

Query 1

Find Duplicate Records

Problem: Identify duplicate rows in a table

Query

```
SELECT column_name, COUNT(*)  
FROM table_name  
GROUP BY column_name  
HAVING COUNT(*) > 1;
```

Use Case : Find repeated orders, duplicate entries, or redundant data.

Query 2

Top N Records per Group

Problem: Retrieve the top N records for each category.

Query

```
SELECT *
FROM (
    SELECT column1, column2,
           RANK() OVER (PARTITION BY category_column ORDER BY value_column DESC) AS rank
    FROM table_name
) ranked
WHERE rank <= N;
```

Use Case : List top-performing products or highest revenue regions

Query 3

Detect Missing Data

Problem: Find gaps in sequential data.

Query

```
SELECT t1.id + 1 AS missing_id  
FROM table_name t1  
LEFT JOIN table_name t2 ON t1.id + 1 = t2.id  
WHERE t2.id IS NULL;
```

Use Case : Identify missing invoice numbers or skipped entries in a series