LEETCODE SQL 10-DAYS PLAN

595. Big Countries

QNS LINK: https://leetcode.com/problems/big-countries/?envType=study-plan&id=sql-i

Solution: USING OR

```
select name, population, area from World where area >= 3000000 OR population >= 25000000;
```

Explanation:

In the question they are asked to retrieve the table where the area should be at least three million (i.e., 3000000 km2), or it has a population of at least twenty-five million (i.e., 25000000).

So, we have to use here where condition to filter the records based on the condition that is required.

And they are asked to satisfy the either of the condition, so we are using OR operator.

Solution: USING UNION

```
SELECT name, population, area FROM World WHERE area > 3000000

UNION

SELECT name, population, area FROM World WHERE population > 25000000
```

In some cases, union is faster than OR.

The reason is that using OR in a query will often cause the Query Optimizer to abandon use of index seeks and revert to scans.

1757. Recyclable and Low-Fat Products

QNS LINK: https://leetcode.com/problems/recyclable-and-low-fat-products/?envType=study-plan&id=sql-i

Solution: USING AND

```
select product_id from Products where low_fats = "Y" AND   recyclable = "Y";
```

Explanation:

In the question they are asked to retrieve the table where the ids of products that are both low fat and recyclable. So we need to consider both conditions, for that we have to use the ‘AND’ operator so it helps us to find the records which column satisfy the both conditions.

584. Find Customer Referee

QNS LINK: https://leetcode.com/problems/find-customer-referee/?envType=study-plan&id=sql-i

Solution: **USING <> and IS NULL**

```
SELECT name FROM customer WHERE referee_id <> 2 OR referee_id IS NULL;
```

Explanation:

They are asked to retrieve the table where referee_id should not be equal to 2 and here we have given the condition referee_id is null because when we use <> or != operator in a query, it ignores any values that are NULL and Null is considered as Unknown and undefined value. Since it is unknown (it does consider the possibility that the value can also be 2). So, it does not fetch the records with null values. because of this we need to include a NULL Condition**.**

Solution: **USING coalesce**

```
select name from customer where coalesce(referee_id, 0) != 2
```

Explanation:

The Coalesce function is used to handle the Null values. The null values are replaced with user-defined values during the expression evaluation process. This function evaluates arguments in a particular order from the provided arguments list and always returns the first non-null value from the series of expressions. The result of the COALESCE function returns NULL only if all the arguments are null.

SYNTAX: COALESCE (expression [1...n])

Solution: **IFNULL**

```
SELECT name FROM customer WHERE IFNULL (referee_id,0) <> 2;
```

Explanation:

The IFNULL () function returns a specified value if the expression is NULL. If the expression is NOT NULL, this function returns the expression.

Syntax: IFNULL (expression, alt_value)

Expression: Required. The expression to test whether is NULL

alt_value: Required. The value to return if expression is NULL

IFNULL (referee_id, 0) means if referee_id is null then take it as 0.

<> 2 means not equals 2. So, it will give us the name of the customers not having referee_id value 2.

183. Customers Who Never Order

QNSLINK: https://leetcode.com/problems/customers-who-never-order/?envType=study-plan&id=sql-i

Solution:

```
select c.Name AS Customers from Customers c LEFT JOIN Orders o ON (o.CustomerId = c.Id)
where o.ID IS NULL
```

Explanation:

We have to report all customers who never order anything.

So, we can use left join to combine both the tables (The `LEFT JOIN` keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match). And by using the where condition of order id is null from the orders table.

Solution:

```
SELECT A.Name from Customers A WHERE A.Id NOT IN (SELECT B.CustomerId from Orders B)
```

Explanation:

We can use NOT IN operator (can only replace the <> or != operators) .