1667. Fix Names in a Table

QNS LINK:

https://leetcode.com/problems/fix-names-in-a-table/description/?envType=study-plan&id=sql-i

Solution:

```
select user_id , concat(UPPER (SUBSTR(name , 1, 1)), LOWER(Substr(name,2))) as name  from
Users Order by user_id ASC ;
```

Explanation:

Here they are asked to SQL query to fix the names so that only the first character is uppercase and the rest are lowercase and to return the result table ordered by user_id.

- So, we don't have any statement or operator to directly work with that scenario. We have two concepts in SQL named UPPER AND LOWER functions. With the help of this functions, we can try to work with the scenario.
- The SUBSTR () function extracts a substring from a string (starting at any position).
- So UPPER (SUBSTR (name ,1,1)) → So according to our use case we have to convert our 1st letter into the 'upper case ' , so we used the substr concept so that it helps us in extracting the first letter from the name.
- LOWER (SUBSTR (name ,2))→Same as the above case it helps us in retrieving the string from the 2nd character and till the end as we haven't mentioned any limit.
- After doing this upper and lower we have to combine the string using concat () function.
- CONCAT(string1, string2, ..., string_n).
- And then we have to order the record by the user_id so we used order by user_id , ASC helps us in arranging the values in ascending order.

1484. Group Sold Products By The Date

QNS LINK:

https://leetcode.com/problems/group-sold-products-by-the-date/?envType=study-plan&id=sql-i

Solution:

```
SELECT sell_date,
        COUNT(DISTINCT(product)) AS num_sold,
        GROUP_CONCAT(DISTINCT product ORDER BY product ASC SEPARATOR ',') AS products
FROM Activities
GROUP BY sell_date
ORDER BY sell_date ASC
```

Explanation:

Here we are asked to write an SQL query to find for each date the number of different products sold and their names. The sold products names for each date should be sorted lexicographically. Return the result table ordered by sell date.

So will divide our problem into steps:

1. We have to find the count of products as name them as num_sold. So as mentioned in the question there is no primary key in the table so table may consist of duplicates then we have to use 'distinct' keyword to fetch the distinct records, and to count the records we have to use 'count' keyword and then name that column name as num_sold using 'alias'.

   For our step 1 query will be: COUNT (DISTINCT (product)) AS num_sold;

2. Then we have to sort the product names in the lexicographical way (meaning in the dictionary order) and the we have to group the products by their sell_date.
So we are using group_cancat to concat or add all the values in a comma-seperated manner and then we have order them in dictionary order so we should use order by and fetch them column name as products.

For our step 2 query will be: GROUP_CONCAT(DISTINCT product ORDER BY product ASC SEPARATOR ',') AS products

Knowledge on group_concat ()

The GROUP_CONCAT () function in MySQL is used to concatenate data from multiple rows into one field. This is an aggregate (GROUP BY) function which returns a String value, if the group contains at least one non-NULL value. Otherwise, it returns NULL.

3. At last, we have to group by sales date and then order by the sales date.

For our step 3 query will be:
GROUP BY sell_date
ORDER BY sell_date ASC

This is about the Group Sold Products by The Date.

1527. Patients With a Condition

QNS LINK:

https://leetcode.com/problems/patients-with-a-condition/?envType=study-plan&id=sql-i

Solution:

```
SELECT * FROM PATIENTS WHERE
conditions LIKE '% DIAB1%' OR
conditions LIKE 'DIAB1%';
```

Explanation:

Here we are asked to write an SQL query to report the patient_id, patient_name all conditions of patients who have Type I Diabetes. Type I Diabetes always starts with DIAB1 prefix.Return the result table in any order.

So, for the above use case we have to use the 'LIKE' operator to find the condition starting with DIAB1 for this we have to use LIKE 'DIAB1%'→it helps us in fetching all the records starting with DIAB1.And we have to fetch records whether there is string 'DIAB1' which is in between (this test case is not mentioned in the question but internally they have considered this).To find that whether DIAB1 is in between we have to use % DIAB% it helps us to fetch the records which consists of DIAB1 in between.

Knowledge on LIKE OPERATORS.

LIKE OPERATOR	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position

For more info on like operators visit : W3SCHOOLS(LIKE OPERATORS)