

# SQL TIPS AND TRICKS

PART 34

SQL convert Rows to Columns  
and Columns to Rows.

MAYURI DANDEKAR





## Sample Example

|   | emp_id | salary_compensation_type | val   |
|---|--------|--------------------------|-------|
| ▶ | 1      | salary                   | 10000 |
|   | 1      | bonus                    | 5000  |
|   | 1      | hike_percent             | 10    |
|   | 2      | salary                   | 15000 |
|   | 2      | bonus                    | 7000  |
|   | 2      | hike_percent             | 6     |
|   | 3      | salary                   | 12000 |
|   | 3      | bonus                    | 6000  |
|   | 3      | hike_percent             | 7     |

# ROWS TO COLUMN

STEP 1 - convert rows content to column as header using CASE WHEN.

```
21 • select emp_id,  
22      case when salary_compensation_type = "salary" then val end as salary,  
23      case when salary_compensation_type = "bonus" then val end as bonus,  
24      case when salary_compensation_type = "hike_percent" then val end as hike_percent  
25 from emp_compensation;
```





<   Filter Rows:  | Export:  | Wrap Cell Content: 

|   | emp_id | salary | bonus | hike_percent |
|---|--------|--------|-------|--------------|
| ▶ | 1      | 10000  | NULL  | NULL         |
|   | 1      | NULL   | 5000  | NULL         |
|   | 1      | NULL   | NULL  | 10           |
|   | 2      | 15000  | NULL  | NULL         |
|   | 2      | NULL   | 7000  | NULL         |
|   | 2      | NULL   | NULL  | 6            |
|   | 3      | 12000  | NULL  | NULL         |
|   | 3      | NULL   | 6000  | NULL         |
|   | 3      | NULL   | NULL  | 7            |

## ROWS TO COLUMN

**STEP 2 - using GROUP BY, group all the emp\_id and SUM all the values from each group to get one row for each emp\_id and add temp table.**

```
21 • CREATE TABLE emp_compensation_pivot AS
22 SELECT emp_id,
23         SUM(CASE WHEN salary_component_type = 'salary' THEN val ELSE 0 END) AS salary,
24         SUM(CASE WHEN salary_component_type = 'bonus' THEN val ELSE 0 END) AS bonus,
25         SUM(CASE WHEN salary_component_type = 'hike_percent' THEN val ELSE 0 END) AS hike_percent
26 FROM emp_compensation
27 GROUP BY emp_id;
28 • SELECT * FROM emp_compensation_pivot;
```

<   Filter Rows:  | Export:  | Wrap Cell Content: 

|   | emp_id | salary | bonus | hike_percent |
|---|--------|--------|-------|--------------|
| ▶ | 1      | 10000  | 5000  | 10           |
|   | 2      | 15000  | 7000  | 6            |
|   | 3      | 12000  | 6000  | 7            |

# COLUMN TO ROW

STEP 3 - using UNION ALL, combine all columns to convert in rows.

Finally drop the temp table

```
31 • SELECT emp_id, 'salary' AS salary_component_type, salary AS val FROM emp_compensation_pivot
32 UNION ALL
33 SELECT emp_id, 'bonus' AS salary_component_type, bonus AS val FROM emp_compensation_pivot
34 UNION ALL
35 SELECT emp_id, 'hike_percent' AS salary_component_type, hike_percent AS val FROM emp_compensation_pivot
36 ORDER BY emp_id, salary_component_type;
37
38 • DROP TABLE emp_compensation_pivot;
```

| Result Grid | Filter Rows:          | Export: | Wrap Cell Content: |
|-------------|-----------------------|---------|--------------------|
| emp_id      | salary_component_type | val     |                    |
| 1           | bonus                 | 5000    |                    |
| 1           | hike_percent          | 10      |                    |
| 1           | salary                | 10000   |                    |
| 2           | bonus                 | 7000    |                    |
| 2           | hike_percent          | 6       |                    |
| 2           | salary                | 15000   |                    |
| 3           | bonus                 | 6000    |                    |
| 3           | hike_percent          | 7       |                    |
| 3           | salary                | 12000   |                    |





# THANK YOU

MAYURI DANDEKAR