Exercse 4 Joins

g. Syntax :

```
SELECT      column 1,
            column 2,
            column 3.
FROM table 1   AS  A
INNER JOIN   table 2  AS  B
ON  A. ~~column~~ common_column  =  B. common_column;
```

1. INNER JOIN

```
SELECT      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

LEFT JOIN

```
2. SELECT ~~product~~ emp_id,
                      emp_name,
                      dept_name
   FROM employees AS A
   LEFT JOIN departments AS B
   ON A.emp_id = B.emp_id; ~~AS~~
```

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

FULL OUTER JOIN

```
3. SELECT product_id,
          product_name,
   ~~FROM~~   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

```
4.  SELECT  order_id,
            customer_id,
            amount,
            customer_name,
    CASE
        WHEN  customer_name  IN ( 'Paul', 'Sarah' )
        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~~	~~cust_type~~
~~1~~				
~~2~~				
~~3~~				
~~NULL~~				

order_id	customer_id	amount	customer_name	customer_type
1	101	500	Paul	Returning Customer
2	102	300	Sarah	Returning Customer
3	105	0	NULL	New Customer
~~NULL~~	[illegible]	~~NULL~~	~~Emma~~	

5.
```
SELECT region_id,
       region_name,
  SUM(amount) AS total_sales
FROM ~~regions~~ sales AS A
LEFT JOIN ~~students~~ region AS B
ON A. region_id = B. region_id
GROUP BY region_id ;
```

region_id	region_name	total_sales
1	North	2000
2	South	3500
4	~~East~~ NULL	1000

6.
```
SELECT student_id,
       name,
       days_present,
  CASE
    WHEN days_present ≥ ~~20~~ 15 THEN 'Excellent'
    ~~WHEN days_present = 18 THEN 'Needs Improvement'~~
    ~~ELSE 'Poor~~
    WHEN days_present BETWEEN 6 AND ~~19~~ 14 THEN
    'Needs Improvement'
    ELSE 'Poor Attendance'
END AS attendance_status
FROM students AS A
LEFT JOIN attendance AS B
ON A. ~~reg~~ student_id = B. student_id ;
```

student_id	name	days present	attendance_status
1	Alice	18	Needs Improvement
2	Bob	5	Poor Attendance
3	Charlie	NULL	~~Poor Attendance~~ NULL

```
7  SELECT A.project_id,
          name
       Count(*) 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
~~1~~	~~AI Chatbot~~	~~1~~
2	Website	1

```
8. SELECT  cust_id,
           order_total,
           return_total,
     CASE
       WHEN return_total = 20   THEN 'No Return'
       WHEN return_total = 100  THEN 'Returned'
     END as return_status
   FROM orders ASA
   FULL OUTER JOIN returns AS B
   ON A. cust_id = B. cust_id
   WHERE. order_total > 100;
```

cust_id	order_total	return_total	return_states
11	120	20	No Return
12	250	NULL	NULL
13	180	NULL	NULL
14	NULL	100	~~Returned~~ NULL

```
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
   ORDER BY login_count DESC;
```

user_id	name	login_count
2	Glorner	2
3	Steve	1
1	Nelson	0 ~~Null~~

```
10. SELECT A.teacher_id,
                teacher_nane,
                subject_name,
      CASE
        WHEN subject_name NOT IN ('Math', 'Science', 'History')
        THEN 'No Subject Assigned'
      End ~~as~~ AS subject_name
   FROM teachers AS A
   LEFT JOIN subjects AS B
   ON A.teacher_id = B.teacher_id
   ORDER BY teacher_name ASC;
```

teacher_id	teacher_name	subject_name
3	Mr. Dlamini	Not Subject Assigned
1	Mr Hlongwane	Math
1	Mr Hlongwane	Science
2	Ms Ndlabo	No Subject Assigned

Exercise 4 corrections Corrections

④
```
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;
```

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;
```

```
6. SELECT  A. student_id,
                 name,
              days-present,
     CASE
        WHEN days-present >= 15 THEN ('Exeellent)
        WHEN days-present BETWEEN 6 AND 14 THEN 'Needs Impraveme
        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. studet_id ;
```

```
8. SELECT  COALESCE (A. cust_id, B. cust_id) AS cust_id,
              order total,
              return_total,
     CASE
           WHEN return_total IS NOT 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 ;
```

```
9. SELECT A. user_id, name, COUNT (login-date) AS login_cou
  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;
```