Exercise4 : JOINS

Q1 - Inner Join

Objective : Join students and grades to display only students who have grades.

Table : students

student_id	student_name
1	Alice
2	Bob
3	Charlie

Table : grades

student_id	grade
2	B
3	A
4	C

```
SELECT student_id,
       student_name,
       grade
FROM students As A
INNER JOIN grades
As B ON
A.student_id = B.student
_id
```

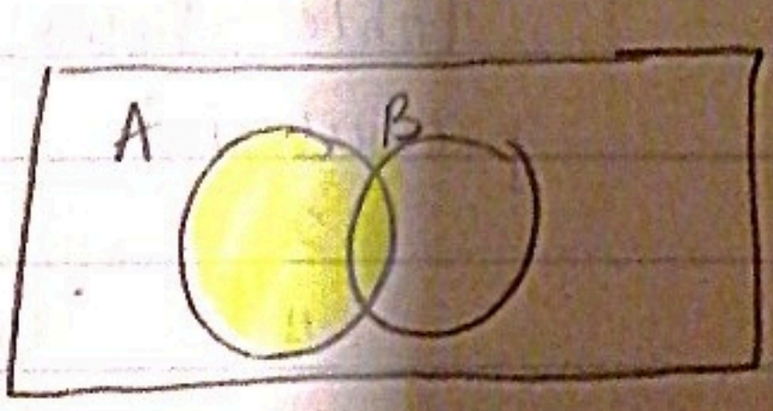
Exp. col / Output :

* student_id
* student_name
* grade

student_id	student_name	grade
2	Bob	B
3	Charlie	A

- Returns data that is in both columns have.
- ~~Matching both~~ Return is always matching in both tables.

Q2. - Left Join - eg. I only have table A but I also want something in table B.



We only take only what matches in both tables.
(Common column)

Objective - Display all employees & the departments they belong to. Include employees with no department.
Exp Output : emp_id, emp_name, dept_name

Table : employees

emp_id	emp_name
1	John
2	Lisa
3	Mike

Table : departments

emp-id	dept-name
2	HR
4	IT

```
SELECT emp_id,
       emp_name      left table
       dept_name
FROM employees As A
LEFT JOIN departments
As B ON A.emp_id
   = B.emp_id;
                Right table
```

Return : - We return a null if there is no match.

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

Question 3 - FULL OUTER JOIN
Objective : Display a complete list of products and their quantities sold. Include products with no sales and sales for unknown products.
Exp output : product_id, product_name

Table : products

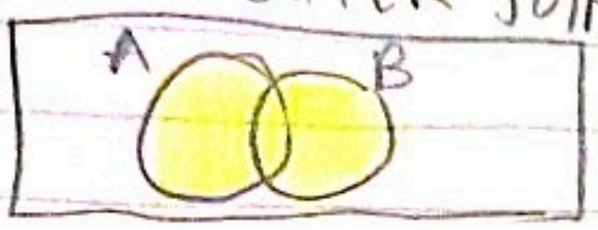
product_id	product_name
1	Laptop
2	Mouse
3	Keyboard

Tables' Sales

product_id	quantity
2	50
4	30

- It returns all the rows in both tables

FULL OUTER JOIN




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

- For the unmatching row we fill in NULL

Return:

product_id	product_name	quantity
1	Laptop	NULL
2	Mouse	50
3	Keyboard	NULL — Return NULL because we don't know the value
4	NULL	30

Put NULL where there are spaces

Question 4: LEFT JOIN + CASE

Objective: Display all orders and indicate whether the customer is "New" or "Returning"

Table: orders

order_id	customer_id	amount
1	101	500
2	102	300
3	105	0

Table: customers

customer_id	customer_name
101	Paul
102	Sarah
104	Emma

Expected Output: order_id, customer_id, amount, customer_name, customer_type ('New Customer'/'Returning Customer')

TABLE A and TableB

```
SELECT A.order_id,
       A.customer_id,
       A.amount,
       B.customer_name
FROM orders As A
LEFT JOIN customers As B. ON A.customer_id
= B.customer_id;
```

Returns everything in orders table as we calculate based on orders taker.

table A

take from table A.

left join

right join

Return

order_id	customer_id	amount	customer_name
1	101	500	Paul
2	102	300	Sarah
3	105	0	NULL

~~Question 5~~ - LEFT JOIN + GROUP BY + SUM

~~Objective~~: Show total sales per region and include regions with no sales.

Tables: orders

order_id	customer_id	amount
1	101	500
2	102	300
3	105	0

Table: customers

customer_id	customer_name
101	Paul
102	Sarah
104	Emma

Q4

LEFT JOIN

(customer_id common column)

Exp col: order_id, customer_id, amount, customer_name, customer_type (New Customer / Returning Customer)

SELECT - Qs. Table

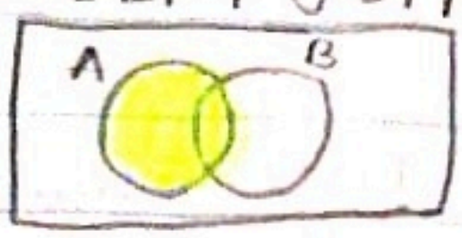
Table : sales

sale_id	region_id	amount
1	1	2000
2	2	3500
3	4	1000

Table : region

region_id	region_name
1	North
2	South
3	East

LEFT JOIN



(common column is region_id)

Qs : "LEFT JOIN + GROUP BY + SUM

Objective : Show total sales per region and include region with no sales.

Exp col : region_id, region_name, total sales

SELECT A. region_id,
B. region_name,
SUM(amount) As total_sales
WHERE a·

SELECT A.region_id,
B. region_name,
CO SUM (sale.amount), 0) As total_sales
FROM sales As A
LEFT JOIN sales region As B. ON region_name = B
region_id ;

Return :

region_id	region_name	total_sales
1	North	2000
2	South	3500
3	East	0

Question c : LEFT JOIN + CASE

Objective : Classify students based on attendance.

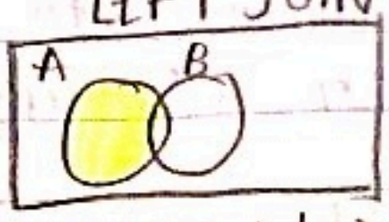
Table : students

student_id	name
1	Alice
2	Bob
3	Charlie

Table : attendance

student_id	days_present
1	18
2	5
3	20

LEFT JOIN



(student_id is common in both tables)

Exp col : student_id, name, days_present, attendance_status (Excellent / Needs Improvement / Poor Attendance)

```
SELECT Astundent_id,
       A. name,
       B. days_present,
   CASE
      WHEN days_present ≥ 18 20 THEN 'Excellent'
      WHEN days_present BETWEEN 5 AND 10 THAN
                        =18 THEN 'Needs Improvement'
      WHEN days_present < 20 5 THEN 'Poor Attendance'
   END As attendance_status
FROM students
LEFT JOIN att days_present As B ON A student_id =
B. student_id ;
```

15/10/2025 Practical 1 : SQL Fundamentals (Swayplate Base SQL Syntax)
Database : RETAIL_SALES — Practical
Schema : SALES dataset: sales retail
to Create new : ITEMS — sales data items sales

Return :

student_id	name	days_present	attendance_status
1	Alice	18	Needs Improvement
2	Bob	5	Poor Attendance
3	Charlie	NULL 20	Excellent

Question 7 - INNER JOIN + COUNT + GROUP BY
Objective : Show number of tasks per project. Only include projects that have tasks.

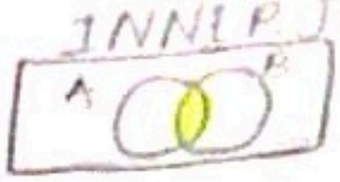
Table : projects

project_id	name
1	AI Chatbox
2	Website
3	Data Report

Table : p tasks

task_id	project_id	status
10	1	Complete
11	1	Pending
12	2	Complete

Exp col : project_id, name, task_count




```
SELECT A.project_id,
       A name,
       COUNT (task_id) As task_count
FROM projects
INNER JOIN project_id As B ON A.project_id
= B.project_id
GROUP B 4 project_id, project_name;
```

Return :

project_id	name	task_count
1	AI Chatbox	2
2	Website	1

Question 8 - FULL OUTER JOIN + CASE + WHERE

Objective : Classify customers based on whether they turned anything and filter by high order total.

Table : orders

order_id	cust_id	order_total
1	11	120
2	12	250
3	13	180

Table : returns

return_id	cust_id	return_total
101	11	20
102	14	100

Exp col : cust_id, order_total, return_total, return_status (Returned / No Return

```
SELECT A.cust_id,
       A.order_total,
       B.return_total,
   CASE
     WHEN return_total IS NOT NULL THEN 'Returned'
     ELSE 'No Return'
END As return_status
FROM orders
FULLER OUTER JOIN returns ON A.cust_id = B.return_id
WHERE A.order_total > 100;
```

Return !

cust_id	order-total	return_total	return_status
11	120	20	Returned
12	250	NULL	No Return
13	180	NULL	No Return

Q 9. - LEFT JOIN + COUNT + ~~OR~~ + ORDER BY

Objective : Count how many times each user logged in.

Table ! users

user_id	name
1	Nelson
2	Gloria
3	Steve

Table ! Logins

user_id	login_date
2	2024-06-01
2	2024-06-02
3	2024-06-03

Exp col ! user_id, name, login_count. Order by login_count DESC.

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

Return !

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

Exercise 4

Question 10 - LEFT JOIN + CASE + ORDER BY

Objective: Show all teachers and all subjects they teach. If no subject, label appropriately.

Table: teachers

teacher_id	teacher_name
1	Mr Hlongwane
2	Ms Ndaba
3	Mr Dlamini

Table: subjects

subject_id	teacher_id	subject_name
1	1	Math
2	1	Science
3	4	History

Exp col: teacher_id, teacher_name, subject_name (or "No Subject Assigned")
Order by teacher_name ASC.

```
SELECT A.teacher_id,
       A.teacher_name,
   COALESCE(B.subject_name, 'No Subject Assigned' As
       stu subject_name
FROM teachers
LEFT JOIN subjects B ON A.teacher_id = B.teacher_id
ORDER BY B.teacher_name ASC;
```

Return:

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