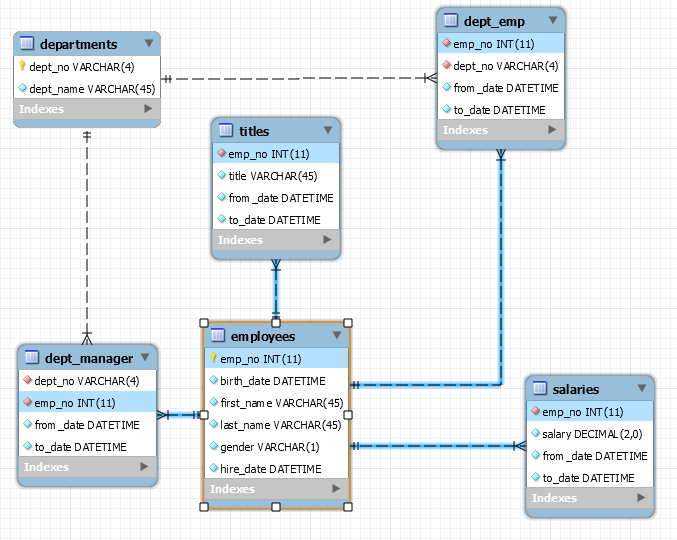
****

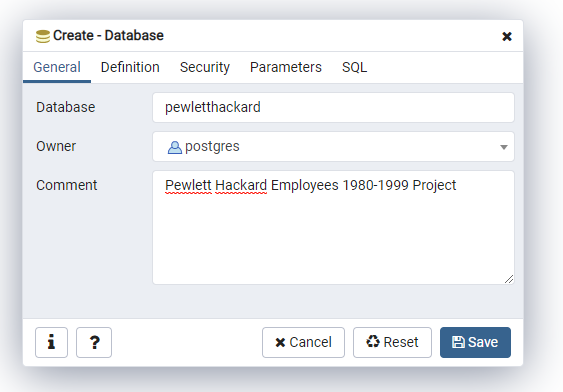
**Background:**

In this assignment, you will design the tables to hold data in the CSVs, import the CSVs into a SQL database, and answer questions about the data. In other words, you will perform:  
  
**Data Modeling:**



**Data Engineering:**  
  
**Observation:** MySQL & PostGres SQL syntax is not to similar, so SQL code generated during ER diagram creation is not really usable to create the PostGres Tables via the Query Tool.

1. **Creating the database in PostGres.**

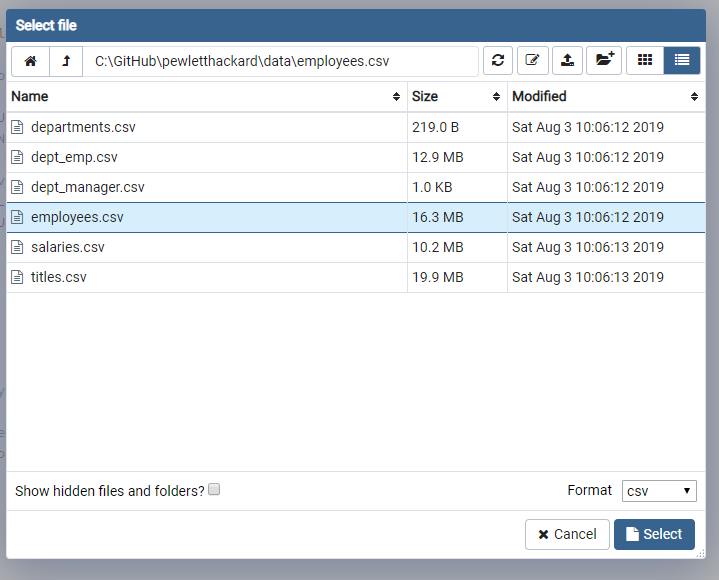
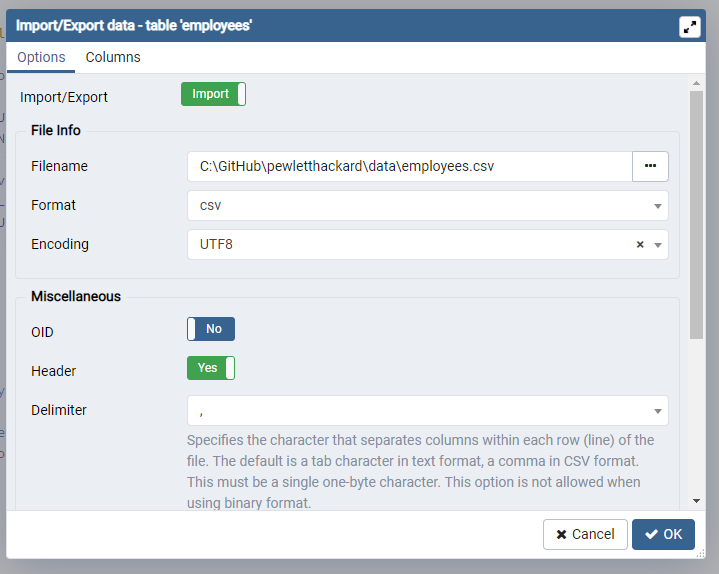
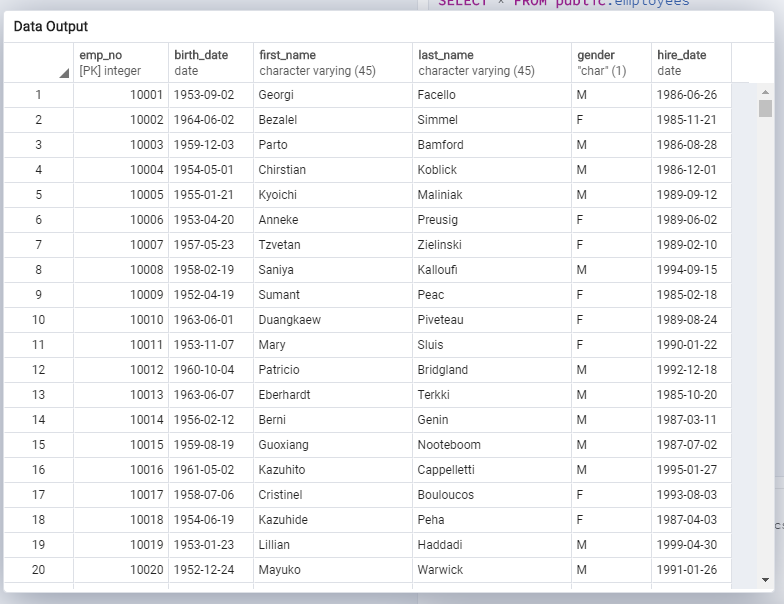
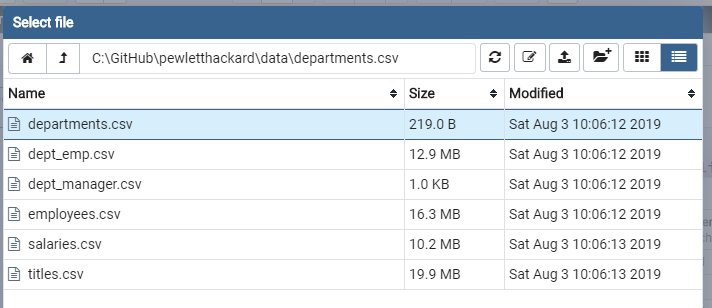
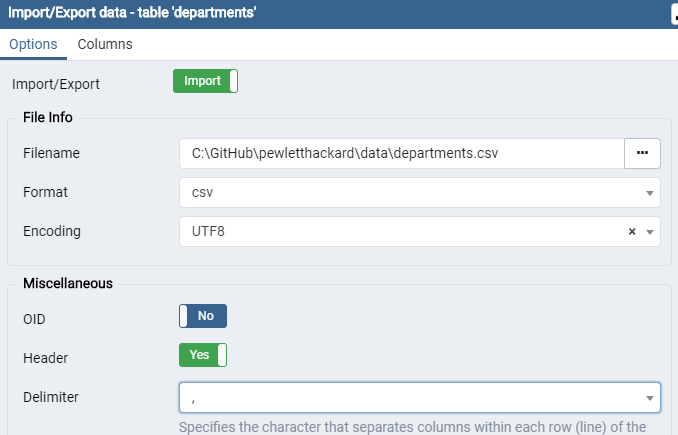
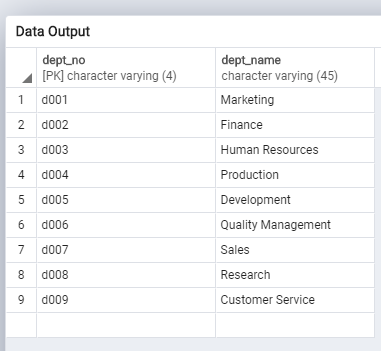
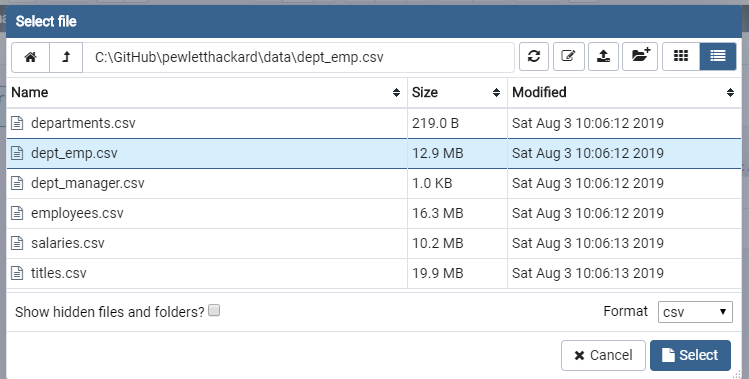
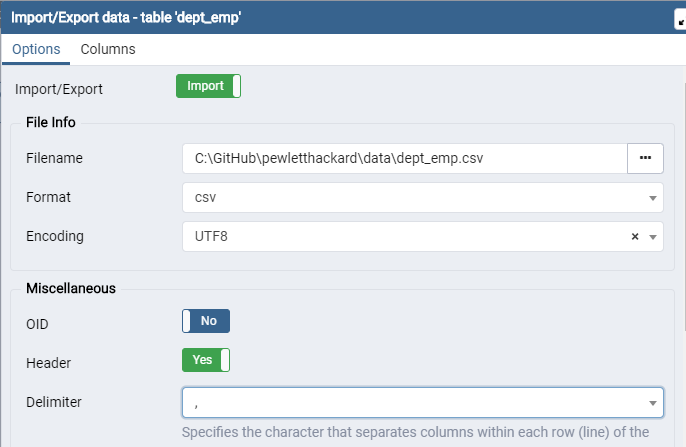
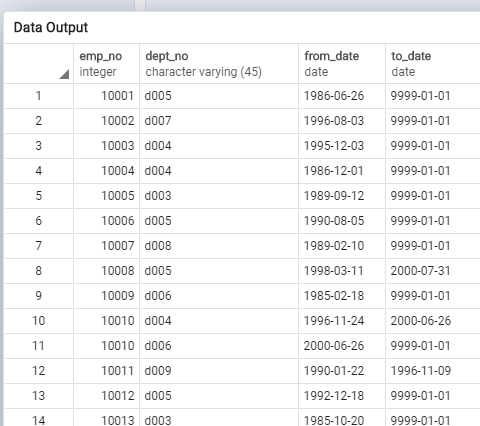
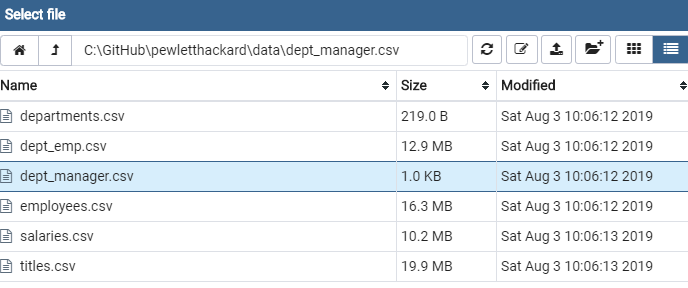
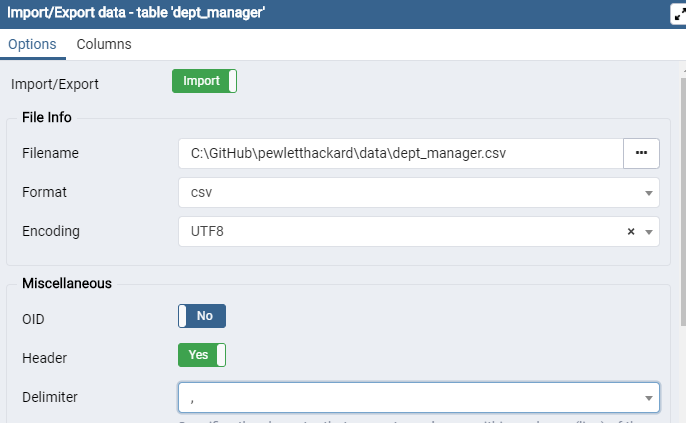


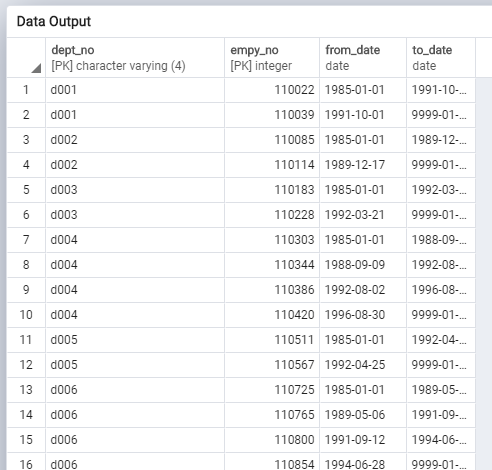
1. **Create SQL tables to house the required CSV files.**

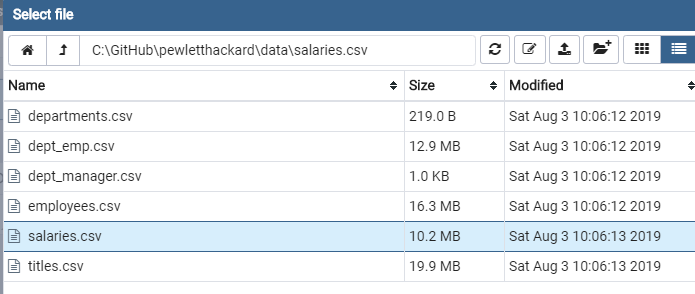
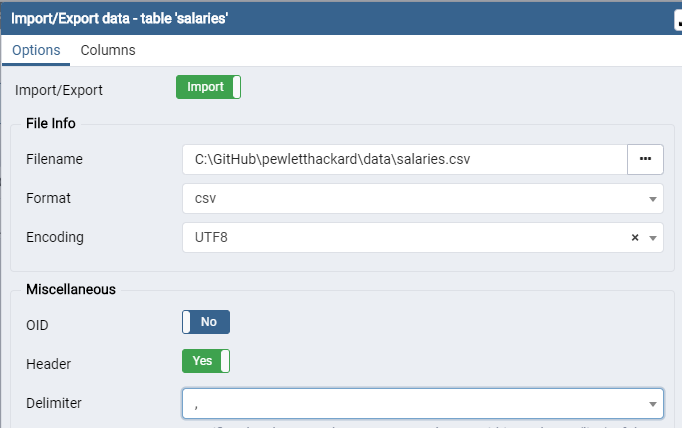
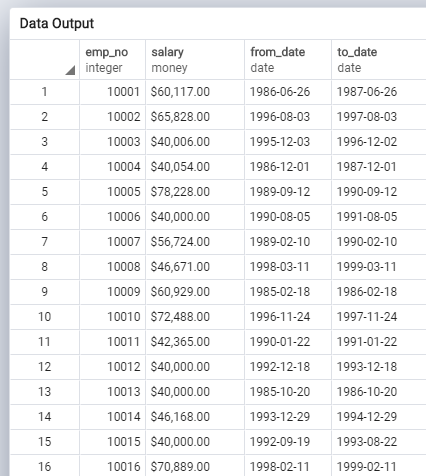
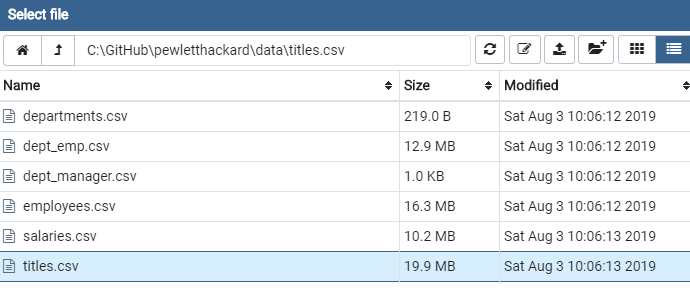
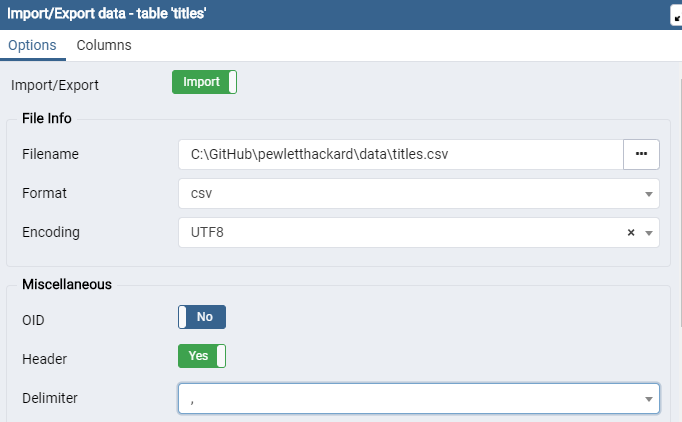
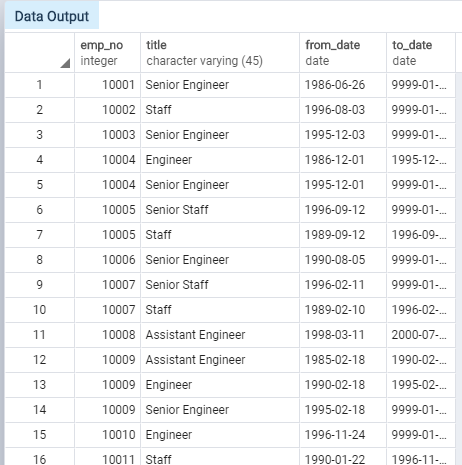
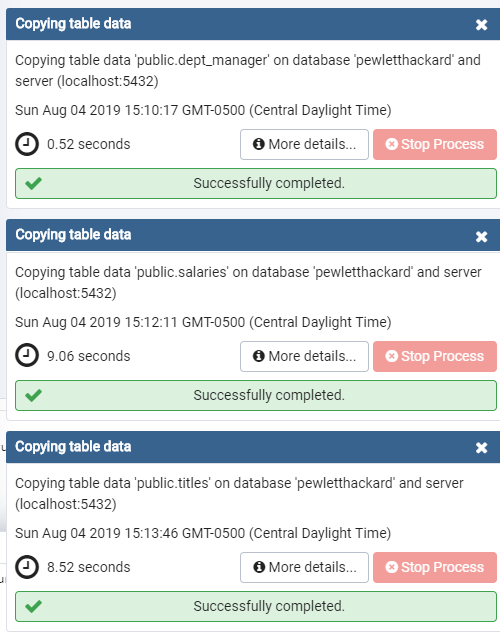
It’s important to note that the sequence of creating the SQL tables is critical in order to properly implement the usage of primary keys & foreign keys constraints during the table creation phase. While it’s possible to alter the tables, it’s always best to map out the proper table creation sequence prior to starting. Creating the ER diagram helps with identifying the proper sequence. You must create the main data tables first that have no foreign key constraints, so Employees & Departments table are created first. The auxiliary supporting tables or passthrough join tables can only be created after as they contain foreign keys which point to the primary data tables.

It’s also important to note that using a **standard naming convention with an underscore** shows that these tables are used as passthrough join tables. dept\_emp table naming implies that this is a join table between the departments table and the employees table with implies 1 department name to many employees. As opposed to a single named table as in employees which only contains employees.

1. Create the Employees table with the emp\_no as the primary key.
2. Create the departments table with the dept\_no as primary key.
3. Creating the salaries & titles tables, they do not have any primary keys but they do have foreign keys back into the employees table.
4. Creating the dept\_manager & dept\_emp tables have 2 foreign keys which reference the employees table and the departments table.
5. When possible, default values were set based on possible future data entries if project is deemed usable. A default hiredate in the employee table was set to CURRENTDATE. Many other date fields called to\_date were set with a default value of 9999-01-01 as a date in the future.
6. Below are the screenshots to document the import CSV file process…

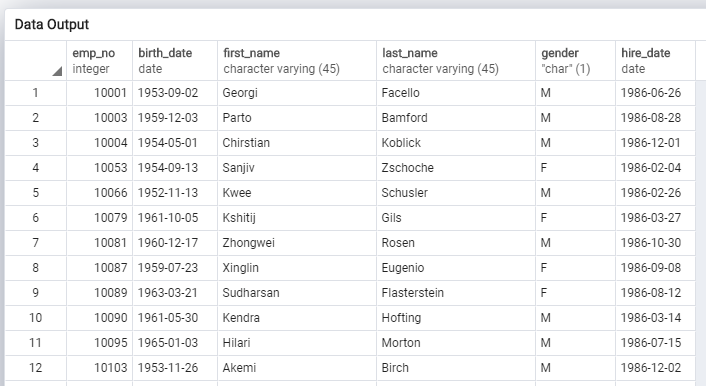
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  




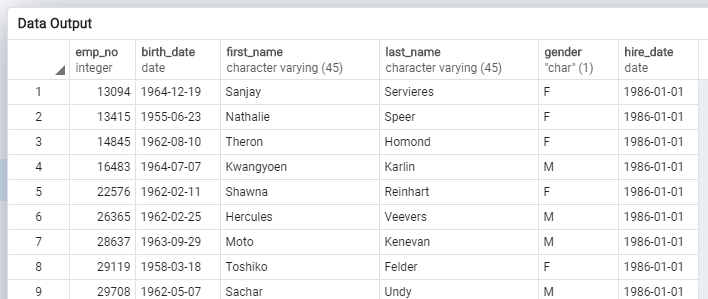
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**Data Analysis**  
Once we have completed the imports, we need to address the following:

**1.List the following details of each employee:   
employee number, last name, first name, gender, and salary.**  
select e.emp\_no, e.last\_name, e.first\_name, e.gender, s.salary from employees eleft join salaries s on s.emp\_no = e.emp\_no order by emp\_no  
  
**Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-01.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-01.csv)  
  


**2. List employees who were hired in 1986.**  
  
**The query with an order by employee number.**

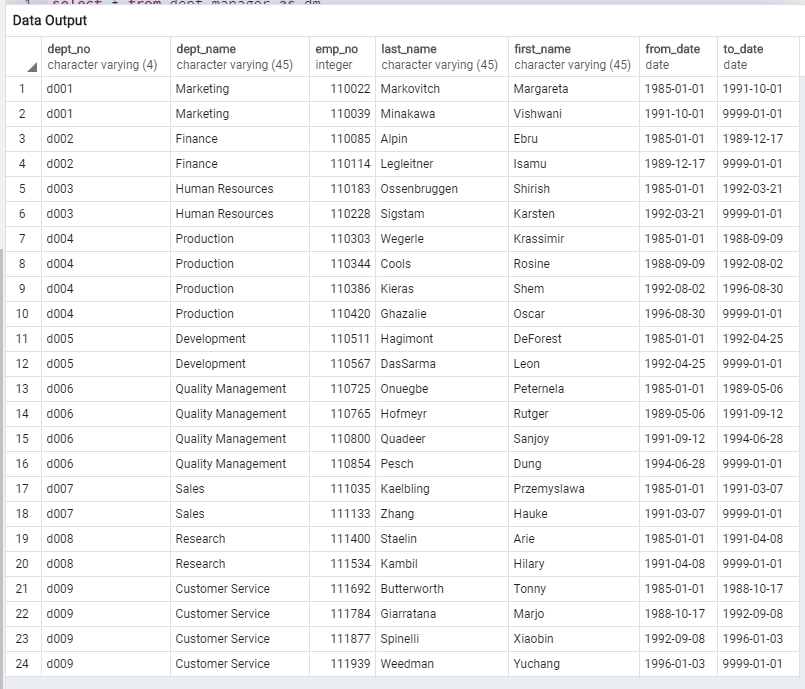
select \* from employees  
where date\_part('year',hire\_date) = 1986  
order by emp\_no  
  
**Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-02-Order-By-Emp-No.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-02-Order-By-Emp-No.csv) ****

or order by 1986 then month then day then employee number.

select \* from employees  
where date\_part('year',hire\_date) = 1986  
order by date\_part('month',hire\_date), date\_part('day',hire\_date),emp\_no   
 **Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-02-Order-By-1986-Month-Day-Emp\_No.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-02-Order-By-1986-Month-Day-Emp_No.csv)  
  


# **3. List the manager of each department with the following information: department number, department name, the manager's employee number, last name, first name, and start and end employment dates.**

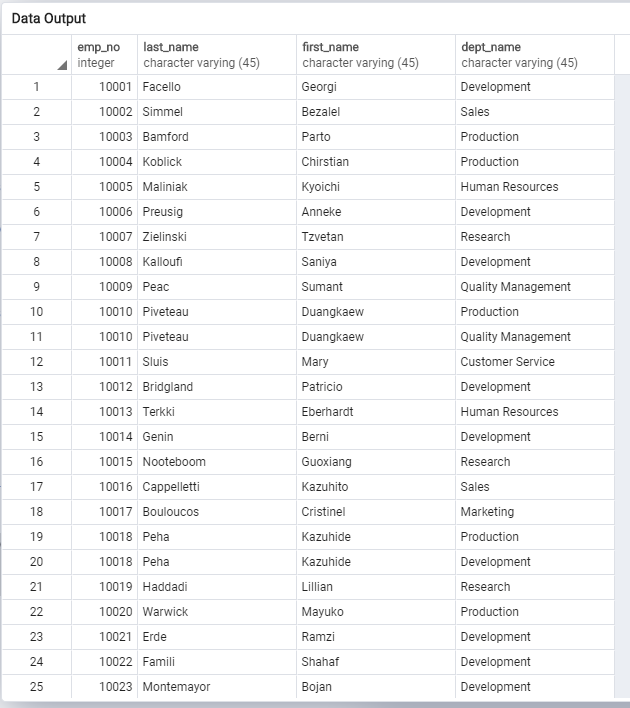
select d.dept\_no, d.dept\_name, e.emp\_no, e.last\_name, e.first\_name, dm.from\_date, dm.to\_date from dept\_manager as dm  
join employees e on dm.emp\_no = e.emp\_no  
join departments d on d.dept\_no = dm.dept\_no



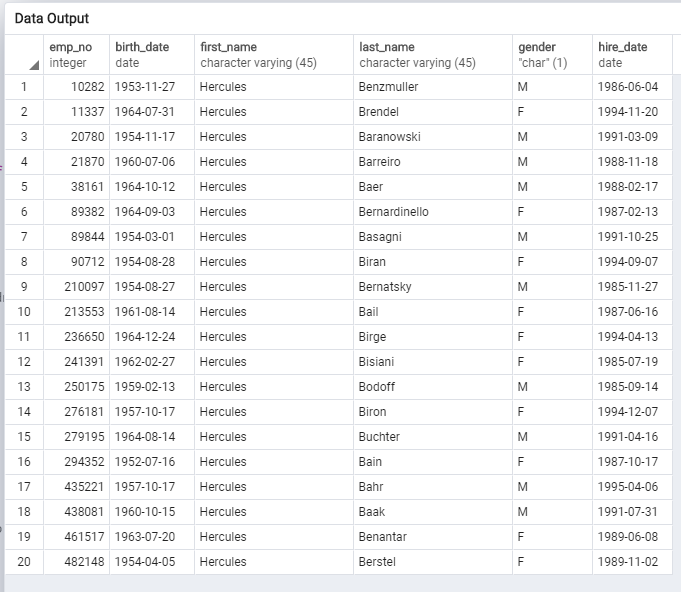
**Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-03-Dept-Managers.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-03-Dept-Managers.csv)

# **4. List the department of each employee with the following information: employee number, last name, first name, and department name.**

**\* Note several employees are possibly in multiple departments.**  
  
select e.emp\_no, e.last\_name, e.first\_name, d.dept\_name from dept\_emp as de  
join employees e on de.emp\_no = e.emp\_no  
join departments d on d.dept\_no = de.dept\_no  
order by emp\_no

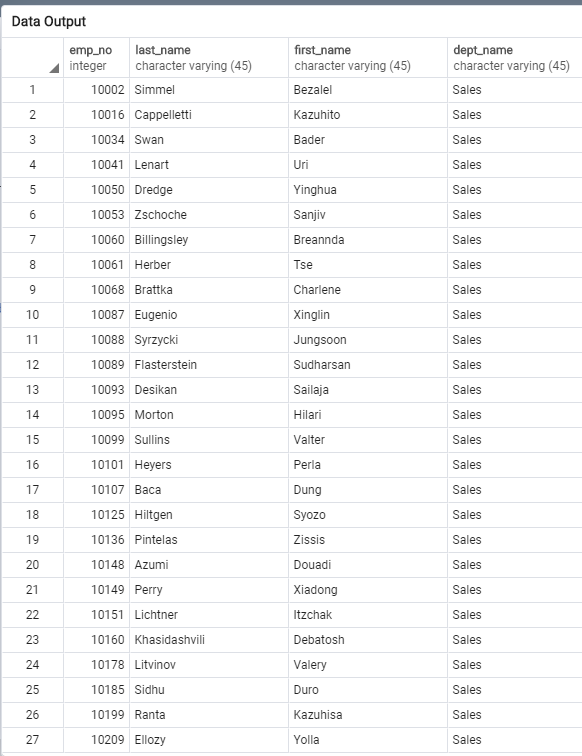


**Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-04-Dept-Employees.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-04-Dept-Employees.csv)

**5. List all employees whose first name is "Hercules" and last names begin with "B."**  
  
select \* from employees  
where first\_name = 'Hercules' and last\_name LIKE 'B%'  
order by emp\_no  
  
  
  
**Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-05-Hercules-B.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-05-Hercules-B.csv)

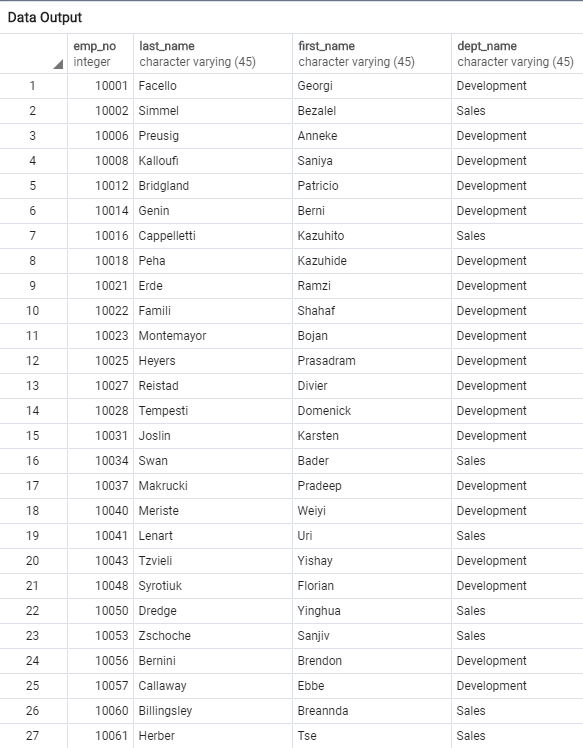
# **6. List all employees in the Sales department, including their employee number, last name, first name, and department name.**

**select e.emp\_no, e.last\_name, e.first\_name, d.dept\_name from dept\_emp as de  
join employees e on de.emp\_no = e.emp\_no  
join departments d on d.dept\_no = de.dept\_no  
where dept\_name = 'Sales'  
order by emp\_no**

  
  
**Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-06-Sales.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-06-Sales.csv)

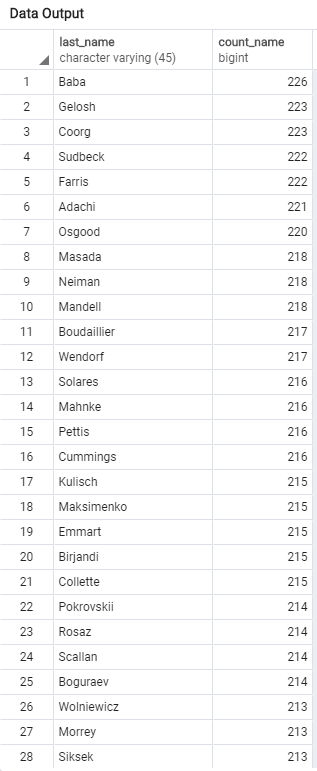
# **7. List all employees in the Sales and Development departments, including their employee number, last name, first name, and department name.**

**select e.emp\_no, e.last\_name, e.first\_name, d.dept\_name from dept\_emp as de  
join employees e on de.emp\_no = e.emp\_no  
join departments d on d.dept\_no = de.dept\_no  
where dept\_name in('Sales','Development')  
order by emp\_no**

  
  
**Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-07-Sales-Development.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-07-Sales-Development.csv)

# **8. In descending order, list the frequency count of employee last names, i.e., how many employees share each last name.**

**select last\_name, count(\*) as count\_name from employees  
group by last\_name  
order by count\_name desc**

  
  
**Download CSV File:** [**2019-08-Pewlett-Hackard-Data-Analysis-Question-08-Last\_Name\_Count\_Desc.csv**](https://github.com/BrianLabelle/pewletthackard/blob/master/data_analysis/2019-08-Pewlett-Hackard-Data-Analysis-Question-08-Last_Name_Count_Desc.csv)

# **BONUS:**

Employee Number : 499942.  
  
