Employee Database:

A Mystery in Two Parts

Introduction

Figure 1. Database Illustration

It is a beautiful spring day and two weeks since I have been hired as a new data engineer at ***Pewlett Hackard***. I am still getting my bearings in my new office when my boss called me to his office. *"Your first major task is a research project on employees of the corporation from the 1980s and 1990s. All that remain of the database of employees from that period are six CSV files"*, he said. He continued, as I vigorously took notes: *"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 – (A) Data Modeling (B) Data Engineering and finally (C) Data Analysis"*. I excused myself, excited about my new assignment – as I closed the door behind me, I heard him say – *"Good luck !!"*

Project planning:

After settling down with a cup of coffee, I decided to chalk out my next steps. The project shall be divided into following sections:

- **Data Modeling:** After inspecting the six CSV (comma-separated values) files, the ERD (entity relationship diagram) of the tables will be sketched out. An ERD shows the relationships between the entity sets in the database. Entities are characterized not only by the relationships, but also by additional properties (attributes) which include identifiers like Primary Key, Foreign Key and so on.

- **Data Engineering:** Using the information from the ERD sketch, we shall create a table schema for each of the six CSV files. The schema will specify data types, primary keys, foreign keys and other constraints (like NOT NULL). Each CSV file will be imported to corresponding SQL tables. Now we have the database ready for executing SQL queries.

- **Data Analysis:** In the analysis section, we shall execute a list of queries and the record the output tables. We shall use ***PostgreSQL***, an open-source object-relational database system along with ***pgAdmin*** which is the most popular and feature rich open-source administration and development platform for PostgreSQL for this project.

Data Modeling: Quick Database Diagrams ***(QuickDBD)*** is a simple online tool to quickly draw database diagrams by typing. The following entity-relationship model was sketched using ***QuickDBD*** tool.

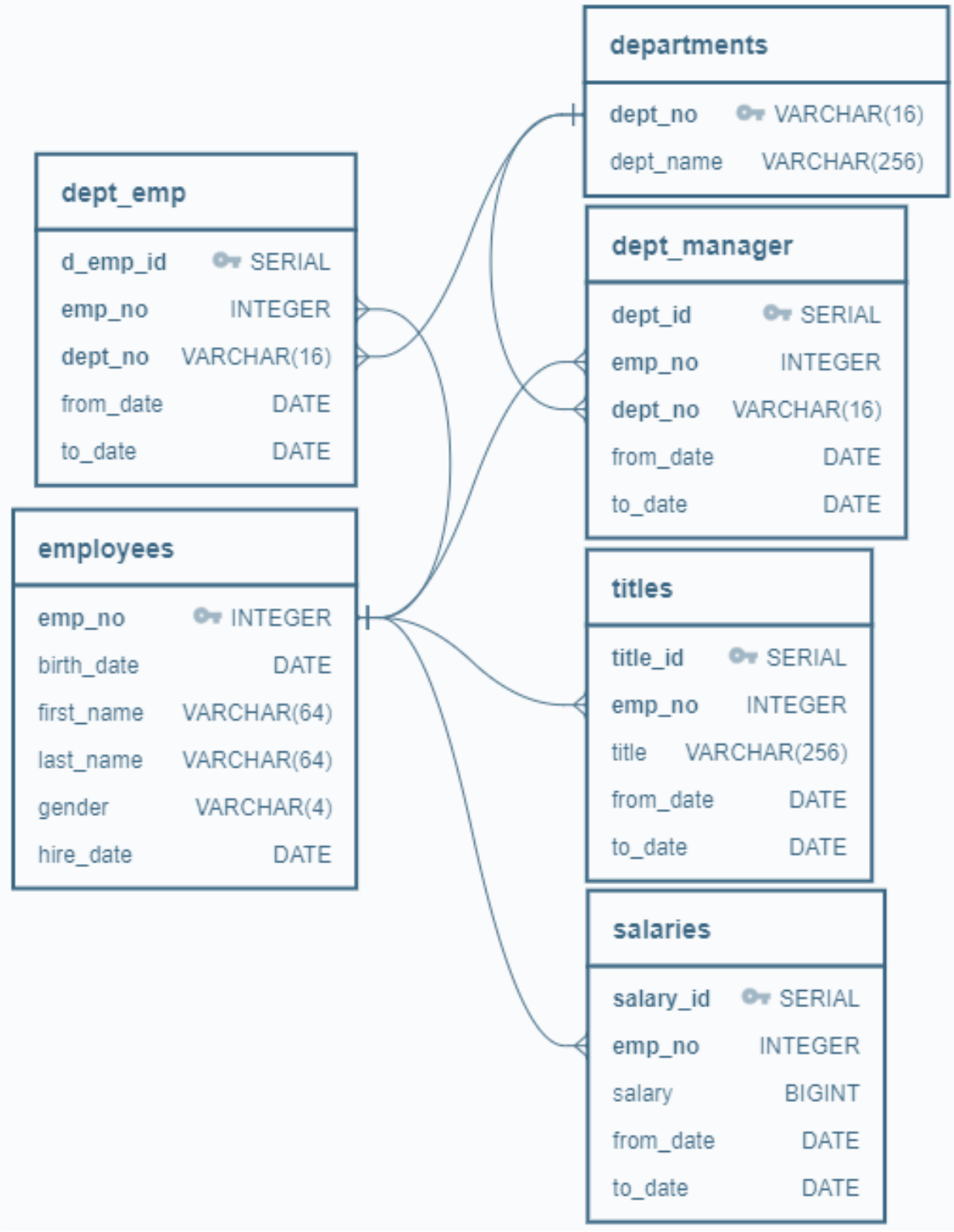

Figure 2. ERD of the database using QuickDBD tool

Lucidchart is the web's leading visual workspace that combines diagramming, data visualization and cross-platform collaboration. As shown in Fig.3, the entities, their relationships and cardinality were drawn using ***Lucidchart***.

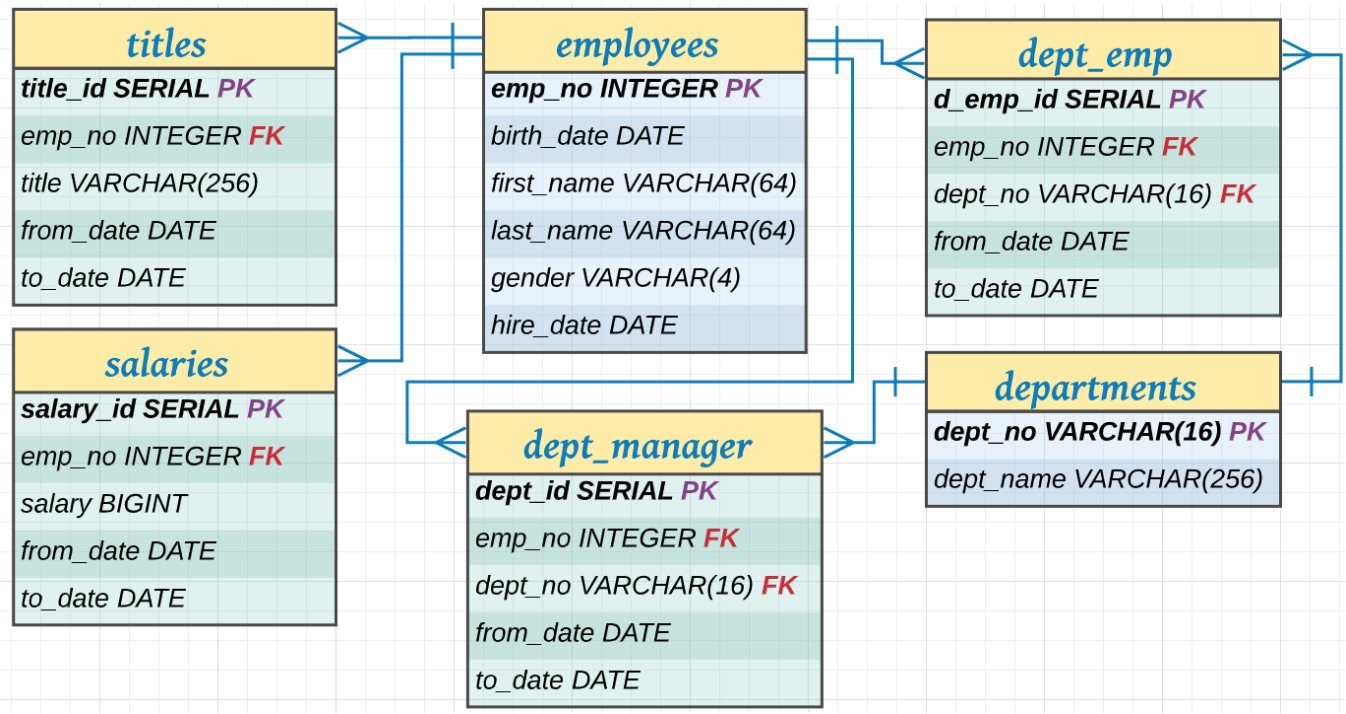

Figure 3. ERD of the database using Lucidchart tool

Data Engineering:

With the ERD in hand, now we are ready to create the database schema. The schema is the structure of the database, a blueprint that describes how the database is constructed (i.e. definition of the tables in this case) along with the integrity constraints. These integrity constraints ensure compatibility between the parts of the database. A new database, ***Pewlett_Hackard_db***, was created before executing the following schema snippet to create employees table:

```
DROP TABLE IF EXISTS employees;
CREATE TABLE employees ( emp_no INTEGER PRIMARY KEY, birth_date DATE NOT NULL,
first_name VARCHAR(64) NOT NULL, last_name VARCHAR(64) NOT NULL, gender VAR-
CHAR(4) NOT NULL, hire_date DATE NOT NULL);
```

The schema for all the six tables is provided in Appendix A. Each CSV file was imported into the corresponding SQL table. Now we have a complete database with six tables of data along with their relationships and constraints regarding the employees in Pewlett Hackard.

Data Analysis:

Queries: With the database in hand, we are ready to execute the initial set of queries as decided by my manager. ***The SQL commands of the queries are provided in Appendix B.*** In this section we shall discuss the outputs obtained from the queries.

#1. List the following details of each employee: employee number, last name, first name, gender, and salary.

	Employee Number integer	Last Name character varying (60)	First Name character varying (60)	Gender character varying (4)	Salary bigint
1	10005	Maliniak	Kyoichi	M	78228
2	10010	Piveteau	Duangkaew	F	72488
3	10011	Sluis	Mary	F	42365
4	10013	Terkki	Eberhardt	M	40000
5	10017	Bouloucos	Cristinel	F	71380
6	10035	Chappelet	Alain	M	41538
7	10037	Makrucki	Pradeep	M	40000
8	10045	Shanbhogue	Moss	M	41971
9	10051	Caine	Hidefumi	M	48817
10	10058	McFarlin	Berhard	M	52787
11	10078	Mondadori	Danel	F	47280
12	10085	Malabarba	Kenroku	M	40000

Figure 4. Output table from Query #1

Fig. 4 shows the first 13 rows of the output table from query #1

#2. List employees who were hired in 1986.

Fig. 5 shows the first 13 rows of the output table from query #2

#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.

	Employee Number integer	Last Name character varying (60)	First Name character varying (60)
1	10001	Facello	Georgi
2	10003	Bamford	Parto
3	10004	Koblick	Chirstian
4	10053	Zschoche	Sanjiv
5	10066	Schusler	Kwee
6	10079	Gils	Kshitij
7	10081	Rosen	Zhongwei
8	10087	Eugenio	Xinglin
9	10089	Flasterstein	Sudharsan
10	10090	Hofting	Kendra
11	10095	Morton	Hilari
12	10103	Birch	Akemi
13	10108	Giveon	Lunjin

Figure 5. Output table from Query #2

Fig. 6 shows the first 18 rows of the output table from query #3

	Department Number character varying (10)	Department Name character varying (255)	Manager employee number integer	Last Name character varying (60)	First Name character varying (60)	Start Date date	End Date date
1	d001	Marketing	110022	Markovitch	Margareta	1985-01-01	1991-10-01
2	d001	Marketing	110039	Minakawa	Vishwani	1991-10-01	9999-01-01
3	d002	Finance	110085	Alpin	Ebru	1985-01-01	1989-12-17
4	d002	Finance	110114	Legleitner	Isamu	1989-12-17	9999-01-01
5	d003	Human Resources	110183	Ossenbruggen	Shirish	1985-01-01	1992-03-21
6	d003	Human Resources	110228	Sigstam	Karsten	1992-03-21	9999-01-01
7	d004	Production	110303	Wegerle	Krassimir	1985-01-01	1988-09-09
8	d004	Production	110344	Cools	Rosine	1988-09-09	1992-08-02
9	d004	Production	110386	Kieras	Shem	1992-08-02	1996-08-30
10	d004	Production	110420	Ghazalie	Oscar	1996-08-30	9999-01-01
11	d005	Development	110511	Hagimont	DeForest	1985-01-01	1992-04-25
12	d005	Development	110567	DasSarma	Leon	1992-04-25	9999-01-01
13	d006	Quality Management	110725	Onuegbe	Peternela	1985-01-01	1989-05-06
14	d006	Quality Management	110765	Hofmeyr	Rutger	1989-05-06	1991-09-12
15	d006	Quality Management	110800	Quadeer	Sanjoy	1991-09-12	1994-06-28
16	d006	Quality Management	110854	Pesch	Dung	1994-06-28	9999-01-01
17	d007	Sales	111035	Kaelbling	Przemyslawa	1985-01-01	1991-03-07
18	d007	Sales	111133	Zhang	Hauke	1991-03-07	9999-01-01

Figure 6. Output table from Query #3

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

Fig. 7 shows the first 13 rows of the output table from query #4.

	Employee Number integer	Last Name character varying (60)	First Name character varying (60)	Department Name character varying (255)
1	10005	Maliniak	Kyoichi	Human Resources
2	10010	Piveteau	Duangkaew	Production
3	10010	Piveteau	Duangkaew	Quality Management
4	10011	Sluis	Mary	Customer Service
5	10013	Terkki	Eberhardt	Human Resources
6	10017	Bouloucos	Cristinel	Marketing
7	10035	Chappelet	Alain	Production
8	10037	Makrucki	Pradeep	Development
9	10045	Shanbhogue	Moss	Production
10	10051	Caine	Hidefumi	Production
11	10058	McFarlin	Berhard	Marketing
12	10078	Mondadori	Danel	Development
13	10085	Malabarba	Kenroku	Production

Figure 7. Output table from Query #4

#5. List all employees whose first name is "Hercules" and last names begin with "B."

Fig. 8 shows the first 13 rows of the output table from query #5.

	Employee Number integer	Last Name character varying (60)	First Name character varying (60)
1	10282	Benzmuller	Hercules
2	11337	Brendel	Hercules
3	20780	Baranowski	Hercules
4	21870	Barreiro	Hercules
5	38161	Baer	Hercules
6	89382	Bernardinello	Hercules
7	89844	Basagni	Hercules
8	90712	Biran	Hercules
9	210097	Bernatsky	Hercules
10	213553	Bail	Hercules
11	236650	Birge	Hercules
12	241391	Bisiani	Hercules
13	250175	Bodoff	Hercules

Figure 8. Output from Query #5

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

Fig. 9 shows the first 13 rows of the output table from query #6

	Employee Number integer	Last Name character varying (60)	First Name character varying (60)	Department Name character varying (255)
1	10002	Simmel	Bezalel	Sales
2	10016	Cappelletti	Kazuhito	Sales
3	10034	Swan	Bader	Sales
4	10041	Lenart	Uri	Sales
5	10050	Dredge	Yinghua	Sales
6	10053	Zschoche	Sanjiv	Sales
7	10060	Billingsley	Breannda	Sales
8	10061	Herber	Tse	Sales
9	10068	Brattka	Charlene	Sales
10	10087	Eugenio	Xinglin	Sales
11	10088	Syrzycki	Jungsoon	Sales
12	10089	Flasterstein	Sudharsan	Sales
13	10093	Desikan	Sailaja	Sales

Figure 9. Output from Query #6

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

Fig. 10 shows the first 13 rows of the output table from query #7

	Employee Number integer	Last Name character varying (60)	First Name character varying (60)	Department Name character varying (255)
1	10001	Facello	Georgi	Development
2	10002	Simmel	Bezalel	Sales
3	10006	Preusig	Anneke	Development
4	10008	Kalloufi	Saniya	Development
5	10012	Bridgland	Patricio	Development
6	10014	Genin	Berni	Development
7	10016	Cappelletti	Kazuhito	Sales
8	10018	Peha	Kazuhide	Development
9	10021	Erde	Ramzi	Development
10	10022	Famili	Shahaf	Development
11	10023	Montemayor	Bojan	Development
12	10025	Heyers	Prasadram	Development
13	10027	Reistad	Divier	Development

Figure 10. Output from Query #7

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

	Last Name character varying (60)	Frequency Count bigint
1	Baba	226
2	Coorg	223
3	Gelosh	223
4	Farris	222
5	Sudbeck	222
6	Adachi	221
7	Osgood	220
8	Masada	218
9	Neiman	218
10	Mandell	218
11	Boudaillier	217
12	Wendorf	217
13	Cummings	216

Figure 11. Output from Query #8

One interesting observation from query #8:

Only one employee with last name: ***Foolsday***

With this interesting database, I decided to investigate further and continued with some additional queries.

Additional Queries: *The SQL commands of the queries are provided in Appendix C.* In this section we shall discuss the outputs obtained from the queries.

#9. How long have the managers worked in their current role?

Fig. 12 shows the total employment years and the years they acted as managers for 18 employees. For example, ***Legleitner Isamu*** was manager for 31 years out of his 35 years of service in the company.

	Last Name character varying (60)	First Name character varying (60)	Employment Years double precision	Managerial Years double precision
1	Markovitch	Margareta	35	6
2	Minakawa	Vishwani	34	29
3	Alpin	Ebru	35	4
4	Legleitner	Isamu	35	31
5	Ossenbruggen	Shirish	35	7
6	Sigstam	Karsten	34	28
7	Wegerle	Krassimir	35	3
8	Cools	Rosine	34	4
9	Kieras	Shem	31	4
10	Ghazalie	Oscar	28	24
11	Hagimont	DeForest	35	7
12	DasSarma	Leon	33	28
13	Onuegbe	Peternela	35	4
14	Hofmeyr	Rutger	31	2
15	Quadeer	Sanjoy	33	3
16	Pesch	Dung	31	26
17	Kaelbling	Przemyslawa	35	6
18	Zhang	Hauke	33	29

Figure 12. Output from Query #9

#10. How many employees per title and on average how long do they work per title?

Fig. 13 shows the number of employees per job title and the average number of years performing that role. We see employees work as ***"technique leader"*** the longest (23 years) while maximum number of employees (115003) are hired as ***"engineer"***. There are 24 managers with average years of 13 years performing in that role.

	title character varying (255)	Number of employees bigint	Avg. Title Years double precision
1	Technique Leader	15159	23
2	Senior Engineer	97750	22
3	Senior Staff	92853	22
4	Manager	24	13
5	Assistant Engineer	15128	10
6	Engineer	115003	10
7	Staff	107391	10

Figure 13. Output from Query #10

#11. What is the average salary per title?

Fig. 14 shows the average annual salary of the employees per job title. As we can see that average is highest for Senior Staff followed by Staff. Though average salary is skewed by outliers, nevertheless, this table provides a valuable insight.

	Department Title character varying (255)	No. of employees bigint	Average Salary numeric
1	Senior Staff	92853	58503.29
2	Staff	107391	58465.27
3	Manager	24	51531.04
4	Technique Leader	15159	48580.51
5	Engineer	115003	48539.78
6	Senior Engineer	97750	48506.75
7	Assistant Engineer	15128	48493.20

Figure 14. Output from Query #11

#12. Do male managers get more salary on average than female managers?

Being a curious data scientist, I wanted to see if there is a gender bias in the average manager's salary.

Fig. 15 shows that female managers on average stick to the job three years lesser as well as receive about 20% lower salary than their male counterparts.

	Gender character varying (4)	Average Salary numeric	Avg. Managerial Years double precision
1	F	46662.38	12
2	M	57284.91	15

Figure 15. Output from Query #12

#13. Do male employees get more salary per year on average than their female counterparts?

As we can see from Fig. 15, there are about ***120K female*** employees and ***180K male*** employees. Even though there is difference in the maximum salary (maximum salary of female employee about ***6K dollars*** less), we see ***no difference in the minimum salary***. Among all the employees the average salary of male employees is quite similar to the average salary of the female employees.

There is ***no gender bias in the average annual salary*** of the employees of Pewlett Hackard.

	Gender character varying (4)	No of Employees bigint
1	F	120051
2	M	179973

	Gender character varying (4)	Maximum Salary bigint
1	F	123477
2	M	129492

	Gender character varying (4)	Minimum Salary bigint
1	F	40000
2	M	40000

	Gender character varying (4)	Average Salary numeric
1	F	52953.84
2	M	52982.00

Figure 16. Output from Query #13

#14. List of oldest hire among employees

Fig. 17A lists the ***last name, first name, gender, age, employment years, department name and salary*** of employees ***sorted by age***, last name and first name in decreasing order. As we can see oldest employees in the company are 68 years old with employment years ranging from 35 to 22 years.

	Last Name character varying (60)	First Name character varying (60)	Gender character varying (4)	Age double precision	Employment Years double precision	Department Name character varying (255)	Salary bigint
1	Aamodt	Ashish	F	68	28	Production	40000
2	Aamodt	Geoffry	M	68	25	Production	75659
3	Aamodt	Geoffry	M	68	25	Research	75659
4	Aamodt	Mani	M	68	28	Development	40000
5	Aamodt	Mats	F	68	30	Marketing	90455
6	Aamodt	Moto	F	68	34	Research	64792
7	Aamodt	Moto	F	68	34	Development	64792
8	Acton	Aimee	F	68	22	Production	59824
9	Acton	Iara	F	68	25	Development	47536
10	Acton	Kristen	F	68	35	Marketing	52117

Figure 17A. Output from Query #14 (oldest by age)

Fig. 17B lists the ***last name, first name, gender, age, employment years, department name and salary*** of employees ***sorted by employment years***, last name and first name in decreasing order. As we can see maximum years any employee has contributed to the company is 35 years.

	Last Name character varying (60)	First Name character varying (60)	Gender character varying (4)	Age double precision	Employment Years double precision	Department Name character varying (255)	Salary bigint
1	Aamodt	Aluzio	M	61	35	Development	41789
2	Aamodt	Conal	F	55	35	Finance	62788
3	Aamodt	Fuqing	M	62	35	Development	62378
4	Aamodt	Gian	M	60	35	Sales	81659
5	Aamodt	Hidefumi	M	62	35	Marketing	52284
6	Aamodt	Luisa	F	59	35	Sales	49358
7	Aamodt	Magy	M	64	35	Development	40000
8	Aamodt	Peternela	M	55	35	Production	52208
9	Aamodt	Rimli	M	63	35	Sales	78573
10	Aamodt	Sigeru	M	67	35	Development	40000

Figure 17B. Output from Query #14 (oldest by employment years)

As I examine the data, I have a creeping suspicion that the dataset is fake. I surmise that my boss handed me spurious data in order to test my data engineering skills. To confirm my hunch, I decide to create some visualizations of the data so that I have concrete proof before I confront my boss.

Data Visualization:

To visualize the data, I decided to import the SQL database into Pandas.

Engine Connection: The engine, which is the starting point for any SQLAlchemy application, was created for our PostgreSQL database by issuing the following call

engine = create_engine('postgresql+psycopg2://'+ Username + ':' + Password + '@localhost:5432/' + DBname)

connection = engine.connect()

This creates a Dialect object (*"postgresql + psycopg2"*) as well as a Pool object which establishes a DBAPI connection at localhost:5432.

All the codes are available in a Jupyter notebook: ***SQLAlchemy_visualization.ipynb***. This report summarizes the visualizations obtained for each query related to the database.

#1. Histogram of Salaries

Fig. 18 shows a histogram to visualize the common salary ranges of all the employees. The median salary (~ $49K) is marked as dashed blue line. As we can see majority of the employees (more than 120,000) earn between $40K- $44K per year

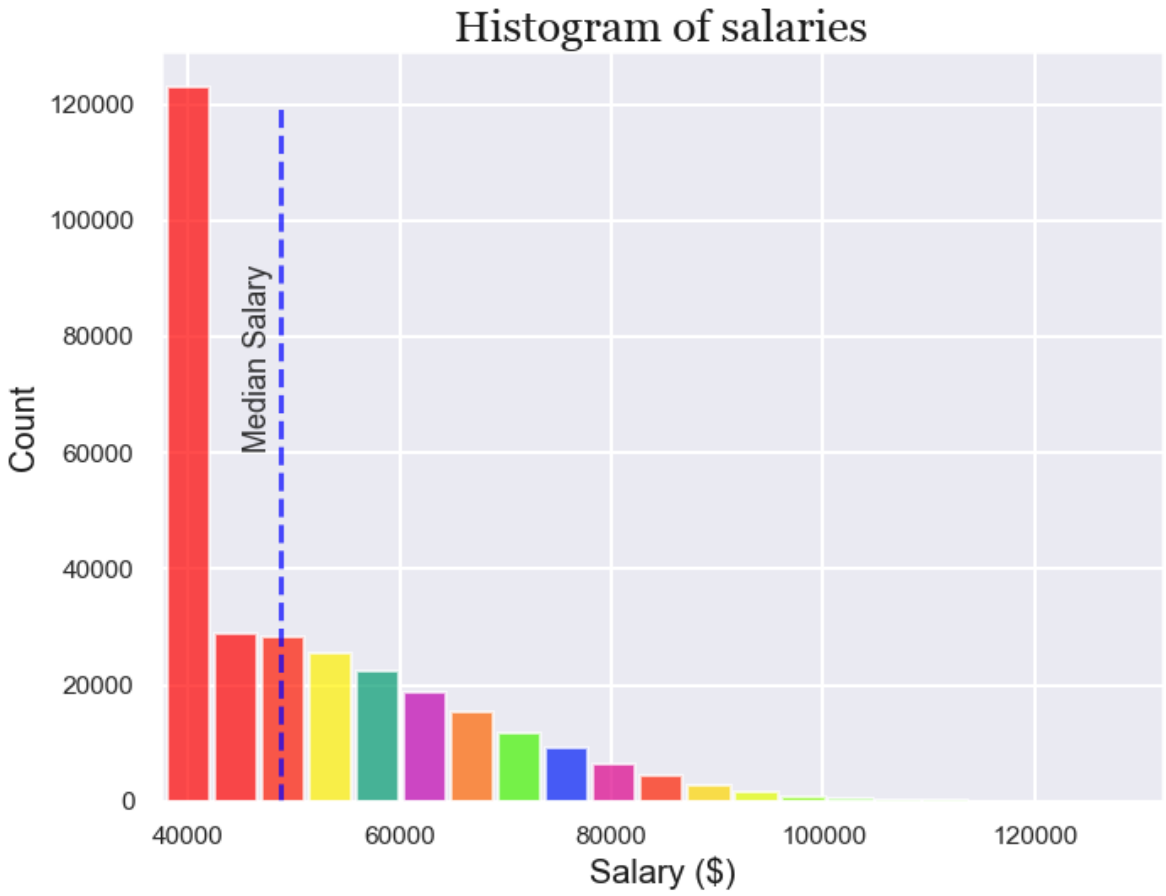

Figure 18. Histogram of salaries of all the employees

#2. Bar chart of average salary by title

Fig. 19 shows a bar chart of average annual salary for each job title in the company. The average salary is quite uniform across all the titles with ***"Senior Staff"*** and ***"Staff"*** earning the highest numbers.

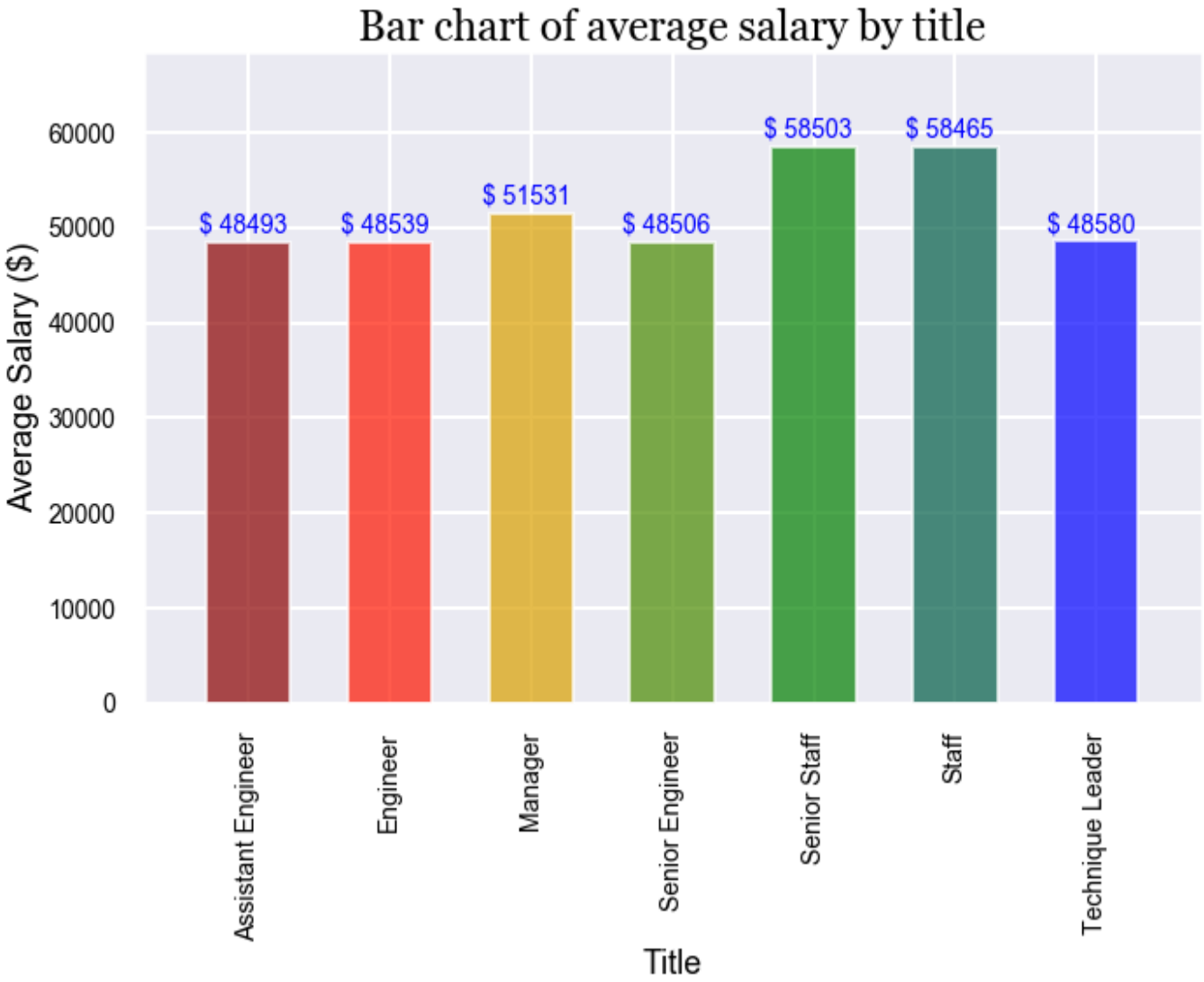

Figure 19. Bar Chart of Average Salary by Title

#3. Pie chart of male and female employees

Fig. 20 shows a pie chart of male and female employees in the company. Among 300,024 employees, 179,973 (60%) employees are male as compared to 120,051 (40%) employees are female.

#4. Scatter Plot of Median Salary vs. Employment Years

Does the median salary depend on the number of years an employee worked in the company?

Fig. 21 shows a scatter plot of the median salary with the employment years. We have selected median salary because mean salary would be susceptible to outliers.

The median salary is unrelated to years employed in the company.

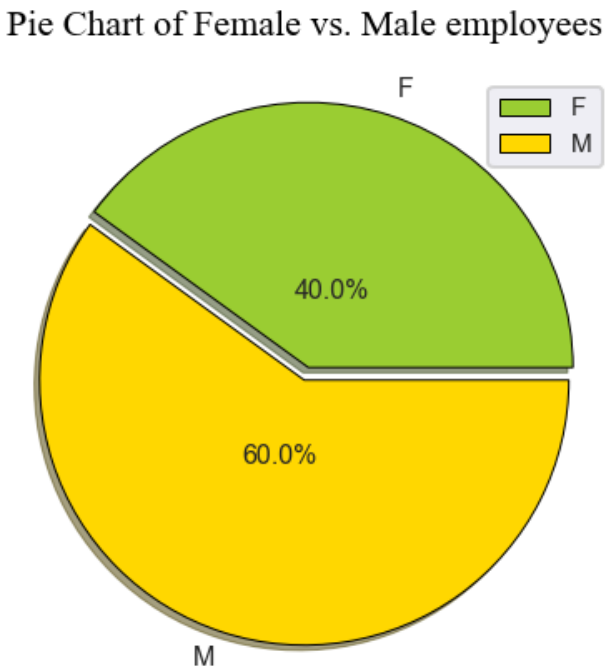

Figure 20. Pie chart based on employee gender

Figure 21. Scatter plot of median salary vs. employment years

#5. Box Plot of Salaries from different Titles

Fig. 22 shows a box plot of salaries with outliers for different employee titles. As we can see, every title except Manager has a large number of outliers.

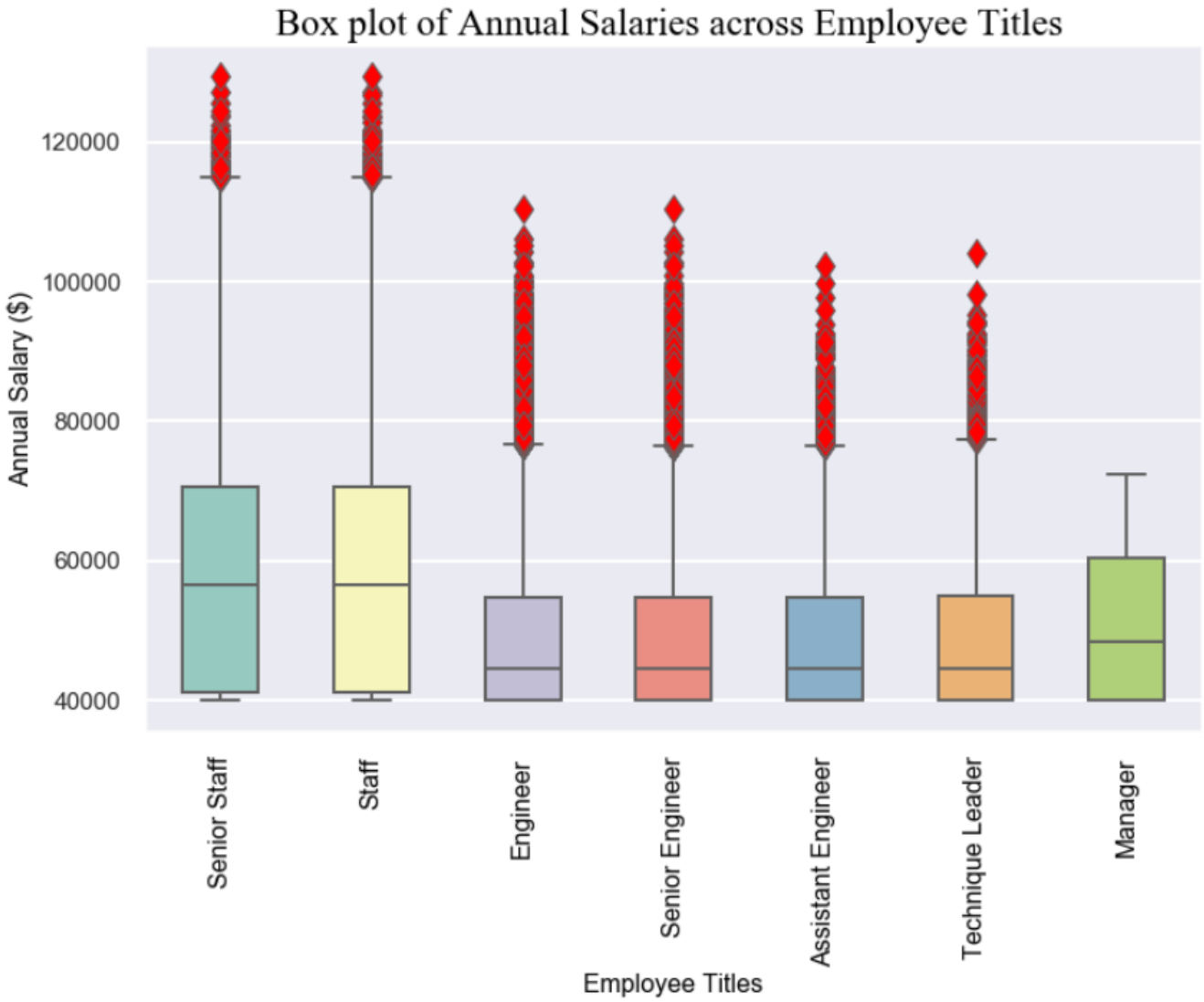

Figure 22. Box plot and outliers shown for salary vs. employee title

Conclusion:

Evidence in hand, I march into my boss's office and present the visualization. With a sly grin, my boss thanks me for my work. On my way out of the office, I hear him say "Search your ID number". I look down in surprise to see my employee ID # 499942.

APPENDIX A

```
**********************************************************************
-- Using the information from the ERD table here is a table schema for each of the six
CSV files.
--**** Creating schema for employees table *********
DROP TABLE IF EXISTS employees;
CREATE TABLE employees (
        emp_no INTEGER PRIMARY KEY,
        birth_date DATE NOT NULL,
    first_name VARCHAR(64) NOT NULL,
        last_name VARCHAR(64) NOT NULL,
    gender VARCHAR(4) NOT NULL,
        hire_date DATE NOT NULL);
--**** Viewing the schema for employees table *******
SELECT * FROM employees;

--**** Creating schema for departments table *********
DROP TABLE IF EXISTS departments;
CREATE TABLE departments (
        dept_no VARCHAR(16) PRIMARY KEY NOT NULL,
        dept_name VARCHAR(256) NOT NULL);
---**** Viewing the schema for departments table *******
SELECT * FROM departments;

--**** Creating schema for titles table *********
DROP TABLE IF EXISTS titles;
CREATE TABLE titles (
        emp_no INTEGER NOT NULL,
        title VARCHAR(256) NOT NULL,
        from_date DATE NOT NULL,
        to_date DATE NOT NULL,
        FOREIGN KEY (emp_no) REFERENCES employees(emp_no));
--**** Viewing the schema for titles table *******
SELECT * FROM titles;
```

```
--**** Creating schema for salaries table *********
DROP TABLE IF EXISTS salaries;
CREATE TABLE salaries (
        emp_no INTEGER NOT NULL,
        salary BIGINT NOT NULL,
        from_date DATE NOT NULL,
        to_date DATE NOT NULL,
        FOREIGN KEY (emp_no) REFERENCES employees(emp_no));
--****Viewing the schema for salaries table *******
SELECT * FROM salaries;

--**** Creating schema for dept_emp table *********
DROP TABLE IF EXISTS dept_emp;
CREATE TABLE dept_emp (
        emp_no INTEGER NOT NULL,
        dept_no VARCHAR(16) NOT NULL,
        from_date DATE NOT NULL,
        to_date DATE NOT NULL,
        FOREIGN KEY (emp_no) REFERENCES employees(emp_no),
        FOREIGN KEY (dept_no) REFERENCES departments(dept_no));
--**** Viewing the schema for dept_emp table *******
SELECT * FROM dept_emp;

--**** Creating schema for dept_manager table *********
DROP TABLE IF EXISTS dept_manager;
CREATE TABLE dept_manager (
        dept_no VARCHAR(16) NOT NULL,
        emp_no INTEGER NOT NULL,
        from_date DATE NOT NULL,
        to_date DATE NOT NULL,
        FOREIGN KEY (emp_no) REFERENCES employees(emp_no),
        FOREIGN KEY (dept_no) REFERENCES departments(dept_no));
--****Viewing the schema for dept_manager table *******
SELECT * FROM dept_manager;
```

APPENDIX B

```
-- ***************************************************************************
-- 1. List the following details of each employee: employee number, last name, first name,
--   gender, and salary.
-- ***************************************************************************
-- View information from employees table
SELECT * FROM employees LIMIT 10;
-- View information from salaries table
SELECT * FROM salaries LIMIT 10;
-- Create the query
SELECT e.emp_no AS "Employee Number",
            e.last_name AS "Last Name",
      e.first_name AS "First Name",
            e.gender AS "Gender",
            s.salary AS "Salary"
FROM employees AS e
INNER JOIN salaries AS s
ON e.emp_no = s.emp_no;
--***************************************************************************
-- 2. List employees who were hired in 1986.
-- ***************************************************************************
SELECT e.emp_no AS "Employee Number",
            e.last_name AS "Last Name",
            e.first_name AS "First Name"
FROM employees AS e
WHERE EXTRACT(YEAR FROM hire_date) = 1986;

--***************************************************************************
-- 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.
--***************************************************************************
-- View information from dept_manager table
SELECT * FROM dept_manager LIMIT 10;
-- View information from departments table
SELECT * FROM departments LIMIT 10;
-- View information from employees table
SELECT * FROM employees LIMIT 10;
-- Create the query
SELECT m.dept_no AS "Department Number",
```

```
        d.dept_name AS "Department Name",
        m.emp_no AS " Manager employee number",
        e.last_name AS "Last Name",
        e.first_name AS "First Name",
        m.from_date AS "Start Date",
        m.to_date AS "End Date"
FROM employees AS e
        INNER JOIN dept_manager AS m
        ON e.emp_no = m.emp_no
                INNER JOIN departments AS d
                ON m.dept_no = d.dept_no;

--*************************************************************************************
-- 4. List the department of each employee with the following information: employee number, last
name, -- first name, and department name.
-- *************************************************************************************
-- Create the query ---
SELECT e.emp_no AS "Employee Number",
        e.last_name AS "Last Name",
        e.first_name AS "First Name",
        d.dept_name AS "Department Name"
FROM employees AS e
        INNER JOIN dept_emp AS de
        ON e.emp_no = de.emp_no
                INNER JOIN departments AS d
                ON de.dept_no = d.dept_no;

--*************************************************************************************
-- 5. List all employees whose first name is "Hercules" and last names begin with "B."
-- *************************************************************************************
-- Create the query ---
SELECT e.emp_no AS "Employee Number",
        e.last_name AS "Last Name",
        e.first_name AS "First Name"
FROM employees AS e
WHERE first_name = 'Hercules' AND last_name like 'B%';

--*************************************************************************************
-- 6. List all employees in the Sales department, including their employee number, last name,
-- first name, and department name.
-- *************************************************************************************
-- Create the query ---
```

```
SELECT e.emp_no AS "Employee Number",
            e.last_name AS "Last Name",
            e.first_name AS "First Name",
            d.dept_name AS "Department Name"
FROM employees AS e
          INNER JOIN dept_emp AS de
          ON e.emp_no = de.emp_no
                    INNER JOIN departments AS d
                    ON de.dept_no = d.dept_no
WHERE dept_name = 'Sales';

--*************************************************************************
-- 7. List all employees in the Sales and Development departments, including their employee number,
--   last name, first name, and department name.
-- *************************************************************************
-- Create the query ---
SELECT e.emp_no AS "Employee Number",
            e.last_name AS "Last Name",
            e.first_name AS "First Name",
            d.dept_name AS "Department Name"
FROM employees AS e
          INNER JOIN dept_emp AS de
          ON e.emp_no = de.emp_no
                    INNER JOIN departments AS d
                    ON de.dept_no = d.dept_no
WHERE dept_name = 'Sales' OR dept_name = 'Development';

-- *************************************************************************
-- 8. In descending order, list the frequency count of employee last names, i.e., how many employees
--   share each last name.
-- *************************************************************************
-- Create the query ---
SELECT e.last_name AS "Last Name",
            count(*) AS "Frequency Count"
FROM employees AS e
GROUP BY "Last Name"
ORDER BY "Frequency Count" DESC;

-- Observation:  Only one employee with last name: Foolsday
```

APPENDIX B

```
--  ************************************************************
```

-- 9. How long have the managers worked in their current role?

```
-- ************************************************************
-- Create the query ---
WITH emp_age_subq AS
                        (SELECT emp_no AS "Employee Number",
                                        last_name AS "Last Name",
                                        first_name AS "First Name",
                                        gender AS "Gender",
                                        EXTRACT(YEAR FROM age(hire_date)) AS "Employment
Years"
                        FROM employees),
        manager_age_subq AS
                                (SELECT       emp_no AS "Manager Number",
                                least(DATE_PART('year', to_date::date),DATE_PART('year', '2020-01-
01'::date)) -
                                DATE_PART('year', from_date::date) AS "Managerial Years"
                                FROM dept_manager)
SELECT "Last Name", "First Name","Employment Years","Managerial Years"
FROM emp_age_subq
INNER JOIN manager_age_subq ON "Employee Number" = "Manager Number";

-- ************************************************************************
```

-- 10. How many employees per title and on average how long do they work per title?

```
-- ************************************************************************
SELECT        title AS "Department Title",
              count(*) AS "Number of employees",
              ROUND(AVG((least(DATE_PART('year', to_date::date),DATE_PART('year', '2020-01-
01'::date)) -
              DATE_PART('year', from_date::date)))) AS "Avg. employment years"
FROM titles
GROUP BY "Department Title"
ORDER BY "Avg. employment years" DESC;

-- ************************************************************************
```

-- 11. What is the average salary per title?

```
-- ************************************************************************
```

```
SELECT        t.title AS "Department Title",
            count(t.emp_no) AS "No. of employees",
                  ROUND(AVG(s.salary),2) AS "Average Salary"
FROM titles AS t
  INNER JOIN salaries AS s
                  ON t.emp_no = s.emp_no
GROUP BY "Department Title"
ORDER BY "Average Salary" DESC;

-- ***************************************************************
-- 12. Do male managers get more salary on average than female managers?
-- ***************************************************************
-- Create the query ---
WITH salary_age_subq AS
                        (SELECT emp_no AS "Employee Salary ID",
                                    salary AS "Salary"
                         FROM salaries),
        manager_age_subq AS
                                (SELECT       emp_no AS "Manager ID",
                                              least(DATE_PART('year',
to_date::date),DATE_PART('year', '2020-01-01'::date)) -
                                              DATE_PART('year', from_date::date) AS "Managerial
Years"
                                FROM dept_manager),
        emp_gender_subq AS
                                (SELECT emp_no AS "Employee ID",
                                              gender AS "Gender"
                                 FROM employees)
SELECT        "Gender",
                  ROUND(AVG("Salary"),2) AS "Average Salary",
                  ROUND(AVG("Managerial Years")) AS "Avg. Managerial Years"
FROM emp_gender_subq
        INNER JOIN manager_age_subq ON "Employee ID" = "Manager ID"
                  INNER JOIN salary_age_subq ON "Employee ID" = "Employee Salary ID"
GROUP BY emp_gender_subq."Gender";

-- ***************************************************************
-- 13. Do male employees get more salary on average than their female counterparts?
-- ***************************************************************
--**** Creating a view table of salary for employees *****
```

```
-- Dropping the view table if exists ---
DROP VIEW IF EXISTS salary_emp;
-- Create the query ---
CREATE VIEW salary_emp AS
        WITH salary_subq AS
                        (SELECT emp_no AS "Employee Salary ID",
                                        salary AS "Salary"
                         FROM salaries),
                emp_subq AS
                        (SELECT emp_no AS "Employee ID",
                                        last_name AS "Last Name",
                                        first_name AS "First Name",
                                        gender AS "Gender",
                                        EXTRACT(YEAR FROM age(hire_date)) AS "Employment
Years"
                        FROM employees)
        SELECT "Employee ID","Last Name", "First Name","Gender","Employment Years","Salary"
        FROM emp_subq
        INNER JOIN salary_subq ON "Employee ID" = "Employee Salary ID";
-- Total number of employees grouped by gender --
SELECT "Gender",
                count(*) AS "No of Employees"
FROM salary_emp
GROUP BY "Gender";
-- Maximum salary of employees grouped by gender ---
SELECT "Gender",
     MAX("Salary") AS "Maximum Salary"
FROM salary_emp
GROUP BY "Gender";
-- Minimum salary of employees grouped by gender ---
SELECT "Gender",
     MIN("Salary") AS "Minimum Salary"
FROM salary_emp
GROUP BY "Gender";
-- Average salary of employees grouped by gender ---
SELECT "Gender",
      ROUND(AVG("Salary"),2) AS "Average Salary"
FROM salary_emp
GROUP BY "Gender";
```

```
-- **********************************************************************
-- 14. List of oldest hire among employees
-- **********************************************************************
--**** Creating a view table with age of all employees *****
-- Dropping the view table if exists ---
DROP VIEW IF EXISTS age_emp;
-- Create the query ---
CREATE VIEW age_emp AS
        WITH emp_subq AS
                        (SELECT emp_no AS "Employee ID",
                                        gender AS "Gender",
                                        last_name AS "Last Name",
                                        first_name AS "First Name",
                                        EXTRACT(YEAR FROM age(birth_date)) AS "Age",
                                        EXTRACT(YEAR FROM age(hire_date)) AS "Employment
Years"
                        FROM employees),
         dept_subq AS
                        (SELECT de.emp_no AS "Employee Dept ID",
                                        de.dept_no AS "Dept ID",
                                        d.dept_name AS "Department Name",
                                        d.dept_no AS "Department ID"
                         FROM dept_emp AS de
                                INNER JOIN departments AS d
                                        ON de.dept_no = d.dept_no),
         salary_subq AS
                        (SELECT emp_no AS "Employee Salary ID",
                                        salary AS "Salary"
                         FROM salaries)
        SELECT "Last Name","First Name","Gender","Age","Employment Years","Department
Name","Salary"
        FROM emp_subq
                INNER JOIN dept_subq ON "Employee ID" = "Employee Dept ID"
                        INNER JOIN salary_subq ON "Employee ID" = "Employee Salary ID";
-- Create a query from age_emp table to find the oldest employee --
SELECT * FROM age_emp ORDER BY "Age" DESC,"Last Name" ASC, "First Name" ASC LIMIT
10;
-- Create a query from age_emp table to find the oldest hire --
SELECT * FROM age_emp ORDER BY "Employment Years" DESC,"Last Name" ASC, "First Name"
ASC LIMIT 10;
```