

## Exercise 2 : SQL JOINS

### (1) INNER JOIN

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	B
3	charlie	A

### (2) LEFT JOIN

SELECT A.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	Hull
2	Tom	HR
3	Mike	Hull

### FULL OUTER JOIN

(3) SELECT

COALESCE (A.product\_id, B.product\_id) AS 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

### LEFT JOIN + CASE

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

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	Hull	New Customer



Scanned with  
MOBILE SCANNER

## LEFT JOIN + GROUP BY + SUM

(5) SELECT A.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

## LEFT JOIN + CASE

SELECT A.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

### INNER JOIN + COUNT + GROUP BY

(7) SELECT A.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

### FULL OUTER JOIN + CASE + WHERE

SELECT

COALESCE(A.cust-id, B.cust-id) AS cust-id,  
order-total, return-total,

CASE

WHEN return-total > 0 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



cust-id	order-total	return-total	return-status
11	120	20	Returned
12	250	NULL	No Return
13	180	NULL	No Return

### LEFT JOIN + COUNT + ORDER BY

(9) `SELECT A.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 A.user_id, name  
ORDER BY login-count DESC`

↓

user_id	name	login-count
2	Gloria	2
3	Steve	1
1	Nelson	0

### LEFT JOIN + CASE + ORDER BY

`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 Mdlaba	No subject assigned