SQL PROBLEM CHALLENGE

DAY 8

- Problem Statement
- Input
- Expected Output
- Solution 1
- Solution 2
- Logic - Recursive CTE

PROBLEM STATEMENT

There are rows with missing JOB_ROLE values. Write a query to fill in those blank fields with appropriate values.
Assume row_id is always in sequence and job_role field is populated only for the first skill.

For Example: In row_id =2, we have a NULL value for job_role so our task is to fill it with the NON-NULL value that occurred before this row_id. In this case, fill it with "Data Engineer"

Similary, in row_id = 7, fill it with Web Developer and so on.

INPUT

INPUT		
ROW_ID	JOB_ROLE	SKILLS
1	Data Engineer	SQL
2		Python
3		AWS
4		Snowflake
5		Apache Spark
6	Web Developer	Java
7		HTML
8		CSS
9	Data Scientist	Python
10		Machine Learning
11		Deep Learning
12		Tableau

EXPECTED OUTPUT

OUTPUT		
ROW_ID	JOB_ROLE	SKILLS
1	Data Engineer	SQL
2	Data Engineer	Python
3	Data Engineer	AWS
4	Data Engineer	Snowflake
5	Data Engineer	Apache Spark
6	Web Developer	Java
7	Web Developer	HTML
8	Web Developer	CSS
9	Data Scientist	Python
10	Data Scientist	Machine Learning
11	Data Scientist	Deep Learning
12	Data Scientist	Tableau

SOLUTION 1

```sql
WITH flag_cte AS
    (
        SELECT *,
        SUM(CASE WHEN job_role Is not null THEN 1 ELSE 0 END)
            OVER (ORDER BY row_id ASC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) as segment
        FROM job_skills
)
SELECT row_id, MAX(job_role) OVER(PARTITION BY segment) AS job_role_n, skills
FROM flag_cte
```

SOLUTION 2

```sql
WITH recursive_cte AS
    (
        SELECT row_id,job_role,skills
        FROM job_skills
        WHERE row_id = 1

        UNION ALL

        SELECT js.row_id, COALESCE(js.job_role,rc.job_role) AS job_role, js.skills
        FROM recursive_cte AS rc
        JOIN job_skills AS js ON js.row_id = rc.row_id+1
    )
SELECT *
FROM recursive_cte
```

LOGIC - RECURSIVE CTE

```
Sample structure

WITH cte AS
    (
        Base Query
        UNION ALL
        Recursive Query with termination condition
    )
SELECT *
FROM cte;
```

We are using recursive CTE to retrieve data from the job_skills table until the given condition.

This is the base Query for our Recursive CTE that selects data for the first row where row_id is equal to 1.

```
SELECT row_id, job_role, skills
FROM job_skills
WHERE row_id = 1
```

LOGIC - RECURSIVE CTE

```
SELECT js.row_id, COALESCE(js.job_role, rc.job_role) AS job_role, js.skills
FROM recursive_cte AS rc
JOIN job_skills AS js ON js.row_id = rc.row_id + 1
```

In Recursive Query:

- UNION ALL is used to combine the results of the anchor query and the recursive query.
- COALESCE(js.job_role, rc.job_role) chooses the job role from the current row if it's not null, otherwise, it chooses the job role from the previous row (rc.job_role).
- Lastly, JOIN condition js.row_id = rc.row_id + 1 ensures that the next row is joined to the previous row based on the row_id. This will also be our TERMINATING condition for the recursion.

THANK YOU