNT(task-id) AS task-count~~  
~~FROM projects AS~~  
~~GROUP BY A.project-id,~~  
~~name;~~

Project-id	name	task-count
1 ✓	AI Chatbot	2 ✓
2 ✓	Website	1 ✓

~~COALESCE(A.cust-id, B.cust-id) AS cust-id~~  
~~Customer~~  
 8. ~~SELECT cust-id,~~  
~~order-total~~  
~~return-total~~  
~~CASE WHEN return-total IS NOT NULL THEN 'Returned'~~  
~~From FullOuterJoin Returns AS A~~  
~~END AS return-status~~  
~~FullOuterJoin Orders AS B~~  
~~ON A.cust-id = B.cust-id~~  
~~ON A.cust-id = B.cust-id~~  
~~WHERE order-total > 100;~~

9. SELECT A.user\_id,  
A.name,  
B.login\_date  
COUNT (user\_id) AS login\_count  
FROM users AS A  
LEFT JOIN logins AS B  
ON A.user\_id = B.user\_id  
.GROUP BY A.name, B.login\_date  
ORDER BY login\_count DESC;

10. SELECT A.teacher\_id,  
teacher\_name,  
Subject\_id,  
teacher\_id,  
subject\_name  
CASE WHEN subject\_name IS NULL THEN 'No S'  
WHEN Subject\_name IN ('Math', 'S', 'H')  
END AS teachers\_subjects  
THEN 'Subject'  
ORDER BY teacher\_name ASC.  
LEFT JOIN subjects AS S  
ON b.teacher\_id = s.teacher\_id