## 1. City-wise Department Performance

**Goal:** Department in each city with the highest total salary payout.

```
WITH dept_salary AS (
    SELECT
        city,
        Department,
        SUM(Salary) AS total_salary
    FROM employees
    GROUP BY city, Department
),
ranked_dept AS (
    SELECT
        city,
        Department,
        total_salary,
        ROW_NUMBER() OVER (PARTITION BY city ORDER BY total_salary
DESC) AS dept_rank
    FROM dept_salary
)
SELECT
    city,
    Department,
    total_salary
FROM ranked_dept
WHERE dept_rank = 1;
```

---

## 2. Salary Distribution & Ranking

**Goal:** Rank employees by salary within each department and city.

```
WITH dept_rank AS (
    SELECT
        *,
        RANK() OVER (PARTITION BY Department ORDER BY Salary DESC)
AS dept_salary_rank
    FROM employees
),
city_rank AS (
```

```
    SELECT
        *,
        RANK() OVER (PARTITION BY city ORDER BY Salary DESC) AS
city_salary_rank
    FROM dept_rank
)
SELECT
    SI_No,
    Emp_Name,
    Department,
    city,
    Salary,
    dept_salary_rank,
    city_salary_rank
FROM city_rank
ORDER BY Department, city, Salary DESC;
```

---

## 3. Salary Tiers and Department Ranking

**Goal:** Categorize employees into salary tiers and show top 2 employees per department.

```
WITH salary_tiers AS (
    SELECT
        *,
        CASE
            WHEN Salary < 30000 THEN 'Low'
            WHEN Salary BETWEEN 30000 AND 70000 THEN 'Medium'
            ELSE 'High'
        END AS salary_tier
    FROM employees
),
dept_rank AS (
    SELECT
        *,
        RANK() OVER (PARTITION BY Department ORDER BY Salary DESC)
AS dept_rank
    FROM salary_tiers
)
SELECT
    SI_No,
    Emp_Name,
```

```
        Department,
        Salary,
        salary_tier
FROM dept_rank
WHERE dept_rank <= 2
ORDER BY Department, dept_rank;
```

---

## 4. Age & Salary Bracket Ranking

**Goal:** Classify employees by age groups and show top 3 earners per group.

```
WITH age_groups AS (
    SELECT
        *,
        CASE
            WHEN Age < 30 THEN 'Young'
            WHEN Age BETWEEN 30 AND 50 THEN 'Mid'
            ELSE 'Senior'
        END AS Age_Group
    FROM employees
),
age_rank AS (
    SELECT
        *,
        RANK() OVER (PARTITION BY Age_Group ORDER BY Salary DESC) AS
age_salary_rank
    FROM age_groups
)
SELECT
    SI_No,
    Emp_Name,
    Age,
    Age_Group,
    Salary
FROM age_rank
WHERE age_salary_rank <= 3
ORDER BY Age_Group, age_salary_rank;
```