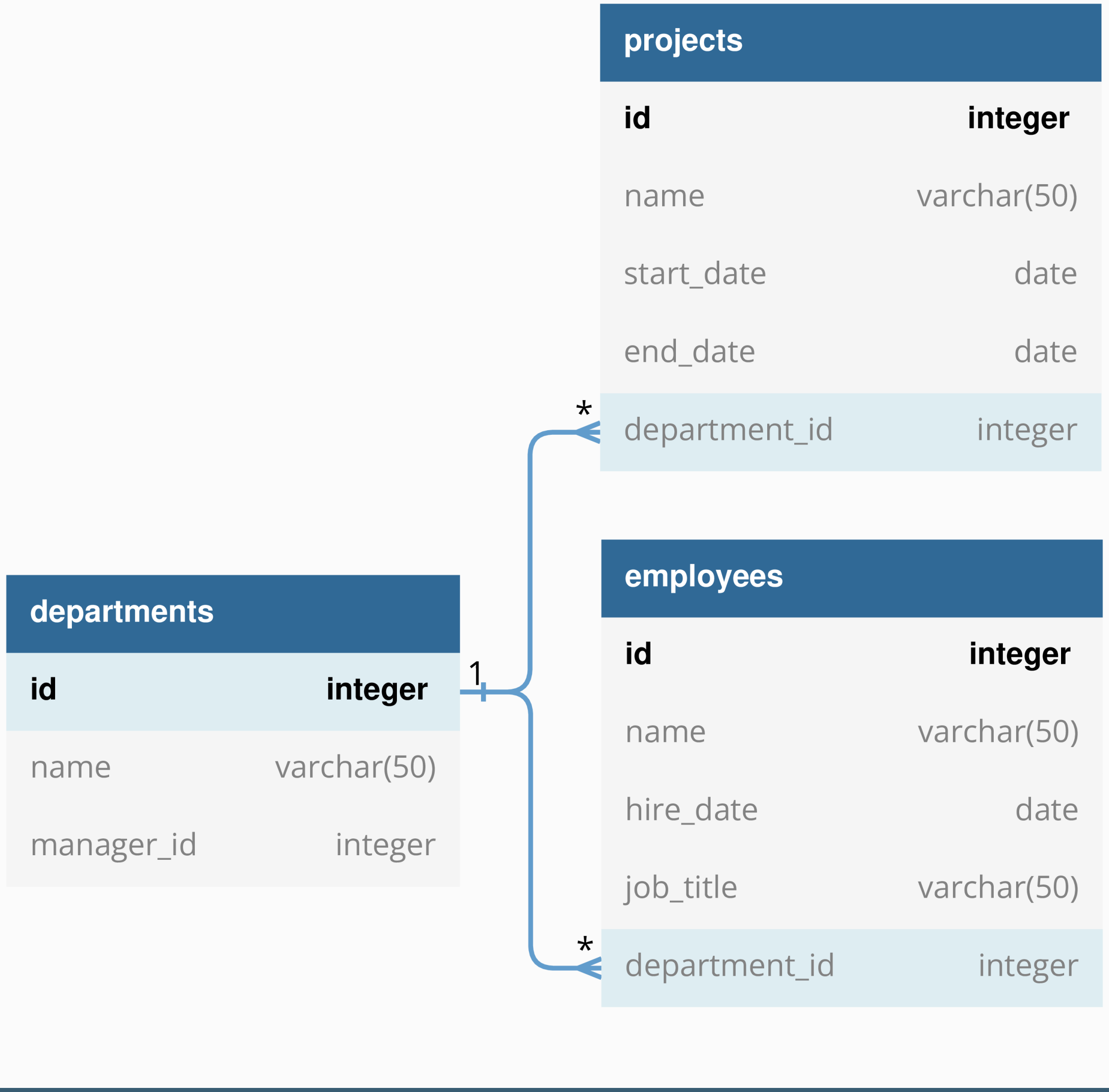
# SQL CASE STUDY

## DATA IN MOTION
## HUMAN RESOURCES

DATA IN MOTION

BY GLADYS CORTES

# ER Diagram


projects
id integer
name varchar(50)
start_date date
end_date date
department_id integer
*
departments
id integer
name varchar(50)
manager_id integer
1
employees
id integer
name varchar(50)
hire_date date
job_title varchar(50)
department_id integer
*

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

WITH cte AS (
    SELECT *
        ,RANK() OVER(
            PARTITION BY department_id
            ORDER BY days_ongoing DESC)
    FROM (
        SELECT
             department_id
            ,name
            ,COALESCE(end_date,CURRENT_DATE) - start_date AS days_ongoing
        FROM projects
        WHERE --project is ongoing
            end_date > CURRENT_DATE OR
            end_date IS NULL
    ) AS sub
)

SELECT
     d.name AS department
    ,cte.name AS project
    ,cte.days_ongoing
FROM
    cte
    JOIN departments d ON cte.department_id = d.id
WHERE --duration is longest
    cte.rank = 1;

```

| | department<br>character varying (50) | project<br>character varying (50) | days_ongoing<br>integer |
|---|---|---|---|
| 1 | HR | HR Project 1 | 180 |
| 2 | IT | IT Project 1 | 180 |
| 3 | Sales | Sales Project 1 | 183 |

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

SELECT
     id
    ,name
    ,job_title
FROM employees
WHERE job_title NOT ILIKE '%manager%';

```

| | id<br>[PK] integer | name<br>character varying (50) | job_title<br>character varying (50) |
|---|---|---|---|
| 1 | 4 | Bob Miller | HR Associate |
| 2 | 5 | Charlie Brown | IT Associate |
| 3 | 6 | Dave Davis | Sales Associate |

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

SELECT
     e.name AS employee
    ,e.hire_date
    ,pr.name AS project
    ,pr.start_date
FROM
    employees e
    JOIN departments d ON e.department_id = d.id
    JOIN projects pr ON d.id = pr.department_id
WHERE
    e.hire_date > pr.start_date;

```

| | employee<br>character varying (50) | hire_date<br>date | project<br>character varying (50) | start_date<br>date |
|---|---|---|---|---|
| 1 | Dave Davis | 2023-03-15 | Sales Project 1 | 2023-03-01 |

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

SELECT
     d.name AS department
    ,e.name AS employee
    ,e.hire_date
    ,RANK() OVER(PARTITION BY d.id ORDER BY e.hire_date)
FROM
    employees e
    JOIN departments d ON e.department_id = d.id;

```

| | department<br>character varying (50) | employee<br>character varying (50) | hire_date<br>date | rank<br>bigint |
|---|---|---|---|---|
| 1 | HR | John Doe | 2018-06-20 | 1 |
| 2 | HR | Bob Miller | 2021-04-30 | 2 |
| 3 | IT | Jane Smith | 2019-07-15 | 1 |
| 4 | IT | Charlie Brown | 2022-10-01 | 2 |
| 5 | Sales | Alice Johnson | 2020-01-10 | 1 |
| 6 | Sales | Dave Davis | 2023-03-15 | 2 |

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

SELECT
     d.name AS department
    ,e.name AS employee
    ,e.hire_date
    ,LEAD(e.hire_date) OVER(
        PARTITION BY d.id
        ORDER BY e.hire_date
    ) - e.hire_date || ' days' AS next_hiring
FROM
    employees e
    JOIN departments d ON e.department_id = d.id;

```

| | department<br>character varying (50) | employee<br>character varying (50) | hire_date<br>date | next_hiring<br>text |
|---|---|---|---|---|
| 1 | HR | John Doe | 2018-06-20 | 1045 days |
| 2 | HR | Bob Miller | 2021-04-30 | [null] |
| 3 | IT | Jane Smith | 2019-07-15 | 1174 days |
| 4 | IT | Charlie Brown | 2022-10-01 | [null] |
| 5 | Sales | Alice Johnson | 2020-01-10 | 1160 days |
| 6 | Sales | Dave Davis | 2023-03-15 | [null] |

# THANK YOU!

GLADYS CORTES