SQL CASE STUDY 2

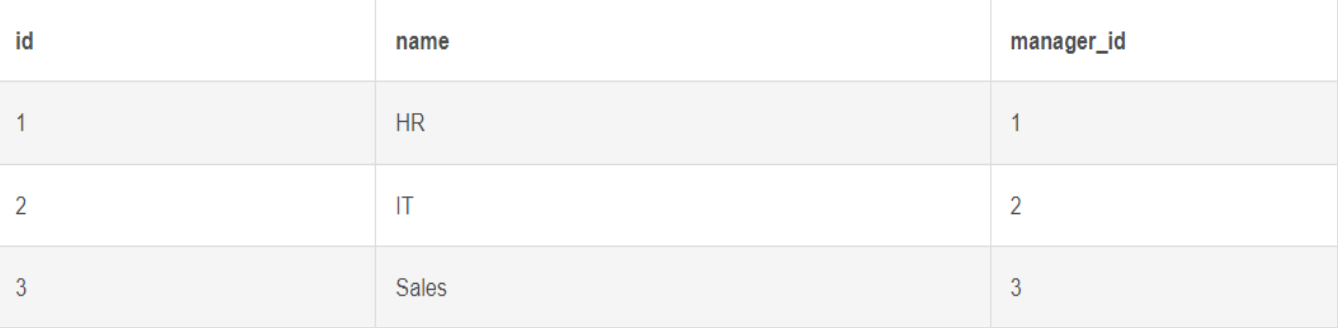
Human Resources

Introduction

The task was to analyze the employees data and project information to gather insights.

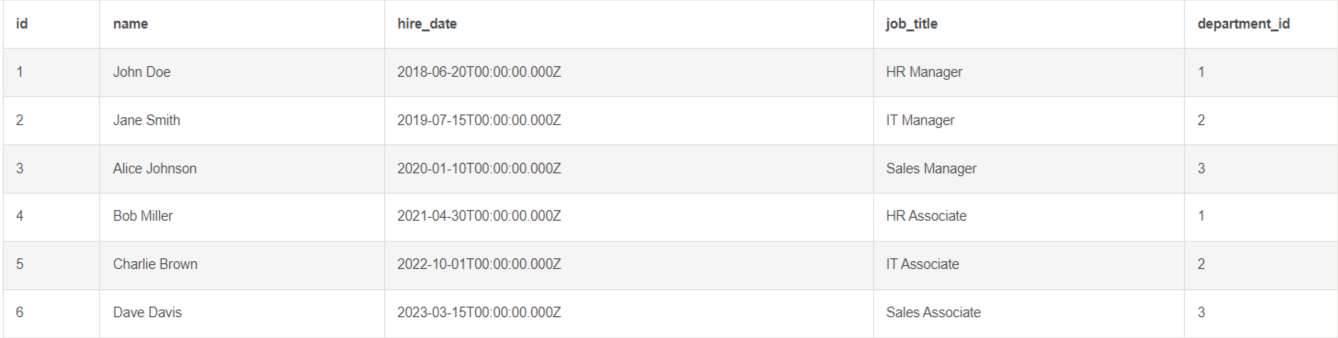
DATASET

DEPARTMENT



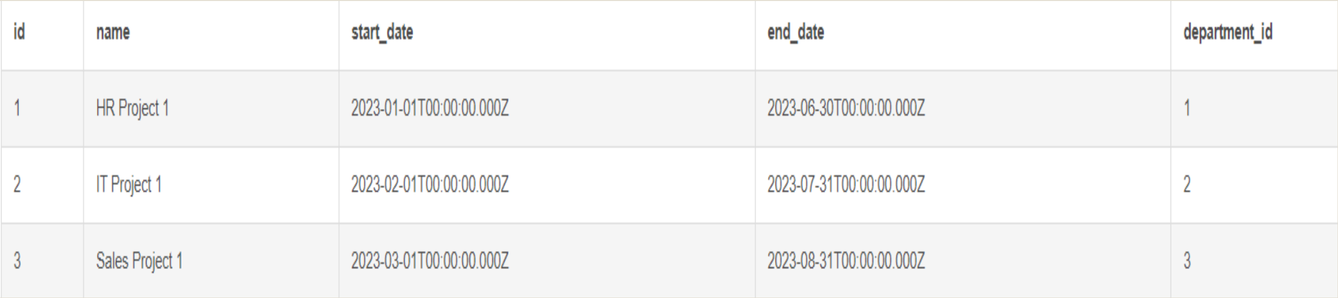
id	name	manager_id
1	HR	1
2	IT	2
3	Sales	3

EMPLOYEES



id	name	hire_date	job_title	department_id
1	John Doe	2018-06-20T00:00:00.000Z	HR Manager	1
2	Jane Smith	2019-07-15T00:00:00.000Z	IT Manager	2
3	Alice Johnson	2020-01-10T00:00:00.000Z	Sales Manager	3
4	Bob Miller	2021-04-30T00:00:00.000Z	HR Associate	1
5	Charlie Brown	2022-10-01T00:00:00.000Z	IT Associate	2
6	Dave Davis	2023-03-15T00:00:00.000Z	Sales Associate	3

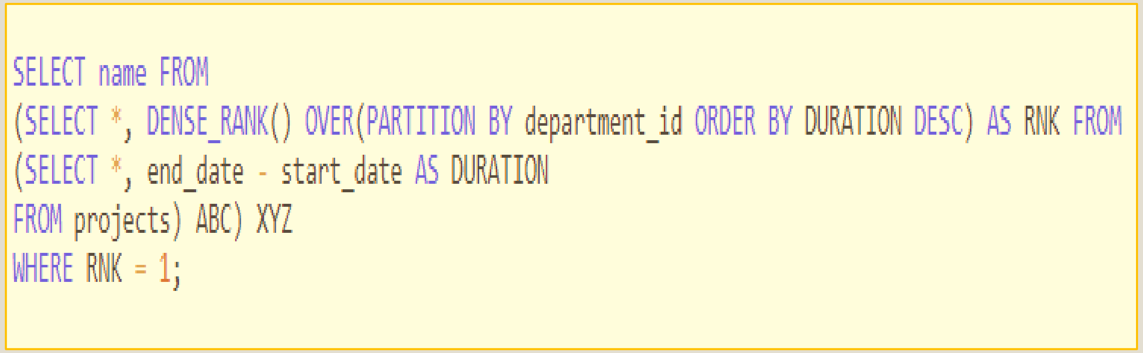
PROJECTS



id	name	start_date	end_date	department_id
1	HR Project 1	2023-01-01T00:00:00.000Z	2023-06-30T00:00:00.000Z	1
2	IT Project 1	2023-02-01T00:00:00.000Z	2023-07-31T00:00:00.000Z	2
3	Sales Project 1	2023-03-01T00:00:00.000Z	2023-08-31T00:00:00.000Z	3

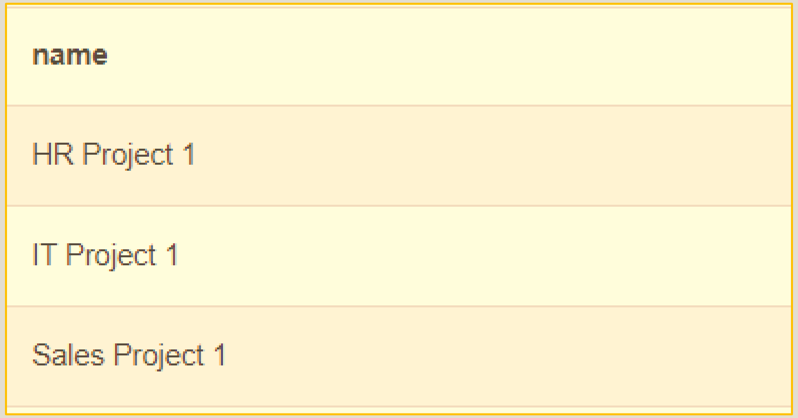
1. Find the longest ongoing project for each department.

QUERY



```
SELECT name FROM
(SELECT *, DENSE_RANK() OVER(PARTITION BY department_id ORDER BY DURATION DESC) AS RNK FROM
(SELECT *, end_date - start_date AS DURATION
FROM projects) ABC) XYZ
WHERE RNK = 1;
```

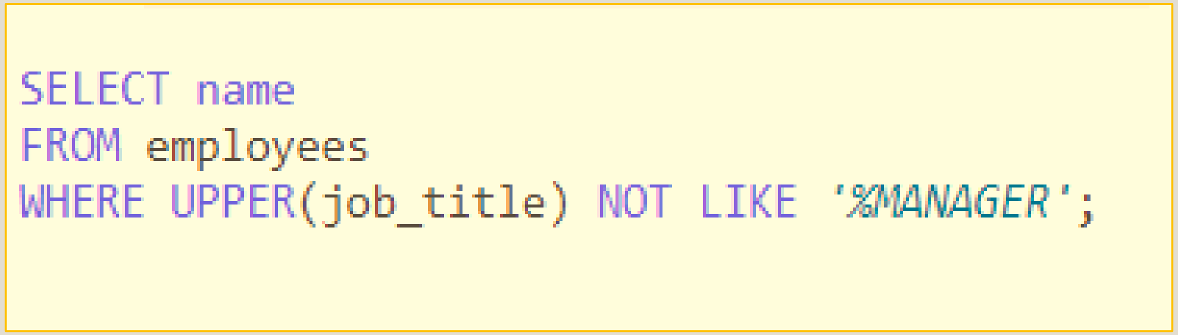
OUTPUT



name
HR Project 1
IT Project 1
Sales Project 1

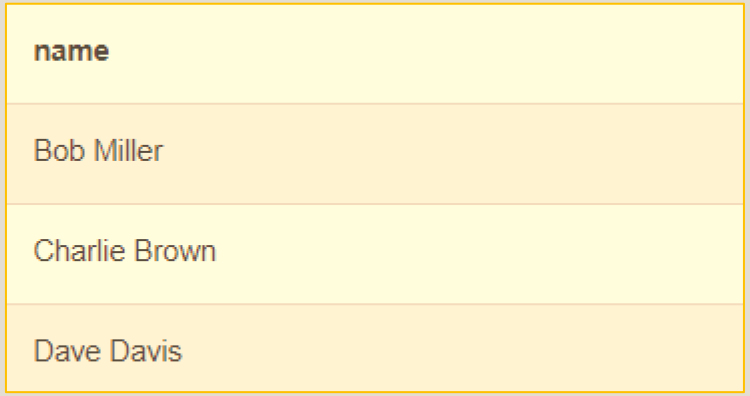
2. Find all employees who are not managers.

QUERY



```
SELECT name
FROM employees
WHERE UPPER(job_title) NOT LIKE '%MANAGER';
```

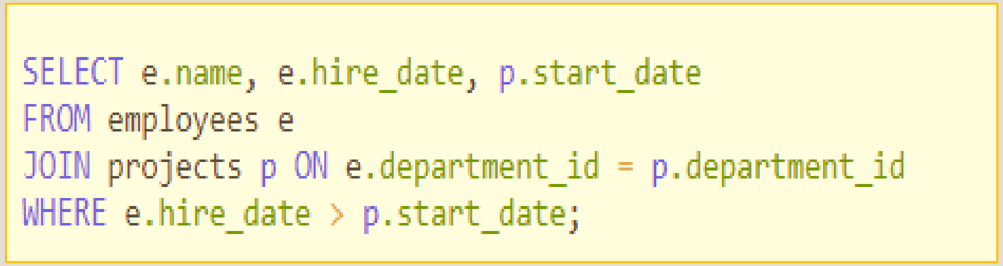
OUTPUT



name
Bob Miller
Charlie Brown
Dave Davis

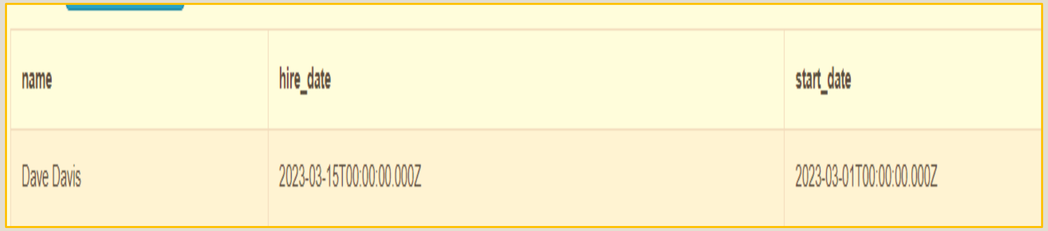
3. Find all employees who have been hired after the start of a project in their department.

QUERY



```
SELECT e.name, e.hire_date, p.start_date
FROM employees e
JOIN projects p ON e.department_id = p.department_id
WHERE e.hire_date > p.start_date;
```

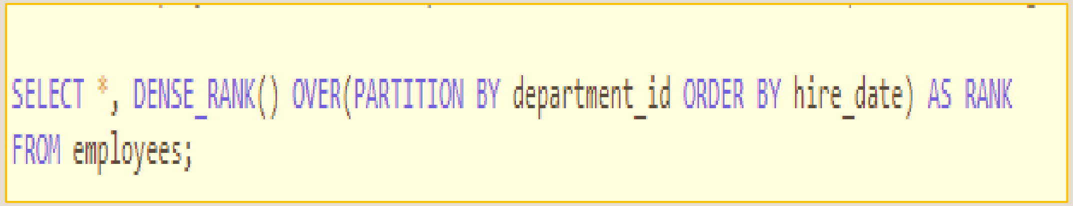
OUTPUT



name	hire_date	start_date
Dave Davis	2023-03-15T00:00:00.000Z	2023-03-01T00:00:00.000Z

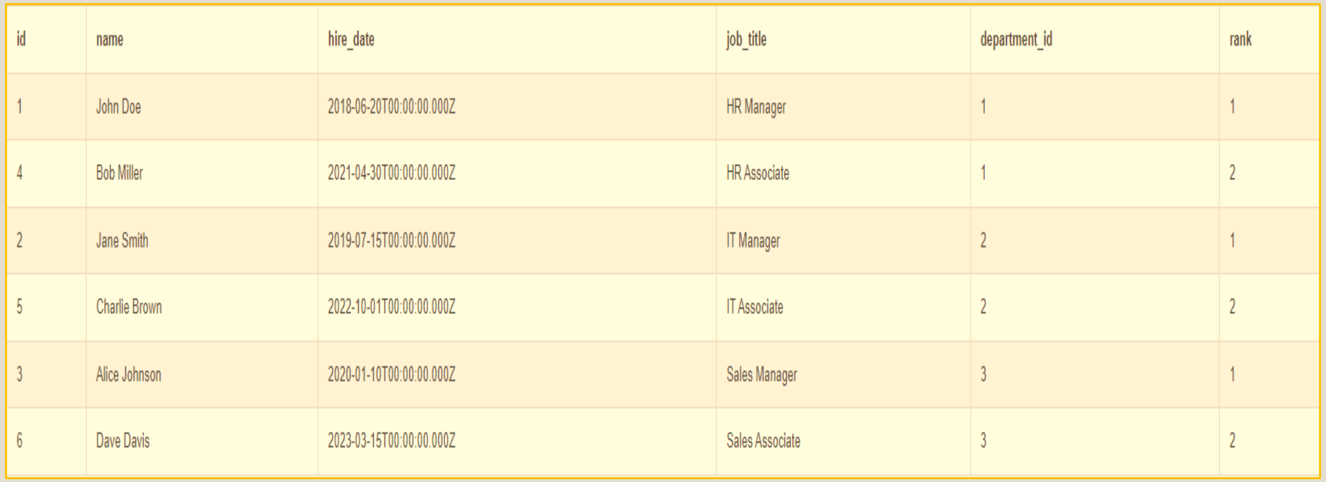
4. Rank employees within each department based on their hire date (earliest hire gets the highest rank).

QUERY



```
SELECT *, DENSE_RANK() OVER(PARTITION BY department_id ORDER BY hire_date) AS RANK
FROM employees;
```

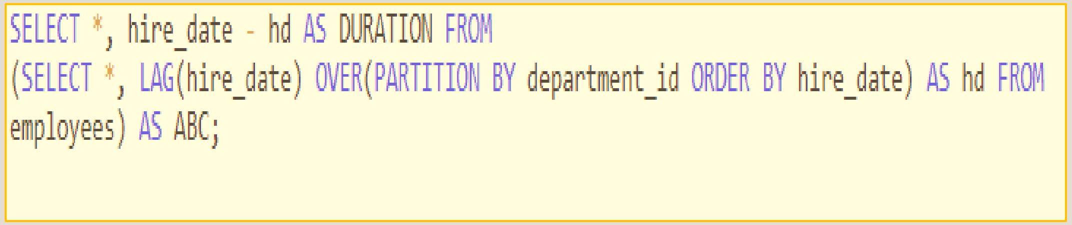
OUTPUT



id	name	hire_date	job_title	department_id	rank
1	John Doe	2018-06-20T00:00:00.000Z	HR Manager	1	1
4	Bob Miller	2021-04-30T00:00:00.000Z	HR Associate	1	2
2	Jane Smith	2019-07-15T00:00:00.000Z	IT Manager	2	1
5	Charlie Brown	2022-10-01T00:00:00.000Z	IT Associate	2	2
3	Alice Johnson	2020-01-10T00:00:00.000Z	Sales Manager	3	1
6	Dave Davis	2023-03-15T00:00:00.000Z	Sales Associate	3	2

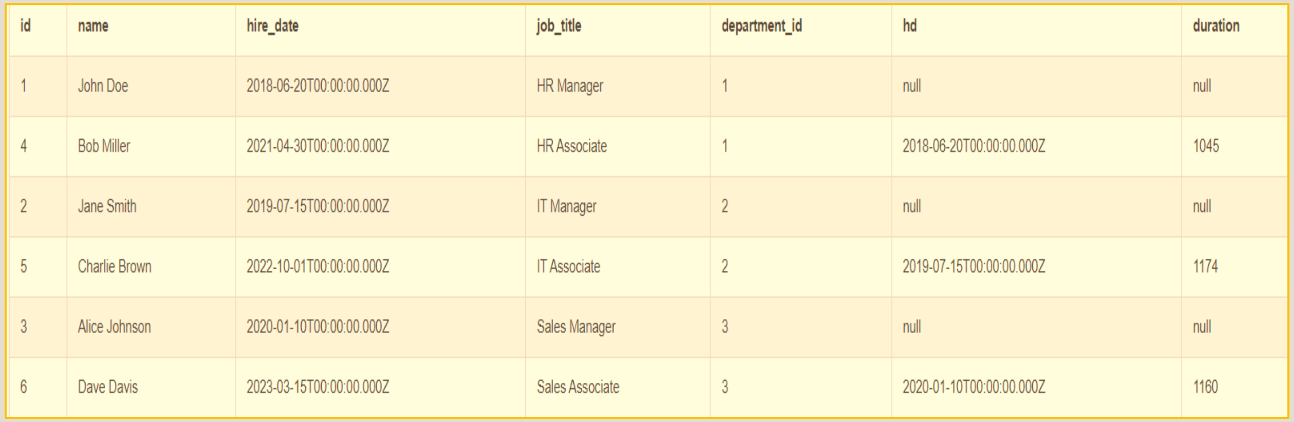
5. Find the duration between the hire date of each employee and the hire date of the next employee hired in the same department.

QUERY



```
SELECT *, hire_date - hd AS DURATION FROM
(SELECT *, LAG(hire_date) OVER(PARTITION BY department_id ORDER BY hire_date) AS hd FROM
employees) AS ABC;
```

OUTPUT



id	name	hire_date	job_title	department_id	hd	duration
1	John Doe	2018-06-20T00:00:00.000Z	HR Manager	1	null	null
4	Bob Miller	2021-04-30T00:00:00.000Z	HR Associate	1	2018-06-20T00:00:00.000Z	1045
2	Jane Smith	2019-07-15T00:00:00.000Z	IT Manager	2	null	null
5	Charlie Brown	2022-10-01T00:00:00.000Z	IT Associate	2	2019-07-15T00:00:00.000Z	1174
3	Alice Johnson	2020-01-10T00:00:00.000Z	Sales Manager	3	null	null
6	Dave Davis	2023-03-15T00:00:00.000Z	Sales Associate	3	2020-01-10T00:00:00.000Z	1160