Name: PAUL BENEDICT G. UNIGO

1. Display all columns from tbl_employees.
2. Display only the firstname and lastname of all employees.
3. Show firstname, lastname, and salary of all employees.
4. Find all employees whose firstname starts with **'S'**.
5. Find all employees whose lastname ends with **'off'**.
6. Find employees with firstname containing **'an'**.
7. Find employees whose firstname second letter is **'e'**.
8. Find employees whose lastname starts with **'R'**.
9. Show distinct position_id values.
10. Show distinct gender values from the table.
11. Display all employees with a salary greater than **60,000**.
12. Display all employees who were hired before **2015-01-01**.
13. Display employees with gender = 'F'.
14. Show employees whose status is ACTIVE.
15. Display employees whose salary is between **50,000** and **70,000**.
16. Display employees sorted by firstname in ascending order.
17. Display employees sorted by salary in descending order.
18. Show employees sorted by date_hired (oldest first).
19. Count how many employees are in each position_id.
20. Count how many employees are grouped by gender.
21. Find the total salary per position_id.

```
MariaDB [db_unigo]> SELECT position_id, SUM(salary) AS total_salary
    -> FROM tbl_employees
    -> GROUP BY position_id
    -> ;
+-------------+--------------+
| position_id | total_salary |
+-------------+--------------+
|           1 |    118000.00 |
|           2 |    167000.00 |
|           3 |     48000.00 |
|           4 |     70000.00 |
|           5 |     52000.00 |
+-------------+--------------+
5 rows in set (0.001 sec)
```

22. Show position_id groups having more than **1 employee**.

```
MariaDB [db_unigo]> SELECT position_id, COUNT(*) AS employee_count
    -> FROM tbl_employees
    -> GROUP BY position_id
    -> HAVING COUNT(*) > 1;
+-------------+----------------+
| position_id | employee_count |
+-------------+----------------+
|           1 |              2 |
|           2 |              2 |
+-------------+----------------+
2 rows in set (0.001 sec)
```

23. Show gender groups where the average salary is above **60,000**.

```
MariaDB [db_unigo]> SELECT gender, AVG(salary) AS avg_salary
    -> FROM tbl_employees
    -> GROUP BY gender
    -> HAVING AVG(salary) > 60000;
+--------+--------------+
| gender | avg_salary   |
+--------+--------------+
| M      | 67400.000000 |
+--------+--------------+
1 row in set (0.001 sec)
```

24. Show only the **first 3 employees** from the table.

```
MariaDB [db_unigo]> SELECT * FROM tbl_employees
    -> ORDER BY id ASC
    -> LIMIT 3;
+----+-----------+----------+-------------+--------+-----------+------------+--------+
| id | firstname | lastname | position_id | gender | salary    | date_hired | status |
+----+-----------+----------+-------------+--------+-----------+------------+--------+
|  1 | Jerwin    | Cruz     |           1 | M      |  60000.00 | 2018-06-30 | ACTIVE |
|  2 | Peter     | Parker   |           2 | M      |  65000.00 | 2011-12-02 | ACTIVE |
|  3 | Tony      | Stark    |           2 | M      | 102000.00 | 2002-02-01 | ACTIVE |
+----+-----------+----------+-------------+--------+-----------+------------+--------+
3 rows in set (0.000 sec)
```

25. Show **3 employees starting from the 3rd record** in the table.

```
MariaDB [db_unigo]> SELECT * FROM tbl_employees
    -> ORDER BY id ASC
    -> LIMIT 2, 3; --equivalent to: LIMIT 3 OFFSET 2
+----+-----------+----------+-------------+--------+-----------+------------+--------+
| id | firstname | lastname | position_id | gender | salary    | date_hired | status |
+----+-----------+----------+-------------+--------+-----------+------------+--------+
|  3 | Tony      | Stark    |           2 | M      | 102000.00 | 2002-02-01 | ACTIVE |
|  4 | Natasha   | Romanoff |           4 | F      |  70000.00 | 2015-10-24 | ACTIVE |
|  5 | Wanda     | Maximoff |           3 | F      |  48000.00 | 2016-09-25 | ACTIVE |
+----+-----------+----------+-------------+--------+-----------+------------+--------+
3 rows in set (0.000 sec)
```