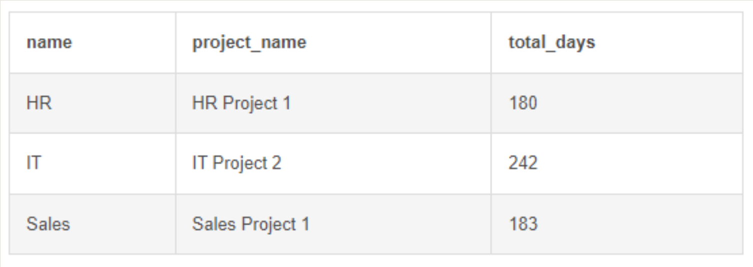
SQL CASE STUDY

Human Resources

Q1. FIND THE LONGEST ONGOING PROJECT FOR EACH DEPARTMENT

OUTPUT



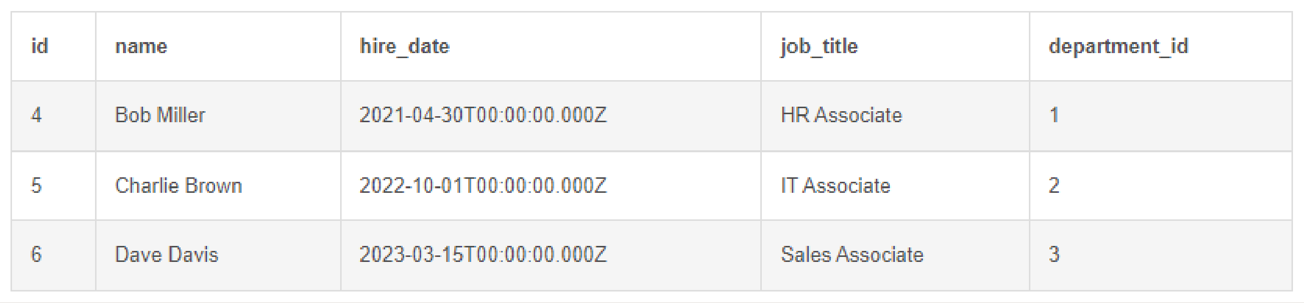
name	project_name	total_days
HR	HR Project 1	180
IT	IT Project 2	242
Sales	Sales Project 1	183

```
with project_duration as(
   select p.department_id dept_id, p.name project_name,
    p.start_date, p.end_date,
          end_date - start_date total_days
   from projects p
   where p.end_date > current_date
),
rank_data as(
  select p.dept_id, p.project_name, p.total_days,
       rank () over (partition by p.dept_id order by total_days
    desc) as rank
  from project_duration p
)
select d.name, r.project_name, r.total_days
from rank_data r
join departments d
on d.id = r.dept_id
where rank = 1
```

Q2. FIND ALL EMPLOYEES WHO ARE NOT MANAGERS

```
select * from employees where id not in (select manager_id
    from departments);
```

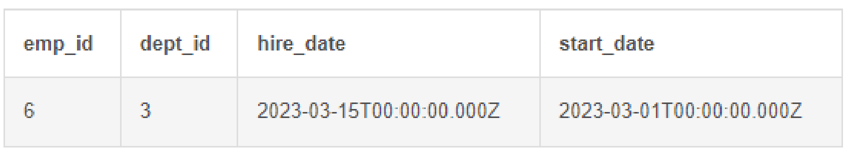
OUTPUT



id	name	hire_date	job_title	department_id
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

Q3. FIND ALL EMPLOYEES WHO HAVE BEEN HIRED AFTER THE START OF A PROJECT IN THEIR DEPARTMENT

OUTPUT



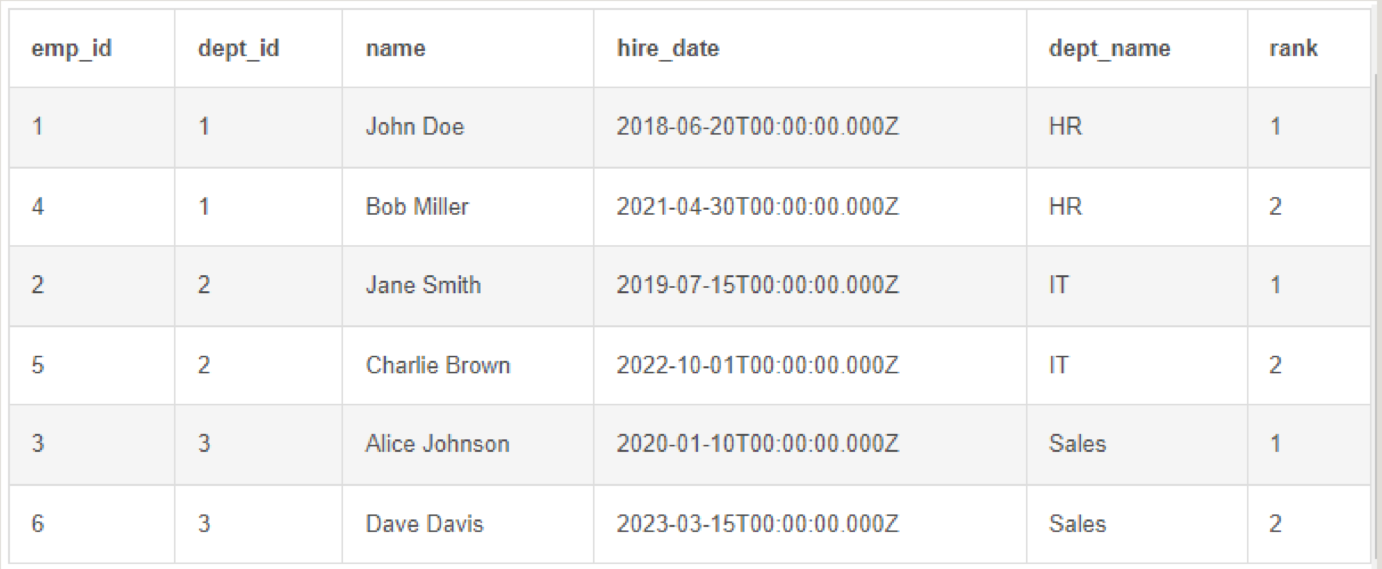
emp_id	dept_id	hire_date	start_date
6	3	2023-03-15T00:00:00.000Z	2023-03-01T00:00:00.000Z

```
with emp_Details as(
 select e.id emp_id, d.id dept_id, e.hire_date
 from employees e
 join departments d
 on e.department_id = d.id
),
project_details as(
 select d.id dept_id, p.id project_id, p.start_date
 from projects p
 join departments d
 on d.id = p.department_id
)
select distinct e.emp_id, p.dept_id, e.hire_date, p.start_date
from emp_Details e
join project_details p
on e.dept_id = p.dept_id
where e.hire_date  > p.start_date
order by p.dept_id;
```

Q4. RANK EMPLOYEES WITHIN EACH DEPARTMENT BASED ON THEIR HIRE DATE (EARLIEST HIRE GETS THE HIGHEST RANK)

```
select e.id emp_id, d.id dept_id, e.name, e.hire_date, d.name dept_name,
     dense_rank() over (partition by e.department_id order by e.hire_date asc) as
    rank
from employees e
join departments d
on e.department_id = d.id
```

OUTPUT

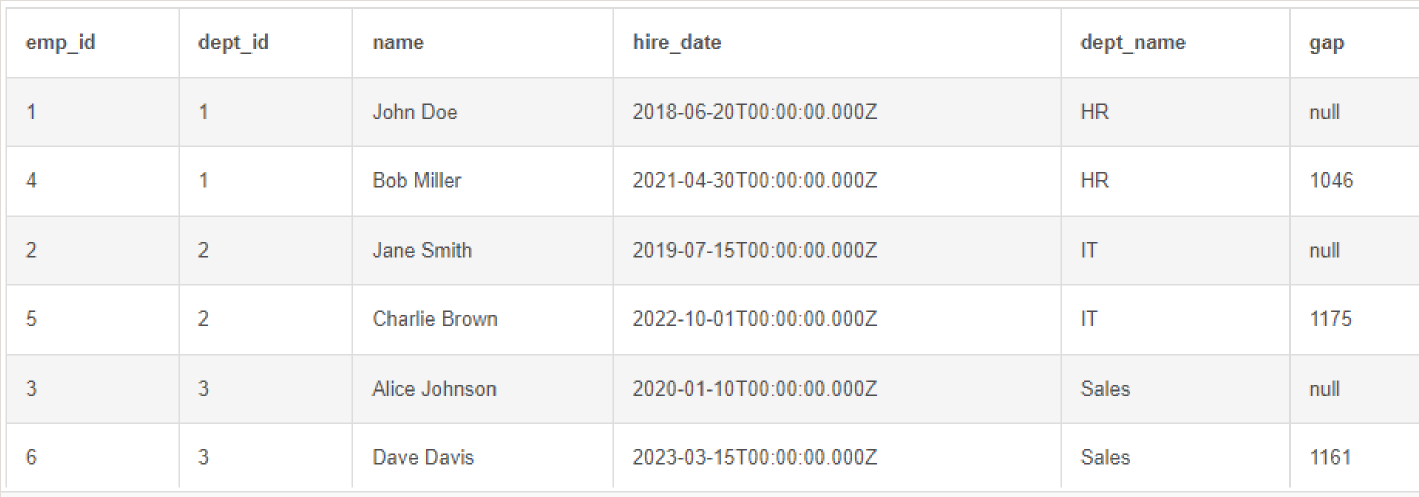


emp_id	dept_id	name	hire_date	dept_name	rank
1	1	John Doe	2018-06-20T00:00:00.000Z	HR	1
4	1	Bob Miller	2021-04-30T00:00:00.000Z	HR	2
2	2	Jane Smith	2019-07-15T00:00:00.000Z	IT	1
5	2	Charlie Brown	2022-10-01T00:00:00.000Z	IT	2
3	3	Alice Johnson	2020-01-10T00:00:00.000Z	Sales	1
6	3	Dave Davis	2023-03-15T00:00:00.000Z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

Q5. FIND THE DURATION BETWEEN THE HIRE DATE OF EACH EMPLOYEE AND THE HIRE DATE OF THE NEXT EMPLOYEE HIRED IN THE SAME DEPARTMENT

```
select e.id emp_id, d.id dept_id, e.name, e.hire_date, d.name dept_name,
    abs(lag(e.hire_date) over (partition by d.id order by e.id) - e.hire_date - 1) as gap
from employees e
join departments d
on e.department_id  = d.id
--group by d.id
order by d.id, e.id, e.hire_date
```

OUTPUT



emp_id	dept_id	name	hire_date	dept_name	gap
1	1	John Doe	2018-06-20T00:00:00.000Z	HR	null
4	1	Bob Miller	2021-04-30T00:00:00.000Z	HR	1046
2	2	Jane Smith	2019-07-15T00:00:00.000Z	IT	null
5	2	Charlie Brown	2022-10-01T00:00:00.000Z	IT	1175
3	3	Alice Johnson	2020-01-10T00:00:00.000Z	Sales	null
6	3	Dave Davis	2023-03-15T00:00:00.000Z	Sales	1161

FUNCTIONS USED

Aggregate Functions: COUNT(), MAX(), MIN()

RANK ()

DENSE_RANK()

PARTITION BY()

GROUP BY()

LAG()

Common Table Expression

Window Function