**# SQL Challenge**

**Context**

As data engineers at Pewlett Hackard, we are tasked with a research project about people whom the company employed during the 1980s and 1990s. The available employee database from that period are six CSV files. For this project, we are expected to:

* design the tables to hold the data from the CSV files,
* import the CSV files into a SQL database, and
* answer questions about the data.

**Tasks and Findings:**

1. **Data Modeling**

The [QuickDBD](https://www.quickdatabasediagrams.com/) was used sketch an Entity Relationship Diagram of the six CSV files.

1. **Data Engineering**

* Using the provided information, a table schema is created for each of the six CSV files.
* The data types, primary keys, foreign keys, and other constraints are specified.
* The 6 CSV files are imported into its corresponding SQL table.

1. **Data Analysis**

The analysis results are the followings:

* Table showing employee number, last name, first name, sex, and salary of each employee,
* Table showing first name, last name, and hire date for the employees who were hired in 1986,
* Table showing manager of each department along with their department number, department name, employee number, last name, and first name,
* Table showing department number for each employee along with that employee’s employee number, last name, first name, and department name,
* Table showing first name, last name, and sex of each employee whose first name is Hercules and whose last name begins with the letter B,
* Table showing each employee in the Sales department (Department no. d007), including their employee number, last name, and first name,
* Table showing each employee in the Sales and Development departments (Department no. d005, d007), including their employee number, last name, first name, and department name, and
* Table showing frequency counts, in descending order, of all the employees' last names (that is, how many employees share each last name).

**References**

Class recordings were reviewed, and online research was leveraged to complete the assignment.

**List of Assignment Files**

EmployeeSQL: folder containing the 6 CSV files,

QuickDBD-ERD\_tables: sketch diagram of the Entity Relationship between the six CSV files

ERD tables\_file: specified data types, primary keys, foreign keys, and other constraints

Employees schema: schema for creating 6 tables specifying data types, primary keys, and foreign keys,

Employees\_Notnull\_Columnlength: using NOT NULL condition and defining value length for columns,

Employees\_query: file of the queries for the data analysis

**Data modeling**

# Entity Relationship Diagram of the tables

# specified data types, primary keys, foreign keys, and other constraints

Departments

-

Department\_no INTEGER PK

Department\_name VARCHAR

-

Titles

-

Title\_id INTEGER PK

Title VARCHAR

-

Employees

-

Emp\_no INTEGER PK

Emp\_title\_id INTEGER FK >- Titles.Title\_id

Birth\_date Date

First\_name VARCHAR

Last\_name VARCHAR

Sex VARCHAR

Hire\_date Date

-

Depart\_Emp

-

Emp\_no INTEGER PK FK >- Employees.Emp\_no

Department\_no INTEGER FK >- Departments.Department\_no

-

Dept\_Manager

-

Dept\_no INTEGER FK >- Departments.Department\_no

Emp\_no INTEGER FK >- Employees.Emp\_no

-

Salaries

-

Emp\_no INTEGER FK >- Employees.Emp\_no

Salary INTEGER

-

**Data engineering**

# Import each CSV file into its corresponding SQL table.

-- Drop table if exists

DROP TABLE departments;

-- Create new table

CREATE TABLE departments (

Department\_no VARCHAR,

Department\_name VARCHAR

);

select \* from departments;

-- Drop table if exists

DROP TABLE titles;

CREATE TABLE titles (

Title\_id VARCHAR,

Title VARCHAR

);

select \* from titles;

-- Drop table if exists

DROP TABLE employees;

CREATE TABLE employees (

Emp\_no VARCHAR,

Emp\_title\_id VARCHAR,

Birth\_date VARCHAR,

First\_name VARCHAR,

Last\_name VARCHAR,

Sex VARCHAR,

Hire\_date VARCHAR

);

select \* from employees;

-- Drop table if exists

DROP TABLE Depart\_emp;

CREATE TABLE Depart\_emp (

Emp\_no INTEGER,

Department\_no VARCHAR

);

select \* from Depart\_emp;

-- Drop table if exists

DROP TABLE Depart\_manager;

CREATE TABLE Depart\_manager (

Dept\_no VARCHAR,

Emp\_no VARCHAR

);

select \* from Depart\_manager;

-- Drop table if exists

DROP TABLE salaries;

CREATE TABLE salaries (

Emp\_no INTEGER,

Salary INTEGER

);

select \* from salaries;

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

-- Creating tables for PHemployee\_db

CREATE TABLE departments (

dept\_no VARCHAR,

dept\_name VARCHAR (9) NOT NULL,

PRIMARY KEY (dept\_no),

UNIQUE (dept\_name)

);

CREATE TABLE titles (

Title\_id VARCHAR,

title VARCHAR (7) NOT NULL,

PRIMARY KEY (Title\_id)

);

CREATE TABLE employees (

emp\_no INT,

emp\_title\_id VARCHAR (4) NOT NULL,

birth\_date DATE NOT NULL,

first\_name VARCHAR NOT NULL,

last\_name VARCHAR NOT NULL,

sex VARCHAR NOT NULL,

hire\_date DATE NOT NULL,

FOREIGN KEY (emp\_title\_id) REFERENCES titles (Title\_id),

PRIMARY KEY (emp\_no)

);

CREATE TABLE dept\_manager (

dept\_no VARCHAR (9) NOT NULL,

emp\_no INT,

FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no),

FOREIGN KEY (dept\_no) REFERENCES departments (dept\_no),

PRIMARY KEY (emp\_no, dept\_no)

);

CREATE TABLE salaries (

emp\_no INT,

salary INT NOT NULL,

FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no),

PRIMARY KEY (emp\_no)

);

CREATE TABLE dept\_emp (

emp\_no INT,

dept\_no VARCHAR (9) NOT NULL,

FOREIGN KEY (emp\_no) REFERENCES employees (emp\_no),

FOREIGN KEY (dept\_no) REFERENCES departments (dept\_no),

PRIMARY KEY (emp\_no, dept\_no)

);

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

-- Create new table

CREATE TABLE departments (

Department\_no VARCHAR,

Department\_name VARCHAR (9) NOT NULL,

PRIMARY KEY (Department\_no)

);

CREATE TABLE titles (

Title\_id VARCHAR,

Title VARCHAR (7) NOT NULL,

PRIMARY KEY (Title\_id)

);

CREATE TABLE employees (

Emp\_no INT,

Emp\_title\_id VARCHAR (4) NOT NULL,

Birth\_date VARCHAR NOT NULL,

First\_name VARCHAR NOT NULL,

Last\_name VARCHAR NOT NULL,

Sex VARCHAR NOT NULL,

Hire\_date VARCHAR NOT NULL,

FOREIGN KEY (Emp\_title\_id) REFERENCES titles (Title\_id),

PRIMARY KEY (Emp\_no)

);

CREATE TABLE Depart\_emp (

Emp\_no INT NOT NULL,

Department\_no VARCHAR (9) NOT NULL,

FOREIGN KEY (Emp\_no) REFERENCES employees (Emp\_no),

FOREIGN KEY (Department\_no) REFERENCES departments (Department\_no),

PRIMARY KEY (Emp\_no, Department\_no)

);

CREATE TABLE Depart\_manager (

Department\_no VARCHAR (9) NOT NULL,

Emp\_no INT NOT NULL,

FOREIGN KEY (Emp\_no) REFERENCES employees (Emp\_no),

FOREIGN KEY (Department\_no) REFERENCES departments (Department\_no),

PRIMARY KEY (Emp\_no, Department\_no)

);

CREATE TABLE salaries (

Emp\_no INT,

Salary INT NOT NULL,

FOREIGN KEY (Emp\_no) REFERENCES employees (Emp\_no),

PRIMARY KEY (Emp\_no)

);

**Data analysis**

SELECT employees.Emp\_no, employees.last\_name, employees.first\_name, employees.sex, salaries.salary

FROM employees

JOIN salaries ON employees.Emp\_no = salaries.Emp\_no;

SELECT first\_name, last\_name, hire\_date

FROM employees

WHERE hire\_date LIKE '%1986';

SELECT depart\_manager.Department\_no, departments.Department\_name, depart\_manager.emp\_no, employees.last\_name, employees.first\_name

FROM depart\_manager

JOIN departments ON depart\_manager.Department\_no = departments.Department\_no

JOIN employees ON depart\_manager.Emp\_no = employees.Emp\_no;

SELECT employees.Emp\_no, employees.last\_name, employees.first\_name, departments.Department\_name

FROM employees

JOIN depart\_emp ON employees.Emp\_no = depart\_emp.Emp\_no

JOIN departments ON depart\_emp.Department\_no = departments.Department\_no;

SELECT first\_name, last\_name, sex

FROM employees

WHERE first\_name = 'Hercules' and last\_name LIKE 'B%';

SELECT employees.Emp\_no, employees.last\_name, employees.first\_name, departments.Department\_name

FROM employees

JOIN depart\_emp ON employees.Emp\_no = depart\_emp.Emp\_no

JOIN departments ON depart\_emp.Department\_no = departments.Department\_no

WHERE departments.Department\_no = 'd007';

SELECT employees.Emp\_no, employees.last\_name, employees.first\_name, departments.Department\_name

FROM employees

JOIN depart\_emp ON employees.Emp\_no = depart\_emp.Emp\_no

JOIN departments ON depart\_emp.Department\_no = departments.Department\_no

WHERE departments.Department\_no = 'd005' or departments.Department\_no = 'd007';

SELECT last\_name, COUNT(last\_name) AS "Freq count"

FROM employees

GROUP BY last\_name

ORDER BY "Freq count" DESC;

Reference: [sql-challenge/EmployeeSQL/employee\_schema.sql at master · Jiuhe2020/sql-challenge (github.com)](https://github.com/Jiuhe2020/sql-challenge/blob/master/EmployeeSQL/employee_schema.sql)