

## EXERCISE 4: SQL JOINS

### Question 1: Inner Join

: Join Students and Grades to display Only Student who have grades.

```
SELECT S.Student-id, Student_name,  
Grade
```

```
FROM Students AS S
```

```
INNER JOIN Grade AS G
```

```
ON S.Student-id = G.Student-id;
```

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

### Question 2: LEFT JOIN

: Display all employees and the department they belong to. Include employees with no department.

```
SELECT E.Emp-id, Emp-Name,  
Dept-name
```

```
FROM Employees AS E
```

```
LEFT JOIN Department AS D
```

```
ON E.Emp-id = D.Emp-id;
```

Emp-id	Emp-Name	Dept-Name
1	John	Null
2	Lisa	HR
3	Mike	Null

Question 3: Full OUTER Join

: Display a complete list of products & their quantities Sold. Include products with ~~their~~ no sales & sales for Unknown product

```
SELECT P.Product_id, Product_name,  
       Quantity
```

```
FROM Products AS P
```

```
Full OUTER JOIN Sales AS S
```

```
ON P.Product_id = S.Product_id;
```

Product id	Product Name	Quantity
1	Laptop	Null
2	Mouse	50
3	Keyboard	Null
4	Null	30

Question 4: LEFT JOIN & CASE

: Display all orders & ~~indicate~~ indicate whether the Customer is 'New or Returning'

```
SELECT Order_id, Customer_id,  
       Amount, Customer_Name,  
       Count(Order_id) AS Order_Count,  
       CASE
```

```
       WHEN Order_Count <= 1 THEN 'New'  
       ELSE 'Returning'
```

```
       END AS Customer_Type
```

```
FROM Orders AS O
```

```
LEFT JOIN Customers AS C
```

```
ON O.Order_id = C.Order_id
```

```
GROUP BY Order_id, Customer_id, Customer_Type,  
         Amount, Customer_name;
```



Order id	Customer id	Amount	Customer Name	Customer type	Order
1	101	500	Paul	New	1
2	102	300	Sarah	New	1
3	105	0	Null	New	1

Question 5 : LEFT Join + Group By + Sum

: Show total sales per region & Include regions with no sales

```
SELECT Region-id, Region-name,
       Sum (Amount) AS Total-Sales
```

```
FROM Sales AS S
```

```
FULL OUTER JOIN Regions AS R
```

```
ON S.Region-id = R.Region-id
```

```
GROUP BY Region-id, Region-name;
```

Region-id	Region-Name	Total Sales
1	North	2000
2	South	3500
3	East	Null
4	Null	Null

Question 6 : LEFT Join + CASE

: Classify students based on attendance

```
SELECT Student-id, Name,
       Days-present,
```

```
CASE
```

```
WHEN Days-present Between 0
```

```
AND 5 THEN 'Poor Attendance'
```

```
WHEN Days-present Between 6
```

```
AND 15 THEN 'Needs-Improvement'
```

```
WHEN Days-present > 15 THEN 'Excellent'
```

END AS Attendance\_Status

FROM Students AS S

LEFT JOIN Attendance AS A

ON S.Student\_id = A.Student\_id;

Student_id	Name	Days Present	Attendance Status
1	Alice	18	Excellent
2	Bob	5	Poor Attendance
3	Charlie	Null	Poor Attendance

Question 7: INNER JOIN + COUNT + GROUP BY

: Show number of tasks per project. Only include projects that have tasks.

SELECT Project\_id, Name,  
COUNT(task\_id) AS Task\_Count

FROM Projects AS P

INNER JOIN Tasks AS T

ON P.Project\_id = T.Project\_id

GROUP BY Project\_id, Name;

Project_id	Name	Task Count
1	AI Chatbot	2
2	Website	1
3	Data Report	Null



Question 8 : Full Outer Join + CASE + WHERE

: Classify Customers based on whether they returned anything and filter by high Order total

```
SELECT Cust_id, Order_Total,  
       Return_total,
```

```
       CASE
```

```
         WHEN Return_Total IS null THEN 'No Return'
```

```
         WHEN Return_Total = 0 THEN 'No_Return'
```

```
         ELSE 'Returned'
```

```
       END AS Return_Status
```

```
FROM Orders AS O
```

```
Full Outer Join Returns AS R
```

```
ON O.Cust_id = R.Cust_id
```

```
WHERE Order Total > 150;
```

Cust id	Order total	Return total	Return Status
12	250	Null	No_Return
13	180	Null	No_Return

Question 9 : LEFT Join + Count + ORDER BY

: Count how many times each user logged in

```
SELECT User_id, Name,
```

```
       Count ( Login date ) AS login Count
```

```
FROM Users AS U
```

```
LEFT Join Logins AS L
```

```
ON U.User_id = L.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	Null

Question 10 : LEFT JOIN + CASE + ORDER BY

: Show all teachers & the subjects they teach. If no subject, label appropriately

SELECT Teacher-id, teacher name,

~~Subject\_name,~~

CASE

WHEN Subject Name Is null

THEN 'Appropriately'

ELSE 'Subject\_name'

END AS 'Subject\_name'

FROM Teacher AS T

LEFT JOIN Subjects AS S

On T.Teacher-id S.Teaching

On T.Teacher-id = S.Teacher-id

ORDER BY Teacher\_name Asc;

Teacher id	Teacher Name	Subject Name
Ms 2	Ms. Ndaba	Appropriately
3	Mr. Dlamini	Appropriately
1	Mr. Hlongwane	Math
1	Mr. Hlongwane	Science