LEETCODE SQL PLAN (10 DAYS)—DAY 2

1873. Calculate Special Bonus

QNSLINK:

https://leetcode.com/problems/calculate-special-bonus/?envType=study-plan&id=sql-i

Solution:

APPROACH-1

```
SELECT employee_id,
  CASE
  WHEN employee_id%2=1 AND name NOT LIKE 'M%' THEN salary
  ELSE 0
  END
AS bonus FROM Employees
ORDER BY employee_id;
```

Explanation: Here they are asked to find the employee to calculate the bonus and order the records by employee_id:

Bonus is 100% of salary: If employee id is odd number and the employee's name does not start with the character 'M'.

Bonus is 0: If the employee doesn't meet the above conditions, then the bonus is 0

So according the problem we can use the CASE it is same like if-else that we use in the programming.

- **So, to find whether the employee_id is odd or not, I used the condition that employee_id%2==1.We know this is basic condition to check whether the number is odd or even if the mod value is 1 then it can be given as odd if not even.**
- **To check whether the name starts with 'M' we have a topic named 'LIKE FUNCTIONS' in SQL so to check that the name starts with we have use this syntax: LIKE 'character that you need to check%'→ in our case 'M%'. So, we have to give condition, that name of the employee shouldn't start with 'M' for this we have to use 'NOT' operator (NOT is a logical operator in SQL that you can put before any conditional statement to select rows for which that statement is false.) so we used NOT in the query.**
- **If the above conditions don't satisfy then 0, so we kept 0 in the else part.**
- **And at last, they asked to order the records by employee_id, so we used order by which helps in ordering the elements.**

APPROACH-2

```
SELECT employee_id,
IF (employee_id%2=1 AND name not like "M%", salary, 0) as bonus
FROM Employees order by employee_id;
```

Explanation: It is similar to the above code and here we are using if condition.

This is the Syntax for the IF Condition in SQL we can understand better by seeing the syntax:

IF(CONDITION, VALUE_IF_TRUE, VALUE_IF_FALSE)

In our code,

Condition: **employee_id%2 AND name not like "M%"**

Value_if_true: **salary (if the above condition is true bonus is 100% of the employee salary).**

Value_if_false: **0 (if the employee doesn't satisfy the condition the bonus is considers as 0) and at last we are naming the column name as bonus by using as (alias in SQL--> as).**

Then we are using order by to order the table by employee_id (in default by using order by we get in ascending order).

This is about the Calculate Special Bonus.

627. Swap Salary

QNSLINK:

https://leetcode.com/problems/swap-salary/?envType=study-plan&id=sql-i

Solution:

Approach 1:

```
UPDATE salary set sex = IF(sex="m" , "f" , "m");
```

Explanation:

Here we are asked to write a query to swap all 'f' and 'm' values (i.e., change all 'f' values to 'm' and vice versa) with a single update statement and no intermediate temporary tables.

So, as we know to update the column values, we have to use UPDATE STATEMENT.

Syntax:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

It is not necessary to have only where condition it can be any condition or clause to give our condition.

- So, we have used if to give the condition: if sex column value is 'm' then it should be update as 'f 'if the value is 'f' then 'm'.
- We have this syntax: IF (CONDITION, VALUE_IF_TRUE, VALUE_IF_FALSE)

CONDITION: 'm' , VALUE_IF_TRUE: 'f ' VALUE_IF_FALSE: 'm'

Approach 2:

```
UPDATE Salary
SET
  sex = CASE sex
    WHEN 'm' THEN 'f'
    ELSE 'm'
  END;
```

I hope the above approach is pretty clear we have used UPDATE statement and then CASE STATEMENT.

 # Knowledge on CASE Statement.

The CASE expression goes through conditions and returns a value when the first condition is met (like an if-then-else statement). So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause.

Syntax:

```
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END;
```

This is about the Swap Salary.

196. Delete Duplicate Emails

QNSLINK:

https://leetcode.com/problems/delete-duplicate-emails/?envType=study-plan&id=sql-i

Solution:

Approach 1:

```
DELETE p2 FROM Person p1 JOIN Person p2 ON p1.Email = p2.Email AND p1.Id < p2.Id;
```

Explanation:

Here we are asked to write a query to delete all the duplicate emails, keeping only one unique email with the smallest id. Note that you are supposed to write a DELETE statement and not a SELECT one.

We are using the self-join to join the table itself.

Syntax:

```
SELECT column name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

As mentioned, we have to use only the delete statement then we wrote the code by starting with the delete statement. After joining the table, we get the table like this:

```
10 •   Select * from  Person p1 JOIN Person p2 ON p1.Email = p2.Email ;
11
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	email	id	email
3	john@example.com	1	john@example.com
1	john@example.com	1	john@example.com
2	bob.com	2	bob.com
3	john@example.com	3	john@example.com
1	john@example.com	3	john@example.com

You will get by self-join of table.

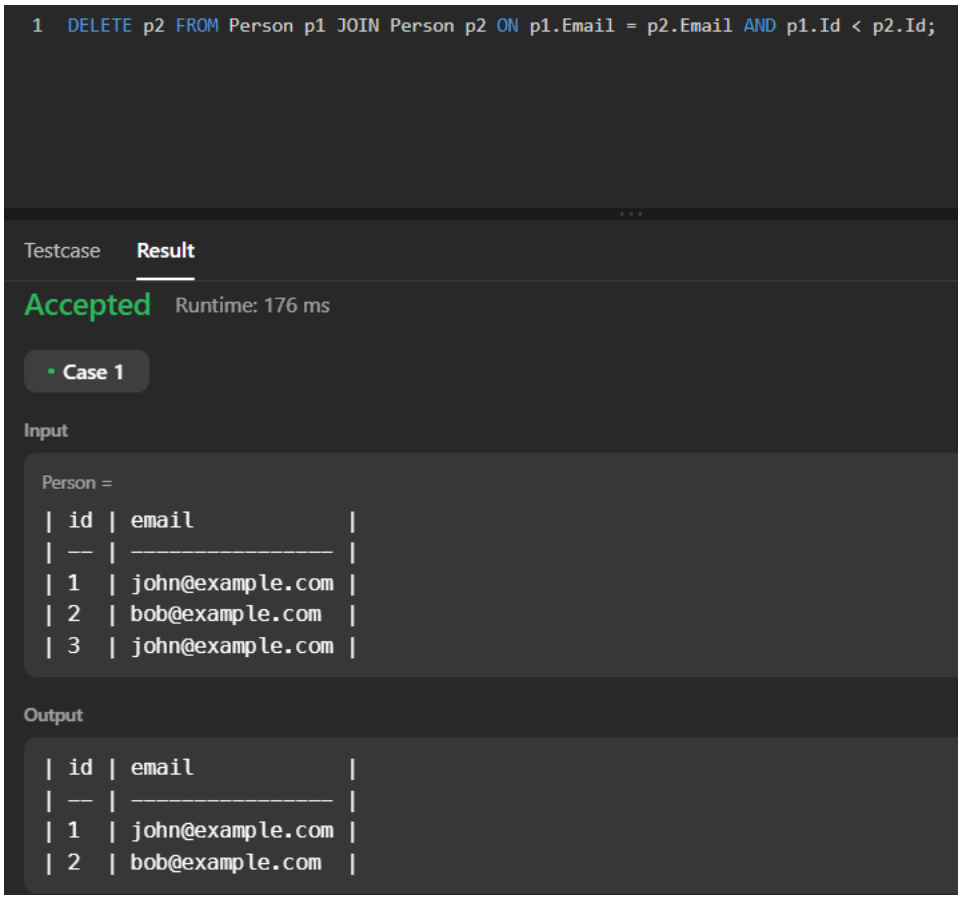
If you add this condition p1.Id < p2.Id , you will get the below output.

```
10 •   Select * from  Person p1 JOIN Person p2 ON p1.Email = p2.Email AND p1.Id < p2.Id;
11
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	email	id	email
1	john@example.com	3	john@example.com

So, at last if you add the delete in the starting you will get the output where duplicate record is deleted.



This is about Delete duplicate emails.