SQL TIPS AND TRICKS

PART 7

Rank on Duplicates

MAYURI DANDEKAR

1. **(CTEs):**

`cte_duplicate`: This CTE selects the `manager_id` from the `emp` table and groups them by `manager_id`. It filters out groups where the count of records for a particular `manager_id` is greater than 1, essentially identifying duplicate `manager_id` entries in the `emp` table.

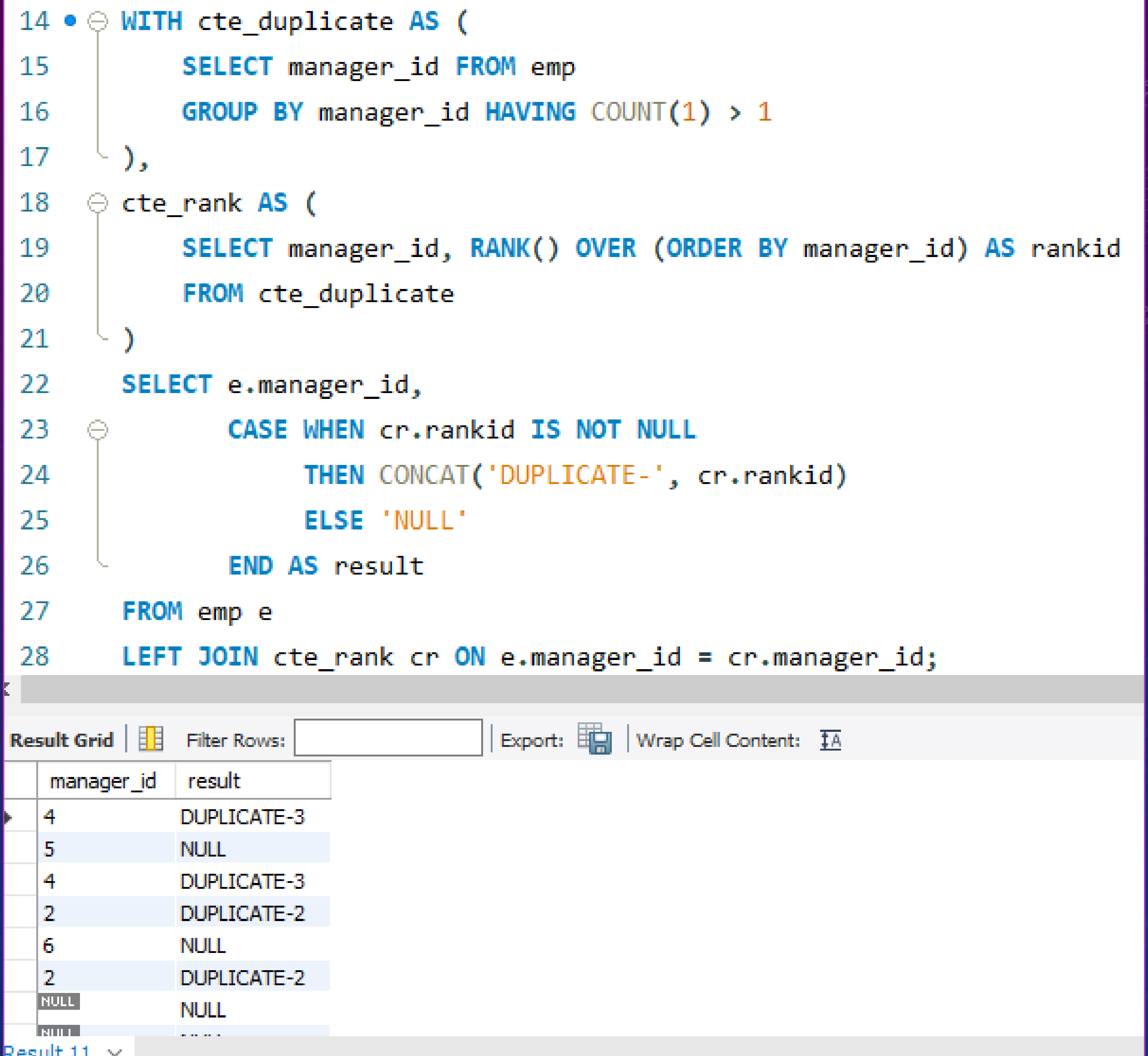
`cte_rank`: This CTE takes the result of `cte_duplicate` and assigns a rank to each `manager_id` using the `RANK()` window function. The `RANK()` function is used to assign a unique rank to each duplicate `manager_id` based on the order of `manager_id`.

2. **Main Query:**

The main query selects columns `e.manager_id` and a derived column `result`.

For `result`, it uses a `CASE` statement to determine the value based on whether the `manager_id` is found in the `cte_rank` CTE or not. If it's found (`cr.rankid` is not null), it concatenates the string 'DUPLICATE' with the rank id (`cr.rankid`). If it's not found, it returns 'NULL'.

It performs a `LEFT JOIN` between the `emp` table (`e`) and the `cte_rank` CTE (`cr`) on `manager_id`.



```sql
WITH cte_duplicate AS (
    SELECT manager_id FROM emp
    GROUP BY manager_id HAVING COUNT(1) > 1
),
cte_rank AS (
    SELECT manager_id, RANK() OVER (ORDER BY manager_id) AS rankid
    FROM cte_duplicate
)
SELECT e.manager_id,
       CASE WHEN cr.rankid IS NOT NULL
            THEN CONCAT('DUPLICATE-', cr.rankid)
            ELSE 'NULL'
       END AS result
FROM emp e
LEFT JOIN cte_rank cr ON e.manager_id = cr.manager_id;
```

manager_id	result
4	DUPLICATE-3
5	NULL
4	DUPLICATE-3
2	DUPLICATE-2
6	NULL
2	DUPLICATE-2
NULL	NULL
NULL	NULL

THANK YOU

MAYURI DANDEKAR