

## TASK 2 OUTPUTS :

### EXAMPLE 1 :

Query    Query History

```
1  SELECT
2      d.dept_name AS Department,
3      COUNT(DISTINCT e.emp_no) AS Employees,
4      ROUND(AVG(s.salary), 2) AS Avg_Salary,
5      MIN(s.salary) AS Min_Salary,
6      MAX(s.salary) AS Max_Salary,
7      MAX(s.salary) - MIN(s.salary) AS Salary_Range
8  FROM departments d
9  JOIN dept_emp de ON d.dept_no = de.dept_no
10 JOIN employees e ON de.emp_no = e.emp_no
11 JOIN salaries s ON e.emp_no = s.emp_no
12 WHERE de.to_date = '9999-01-01'
13     AND s.to_date = '9999-01-01'
14 GROUP BY d.dept_name
15 ORDER BY Avg_Salary DESC;
```

Data Output    Messages    Notifications

≡+ 📁 ⏪ ⏴ 🗑️ 🔍 ↻ SQL

	department character varying (50)	employees bigint	avg_salary numeric	min_salary integer	max_salary integer	salary_range 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 :

Query Query History

```
1 ▾ SELECT
2     EXTRACT(YEAR FROM hire_date) AS Hire_Year,
3     COUNT(*) AS New_Hires,
4     COUNT(*) - LAG(COUNT(*)) OVER (ORDER BY EXTRACT(YEAR FROM hire_date)) AS YoY_Change,
5     CONCAT(ROUND(100 * (COUNT(*) - LAG(COUNT(*)) OVER ()) / NULLIF(LAG(COUNT(*)) OVER (),0),1), '%') AS Growth_Rate
6 FROM employees
7 GROUP BY Hire_Year
8 ORDER BY Hire_Year;
9
```

Data Output Messages Notifications

Showing rows: 1 to 1

	hire_year numeric	new_hires bigint	yo_y_change bigint	growth_rate 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 :

Query    Query History

```
1  SELECT
2      d.dept_name,
3      ROUND(AVG(s.salary)) AS avg_salary,
4      COUNT(*) AS employees
5  FROM departments d
6  JOIN dept_emp de ON d.dept_no = de.dept_no
7  JOIN salaries s ON de.emp_no = s.emp_no
8  WHERE s.to_date = '9999-01-01'
9  GROUP BY d.dept_name
10 HAVING AVG(s.salary) > 70000
11     AND COUNT(*) > 50; -- Adjust threshold for 500 sample dataset
12
```

Data Output    Messages    Notifications

≡+ ↻ ⌂ ↻ ⏷ ↻ ⏸ ↻ ⏹ ↻ SQL

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

## EXAMPLE 4 :

Query Query History

```
1 ▾ SELECT
2     d.dept_name,
3     FLOOR(s.salary/10000)*10000 AS Salary_Bucket,
4     COUNT(*) AS Employees,
5     ROUND(100.0*COUNT(*)/SUM(COUNT(*))) OVER (PARTITION BY d.dept_name),1) || '%' AS Distribution
6   FROM departments d
7   JOIN dept_emp de ON d.dept_no = de.dept_no
8   JOIN salaries s ON de.emp_no = s.emp_no
9   WHERE s.to_date = '9999-01-01'
10  GROUP BY d.dept_name, Salary_Bucket
11  ORDER BY d.dept_name, Salary_Bucket;
12
```

Data Output Messages Notifications

SQL

	dept_name character varying (50)	salary_bucket double precision	employees bigint	distribution 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%