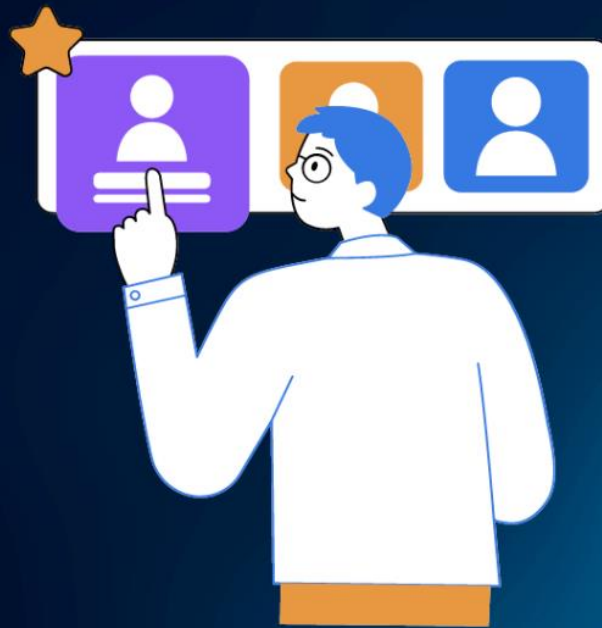


SQL CASE STUDY

DATA IN MOTION HUMAN RESOURCES



DATA IN MOTION

-- Create tables and insert data for analysis

-- Create 'departments' table

```
CREATE TABLE departments (  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    name VARCHAR(50),  
    manager_id INT  
);
```

--Create 'employees' table

```
CREATE TABLE employees (  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    name VARCHAR(50),  
    hire_date DATE,  
    job_title VARCHAR(50),  
    department_id INT REFERENCES departments(id)  
);
```

-- Create 'projects' table

```
CREATE TABLE projects (  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    name VARCHAR(50),  
    start_date DATE,  
    end_date DATE,  
    department_id INT REFERENCES departments(id)  
);
```

-- Insert data into 'departments'

```
INSERT INTO departments (name, manager_id)  
VALUES ('HR', 1), ('IT', 2), ('Sales', 3);
```

-- Insert data into 'employees'

```
INSERT INTO employees (name, hire_date, job_title, department_id)
```

```
VALUES ('John Doe', '2018-06-20', 'HR Manager', 1),
      ('Jane Smith', '2019-07-15', 'IT Manager', 2),
      ('Alice Johnson', '2020-01-10', 'Sales Manager', 3),
      ('Bob Miller', '2021-04-30', 'HR Associate', 1),
      ('Charlie Brown', '2022-10-01', 'IT Associate', 2),
      ('Dave Davis', '2023-03-15', 'Sales Associate', 3);
```

-- Insert data into 'projects'

```
INSERT INTO projects (name, start_date, end_date, department_id)
VALUES ('HR Project 1', '2023-01-01', '2023-06-30', 1),
      ('IT Project 1', '2023-02-01', '2023-07-31', 2),
      ('Sales Project 1', '2023-03-01', '2023-08-31', 3);
```

UPDATE departments

```
SET manager_id = (SELECT id FROM employees WHERE name = 'John Doe')
WHERE name = 'HR';
```

UPDATE departments

```
SET manager_id = (SELECT id FROM employees WHERE name = 'Jane Smith')
WHERE name = 'IT';
```

UPDATE departments

```
SET manager_id = (SELECT id FROM employees WHERE name = 'Alice Johnson')
WHERE name = 'Sales';
```

-- SQL Challenge Questions

--1. Find the longest ongoing project for each department.

```
SELECT d.name AS department_name,
       p.name AS project_name,
       p.start_date AS project_start_date,
       p.end_date AS project_end_date,
       DATEDIFF(DAY, p.start_date, GETDATE()) AS project_duration_days
FROM departments d
LEFT JOIN projects p ON d.id = p.department_id
WHERE p.end_date <= GETDATE();
```

ORDER BY d.name, project_duration_days DESC;

Results Messages

	department_n...	project_...	project_start_...	project_end_date	project_d...
1	HR	HR Project 1	2023-01-01	2023-06-30	250
2	IT	IT Project 1	2023-02-01	2023-07-31	219
3	Sales	Sales Proje...	2023-03-01	2023-08-31	191

--2. Find all employees who are not managers.

```
SELECT e.name AS employee_name, e.job_title
FROM employees e
LEFT JOIN departments d ON e.id = d.manager_id
WHERE d.id IS NULL;
```

Results Messages

	employee_name	job_title
1	Bob Miller	HR Associate
2	Charlie Brown	IT Associate
3	Dave Davis	Sales Associate

--3. Find all employees who have been hired after the start of a project in their department.

```
SELECT e.name AS employee_name, e.hire_date, p.name AS project_name, p.start_date AS project_start_date
FROM employees e
INNER JOIN projects p ON e.department_id = p.department_id
WHERE e.hire_date > p.start_date;
```

Results Messages

	employee_name	hire_date	project_name	project_start_date
1	Dave Davis	2023-03-15	Sales Project 1	2023-03-01

--4. Rank employees within each department based on their hire date (earliest hire gets the highest rank).

```
SELECT e.name AS employee_name, e.hire_date, d.name AS department_name,
       RANK() OVER (PARTITION BY e.department_id ORDER BY e.hire_date) AS hire_rank
```

```

FROM employees e
INNER JOIN departments d ON e.department_id = d.id
ORDER BY e.department_id, hire_rank;

```

Results Messages

	employee_name	hire_date	department_name	hire_rank
1	John Doe	2018-06-20	HR	1
2	Bob Miller	2021-04-30	HR	2
3	Jane Smith	2019-07-15	IT	1
4	Charlie Brown	2022-10-01	IT	2
5	Alice Johnson	2020-01-10	Sales	1
6	Dave Davis	2023-03-15	Sales	2

--5. Find the duration between the hire date of each employee and the hire date of the next employee hired in the same department.

```

WITH EmployeeWithNextHire AS (
    SELECT e.name AS employee_name, e.hire_date, d.name AS department_name,
           LEAD(e.hire_date) OVER (PARTITION BY e.department_id ORDER BY e.hire_date) AS next_hire_date
    FROM employees e
    INNER JOIN departments d ON e.department_id = d.id
)

```

```

SELECT employee_name, hire_date, department_name,
       DATEDIFF(DAY, hire_date, next_hire_date) AS hire_duration_days
FROM EmployeeWithNextHire;

```

Results Messages

	employee_name	hire_date	department_name	hire_duration_days
1	John Doe	2018-06-20	HR	1045
2	Bob Miller	2021-04-30	HR	NULL
3	Jane Smith	2019-07-15	IT	1174
4	Charlie Brown	2022-10-01	IT	NULL
5	Alice Johnson	2020-01-10	Sales	1160
6	Dave Davis	2023-03-15	Sales	NULL