TASK-3 :

1. INNER JOIN: Employee-Department Mapping

Query | Query History

```
SELECT
    e.emp_no,
    CONCAT(e.first_name,' ',e.last_name) AS name,
    d.dept_name
FROM employees e
INNER JOIN dept_emp de ON e.emp_no = de.emp_no
INNER JOIN departments d ON de.dept_no = d.dept_no
WHERE de.to_date = '9999-01-01'
LIMIT 5;
```

Data Output | Messages | Notifications

	emp_no integer	name text	dept_name character varying (50)
1	10001	Mia Johnson	Marketing
2	10002	Sophia Thomas	Finance
3	10003	Robert Moore	Sales
4	10004	Sophia Martin...	Production
5	10005	Linda Rodriguez	Sales

2. LEFT JOIN: Employees with Manager Info

```sql
SELECT
    e.emp_no,
    CONCAT(e.first_name,' ',e.last_name) AS employee,
    CONCAT(m.first_name,' ',m.last_name) AS manager
FROM employees e
LEFT JOIN dept_manager dm ON dm.dept_no = (
    SELECT dept_no FROM dept_emp
    WHERE emp_no = e.emp_no
    AND to_date = '9999-01-01'
)
LEFT JOIN employees m ON dm.emp_no = m.emp_no
WHERE dm.to_date = '9999-01-01'
LIMIT 5;
```

Data Output | Messages | Notifications

	emp_no [PK] integer	employee text	manager text
1	10496	Robert Martinez	Sophia Martinez
2	10494	Amelia Miller	Sophia Martinez
3	10490	David Smith	Sophia Martinez
4	10478	Richard Thompson	Sophia Martinez
5	10474	Charles Thomas	Sophia Martinez

3. UNION: Combine Active/Former Employees

Query | Query History

```sql
-- Active employees in Sales
SELECT
    de.emp_no,
    'Active' AS status,
    d.dept_name
FROM dept_emp de
JOIN departments d ON de.dept_no = d.dept_no
WHERE de.to_date = '9999-01-01'
AND d.dept_name = 'Sales'

UNION ALL

-- Former employees in Sales
SELECT
    de.emp_no,
    'Former' AS status,
    d.dept_name
FROM dept_emp de
JOIN departments d ON de.dept_no = d.dept_no
WHERE de.to_date <> '9999-01-01'
AND d.dept_name = 'Sales'
LIMIT 5;
```

Data Output | Messages | Notifications

	emp_no integer	status text	dept_name character varying (50)
1	10003	Active	Sales
2	10005	Active	Sales
3	10016	Active	Sales
4	10019	Active	Sales
5	10026	Active	Sales

4. Correlated Subquery: Current Salary

Query | Query History

```
1  SELECT
2      e.emp_no,
3      e.first_name,
4      (SELECT s.salary
5       FROM salaries s
6       WHERE s.emp_no = e.emp_no
7         AND s.to_date = '9999-01-01') AS current_salary
8  FROM employees e
9  WHERE e.emp_no BETWEEN 10001 AND 10010;
10
```

Data Output | Messages | Notifications

SQL

	emp_no [PK] integer	first_name character varying (50)	current_salary integer
1	10001	Mia	42604
2	10002	Sophia	92665
3	10003	Robert	60110
4	10004	Sophia	120258
5	10005	Linda	134486
6	10006	Mia	73222
7	10007	Joseph	113727
8	10008	Laura	87038
9	10009	Sophia	68418
10	10010	David	70496

5. Duplicate Handling: Unique Job Titles

Query Query History

```
1 SELECT DISTINCT t.title AS unique_titles
2 FROM titles t
3 WHERE t.to_date = '9999-01-01';
4
```

Data Output Messages Notifications

SQL

	unique_titles character varying (50)
1	Engineer
2	Junior Engineer
3	Senior Engineer
4	Manager
5	Analyst
6	Senior Analyst
7	Director
8	Staff
9	Senior Staff

6. Complex Scenario: Department Gender Ratios

Query | Query History

```sql
SELECT
    d.dept_name,
    ROUND(100.0 * SUM(CASE WHEN e.gender='F' THEN 1 ELSE 0 END) / COUNT(*),2) AS female_pct,
    ROUND(100.0 * SUM(CASE WHEN e.gender='M' THEN 1 ELSE 0 END) / COUNT(*),2) AS male_pct
FROM departments d
JOIN dept_emp de ON d.dept_no = de.dept_no
JOIN employees e ON de.emp_no = e.emp_no
WHERE de.to_date = '9999-01-01'
GROUP BY d.dept_name
HAVING COUNT(*) > 10;  -- Adjusted for 500 sample dataset

```

Data Output | Messages | Notifications

	dept_name character varying (50)	female_pct numeric	male_pct numeric
1	Research	48.04	51.96
2	Marketing	45.79	54.21
3	Finance	52.17	47.83
4	Sales	49.49	50.51
5	Production	45.00	55.00