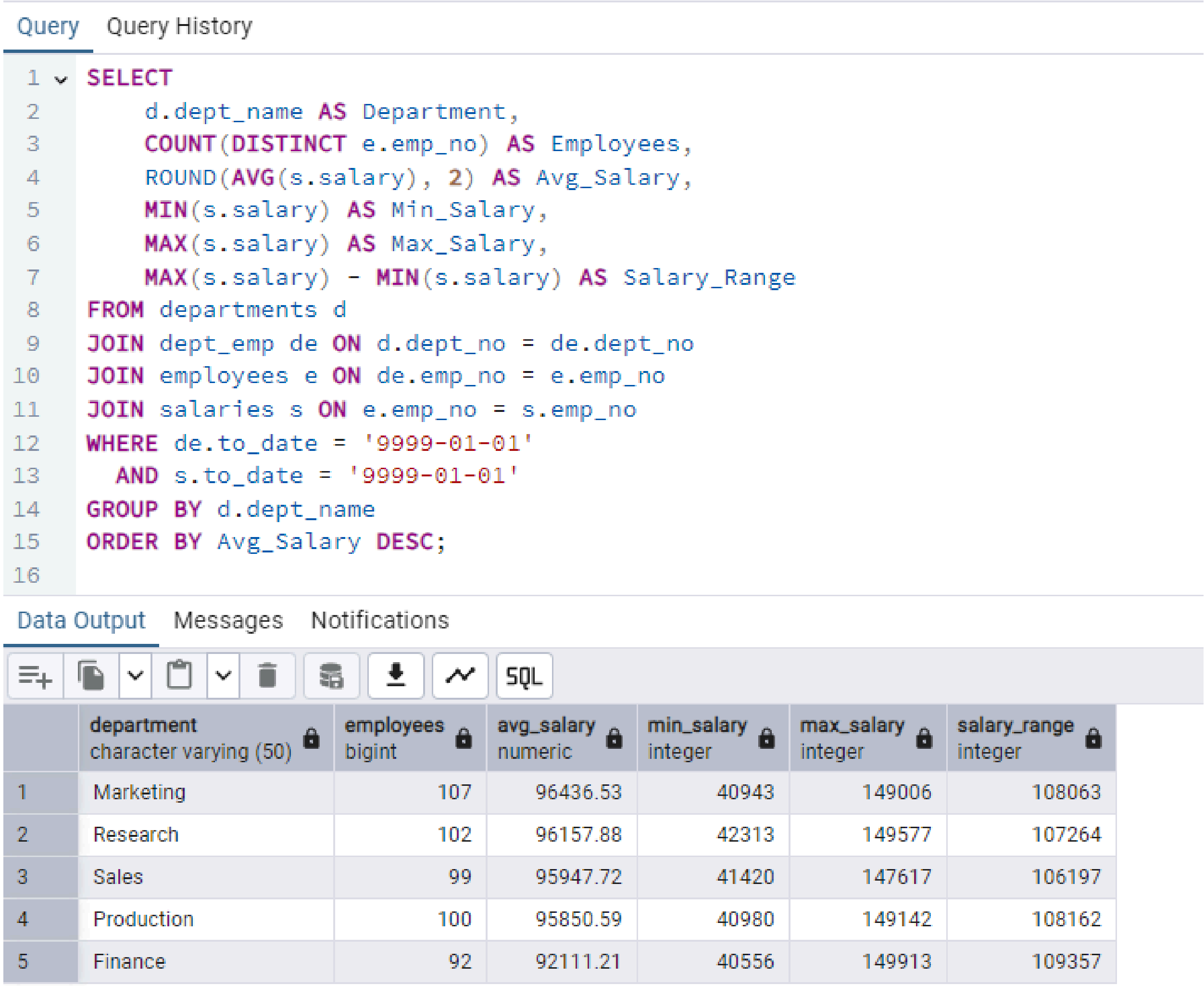
# TASK 2 :Data Aggregation & Reporting with Real Dataset (OUTPUTS)

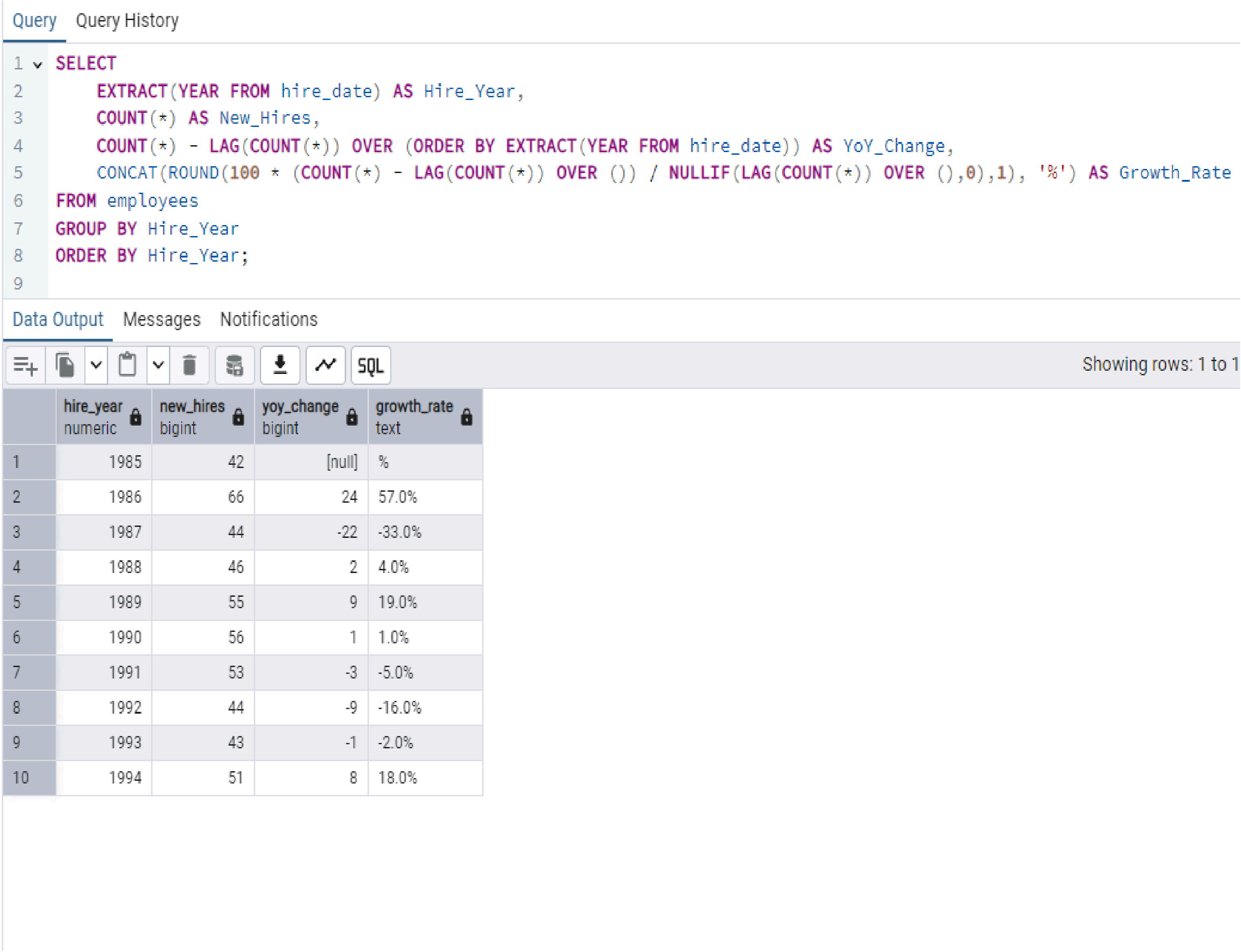
EXAMPLE 1 :



```sql
SELECT
    d.dept_name AS Department,
    COUNT(DISTINCT e.emp_no) AS Employees,
    ROUND(AVG(s.salary), 2) AS Avg_Salary,
    MIN(s.salary) AS Min_Salary,
    MAX(s.salary) AS Max_Salary,
    MAX(s.salary) - MIN(s.salary) AS Salary_Range
FROM departments d
JOIN dept_emp de ON d.dept_no = de.dept_no
JOIN employees e ON de.emp_no = e.emp_no
JOIN salaries s ON e.emp_no = s.emp_no
WHERE de.to_date = '9999-01-01'
  AND s.to_date = '9999-01-01'
GROUP BY d.dept_name
ORDER BY Avg_Salary DESC;
```

| | department<br>character varying (50) | employees<br>bigint | avg_salary<br>numeric | min_salary<br>integer | max_salary<br>integer | salary_range<br>integer |
|---|---|---|---|---|---|---|
| 1 | Marketing | 107 | 96436.53 | 40943 | 149006 | 108063 |
| 2 | Research | 102 | 96157.88 | 42313 | 149577 | 107264 |
| 3 | Sales | 99 | 95947.72 | 41420 | 147617 | 106197 |
| 4 | Production | 100 | 95850.59 | 40980 | 149142 | 108162 |
| 5 | Finance | 92 | 92111.21 | 40556 | 149913 | 109357 |

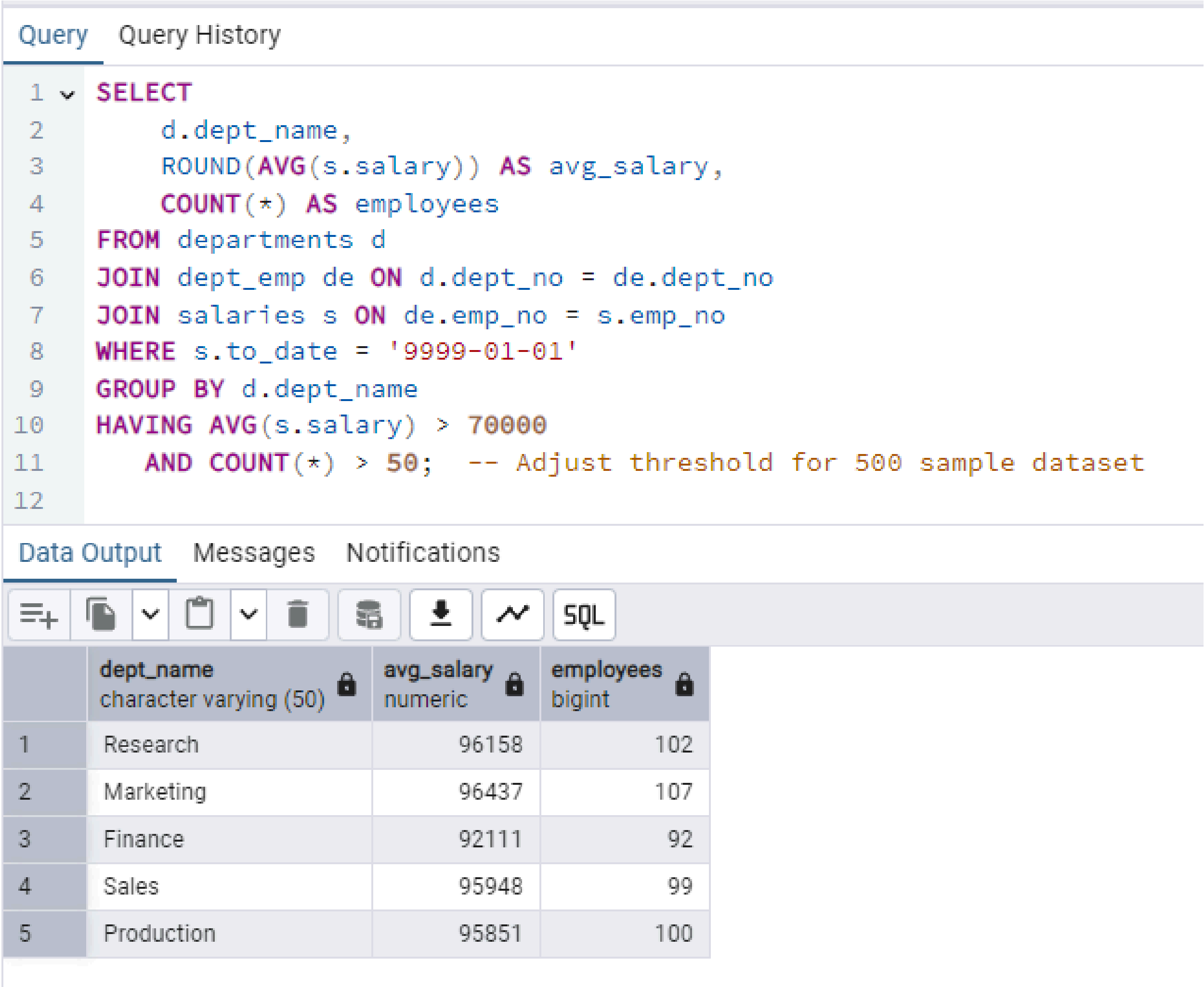
# EXAMPLE 2 :



```sql
SELECT
    EXTRACT(YEAR FROM hire_date) AS Hire_Year,
    COUNT(*) AS New_Hires,
    COUNT(*) - LAG(COUNT(*)) OVER (ORDER BY EXTRACT(YEAR FROM hire_date)) AS YoY_Change,
    CONCAT(ROUND(100 * (COUNT(*) - LAG(COUNT(*)) OVER ()) / NULLIF(LAG(COUNT(*)) OVER (),0),1), '%') AS Growth_Rate
FROM employees
GROUP BY Hire_Year
ORDER BY Hire_Year;
```

| | hire_year<br>numeric | new_hires<br>bigint | yoy_change<br>bigint | growth_rate<br>text |
|---|---|---|---|---|
| 1 | 1985 | 42 | [null] | % |
| 2 | 1986 | 66 | 24 | 57.0% |
| 3 | 1987 | 44 | -22 | -33.0% |
| 4 | 1988 | 46 | 2 | 4.0% |
| 5 | 1989 | 55 | 9 | 19.0% |
| 6 | 1990 | 56 | 1 | 1.0% |
| 7 | 1991 | 53 | -3 | -5.0% |
| 8 | 1992 | 44 | -9 | -16.0% |
| 9 | 1993 | 43 | -1 | -2.0% |
| 10 | 1994 | 51 | 8 | 18.0% |

EXAMPLE 3 :

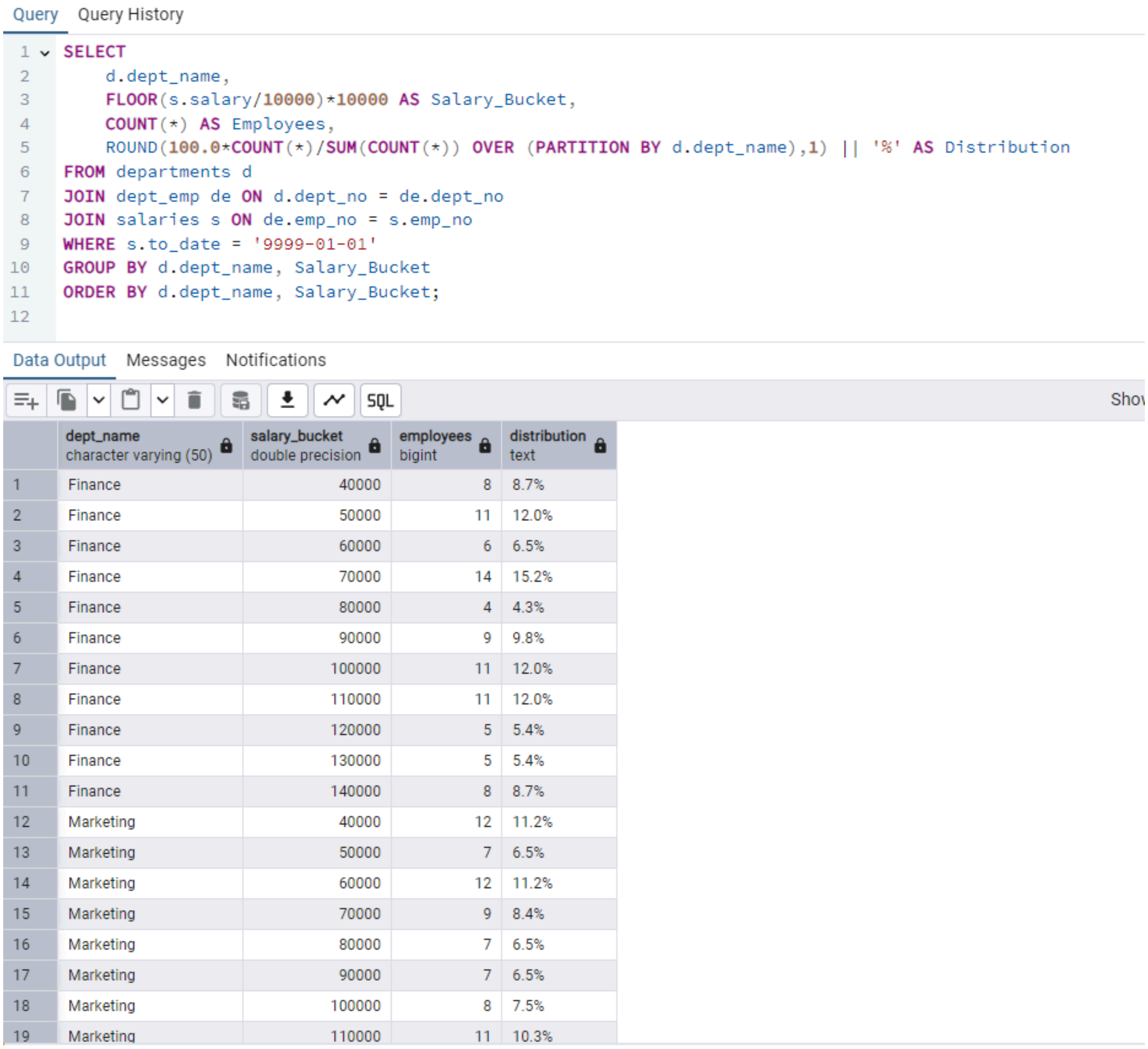


```sql
SELECT
    d.dept_name,
    ROUND(AVG(s.salary)) AS avg_salary,
    COUNT(*) AS employees
FROM departments d
JOIN dept_emp de ON d.dept_no = de.dept_no
JOIN salaries s ON de.emp_no = s.emp_no
WHERE s.to_date = '9999-01-01'
GROUP BY d.dept_name
HAVING AVG(s.salary) > 70000
   AND COUNT(*) > 50;  -- Adjust threshold for 500 sample dataset

```

| | dept_name<br>character varying (50) | avg_salary<br>numeric | employees<br>bigint |
|---|---|---|---|
| 1 | Research | 96158 | 102 |
| 2 | Marketing | 96437 | 107 |
| 3 | Finance | 92111 | 92 |
| 4 | Sales | 95948 | 99 |
| 5 | Production | 95851 | 100 |

EXAMPLE 4 :



```sql
SELECT
    d.dept_name,
    FLOOR(s.salary/10000)*10000 AS Salary_Bucket,
    COUNT(*) AS Employees,
    ROUND(100.0*COUNT(*)/SUM(COUNT(*)) OVER (PARTITION BY d.dept_name),1) || '%' AS Distribution
FROM departments d
JOIN dept_emp de ON d.dept_no = de.dept_no
JOIN salaries s ON de.emp_no = s.emp_no
WHERE s.to_date = '9999-01-01'
GROUP BY d.dept_name, Salary_Bucket
ORDER BY d.dept_name, Salary_Bucket;
```

| | dept_name<br>character varying (50) | salary_bucket<br>double precision | employees<br>bigint | distribution<br>text |
|---|---|---|---|---|
| 1 | Finance | 40000 | 8 | 8.7% |
| 2 | Finance | 50000 | 11 | 12.0% |
| 3 | Finance | 60000 | 6 | 6.5% |
| 4 | Finance | 70000 | 14 | 15.2% |
| 5 | Finance | 80000 | 4 | 4.3% |
| 6 | Finance | 90000 | 9 | 9.8% |
| 7 | Finance | 100000 | 11 | 12.0% |
| 8 | Finance | 110000 | 11 | 12.0% |
| 9 | Finance | 120000 | 5 | 5.4% |
| 10 | Finance | 130000 | 5 | 5.4% |
| 11 | Finance | 140000 | 8 | 8.7% |
| 12 | Marketing | 40000 | 12 | 11.2% |
| 13 | Marketing | 50000 | 7 | 6.5% |
| 14 | Marketing | 60000 | 12 | 11.2% |
| 15 | Marketing | 70000 | 9 | 8.4% |
| 16 | Marketing | 80000 | 7 | 6.5% |
| 17 | Marketing | 90000 | 7 | 6.5% |
| 18 | Marketing | 100000 | 8 | 7.5% |
| 19 | Marketing | 110000 | 11 | 10.3% |