



Introduction

In this project I will be using an employee database from MySQL to answer several questions for a company. The database is based on a fictional company. I will be using SQL to answer the questions and different techniques such as joins, common table expressions, nested queries, window, and aggregate functions. Now let's get to the list of questions the company wants to answer.

I will break up the questions into 2 groups:

Averages:

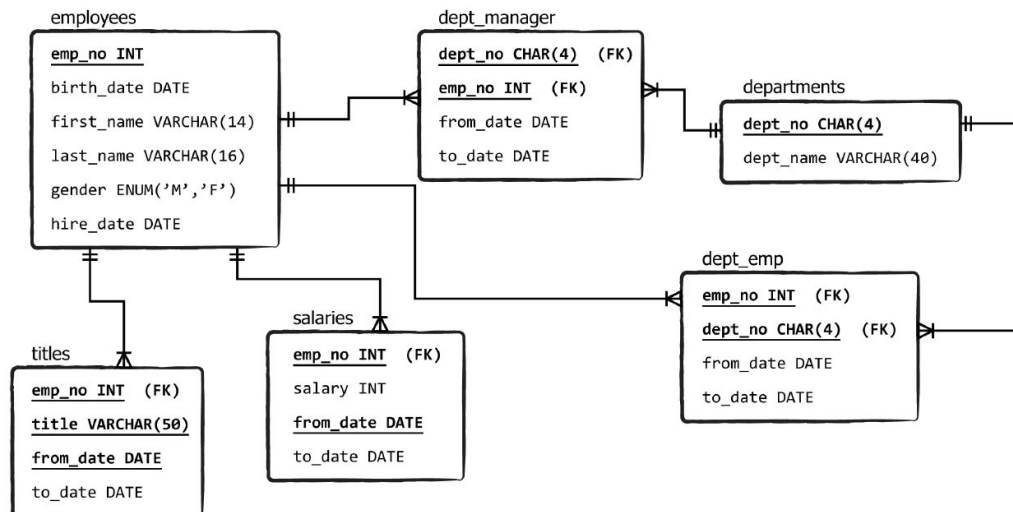
1) What is the average salary per department? 2) What is the average salary per job title? 3) What is the average salary per gender per department? 4) What is the number of male employees whose highest salaries have been below the all-time average. 5) What is the number of female employees whose highest salaries have been below the all-time average.

Salaries:

6) What is the minimum and maximum salaries in each department? Part B what is the difference between the min max? 7) How many employees make over 100,000 a year? Part B which department has the most salaries over 100,000? 8) What is the historical growth of salaries each year in the company? 9) Rank the manager salaries from highest to lowest. How many department managers make over 100,000?

Below is the schema of the database.

Database: employees



Starting with the average questions:

In [6]: `%%sql`
`#What is the average salary per department?`

```
SELECT

d.dept_name AS department,
ROUND(AVG(s.salary),2) AS avg_salary

FROM

salaries AS s

JOIN

dept_emp AS de ON de.emp_no = s.emp_no

JOIN

departments AS d ON de.dept_no = d.dept_no

GROUP BY department

ORDER BY department;
```

```
* mysql+mysqldb://root:***@127.0.0.1/employees
9 rows affected.
```

Out[6]:

department	avg_salary
Customer Service	58755.44
Development	59503.57
Finance	70159.47
Human Resources	55353.52
Marketing	71901.72
Production	59539.79
Quality Management	57294.66

Research	59866.24
Sales	80776.62

Viewing the table above we can see the average salary range is between 57,000 and 81,000. With Sales department being the highest average salary.

```
In [7]: %%sql
# What is the average salary per job title?

SELECT

t.title AS job_title,
ROUND(AVG(s.salary),2) AS avg_salary

FROM employees AS e

JOIN
salaries AS s ON e.emp_no = s.emp_no

JOIN

titles AS t ON s.emp_no = t.emp_no

GROUP BY

job_title

ORDER BY

job_title;

* mysql+mysqldb://root:***@127.0.0.1/employees
7 rows affected.
```

```
Out[7]:
```

job_title	avg_salary
Assistant Engineer	59377.45
Engineer	59486.10
Manager	66924.27
Senior Engineer	60552.28
Senior Staff	70398.25
Staff	69241.76
Technique Leader	59138.20

The average salary range per position is between 59,138 with Technique Leader being the lowest and Senior staff has the highest average salary of 70,398. The range falls between 59,138-70,398.

```
In [8]: %%sql
# What is the average salary per gender per department?

SELECT

e.gender AS gender,
d.dept_name AS department_name,
ROUND(AVG(s.salary),2) AS salary
FROM

salaries AS s
```

JOIN

```
employees AS e ON s.emp_no = e.emp_no
```

JOIN

```
dept_emp AS de ON de.emp_no = e.emp_no
```

JOIN

```
departments AS d ON d.dept_no = de.dept_no
```

```
GROUP BY d.dept_no , e.gender
```

```
ORDER BY d.dept_no, salary;
```

```
* mysql+mysqldb://root:***@127.0.0.1/employees
18 rows affected.
```

Out[8]:

gender	department_name	salary
--------	-----------------	--------

F	Marketing	71464.48
M	Marketing	72198.19
F	Finance	69914.92
M	Finance	70327.03
M	Human Resources	55196.55
F	Human Resources	55596.37
F	Production	59456.00
M	Production	59596.36
F	Development	59391.95
M	Development	59576.33
M	Quality Management	57206.90
F	Quality Management	57423.31
F	Sales	80626.56
M	Sales	80879.76
F	Research	59712.78
M	Research	59965.77
M	Customer Service	58590.99
F	Customer Service	58998.73

Here we can see that the Male gender salaries are higher in 8 out of 9 departments. The only department where females make more is in the human resources department. Even though male employees average higher than females the gap between average salaries are small.

In [31]:

```
%%sql
#What is the number of male employees whose highest salaries have been below the all-time

WITH c_avg_salary AS

(

SELECT AVG(salary) AS avg_salary FROM salaries
```

```

),

c_m_highest_salary AS (

SELECT s.emp_no, MAX(s.salary) AS max_salary

FROM salaries s JOIN employees e ON e.emp_no = s.emp_no AND e.gender = 'M'

GROUP BY s.emp_no

)

SELECT

COUNT(CASE WHEN c2.max_salary < c1.avg_salary THEN c2.max_salary ELSE NULL END) AS male_

FROM c_m_highest_salary c2

JOIN c_avg_salary c1;

* mysql+mysqldb://root:***@127.0.0.1/employees
1 rows affected.

```

Out[31]: male_salary_below_average

24753

In [32]: `%%sql`
#What is the number of female employees whose highest salaries have been below the all-t

```

WITH c_avg_salary AS

(

SELECT AVG(salary) AS avg_salary FROM salaries

),

c_m_highest_salary AS (

SELECT s.emp_no, MAX(s.salary) AS max_salary

FROM salaries s JOIN employees e ON e.emp_no = s.emp_no AND e.gender = 'F'

GROUP BY s.emp_no

)

SELECT

COUNT(CASE WHEN c2.max_salary < c1.avg_salary THEN c2.max_salary ELSE NULL END) AS femal

FROM c_m_highest_salary c2

JOIN c_avg_salary c1;

* mysql+mysqldb://root:***@127.0.0.1/employees
1 rows affected.

```

Out[32]: female_salaries_below_average

16590

Salary questions:

```

In [9]: %%sql
# What is the minimum and maxium salaries in each department? Part B what is the difference

SELECT

d.dept_name AS department,
MIN(s.salary) AS min_salary,
MAX(s.salary) AS max_salary,
MAX(s.salary) - MIN(s.salary) AS difference

FROM

employees AS e

JOIN
salaries AS s ON e.emp_no = s.emp_no

JOIN
dept_emp AS de ON e.emp_no = de.emp_no

JOIN
departments AS d ON de.dept_no = d.dept_no

GROUP BY d.dept_name
ORDER BY d.dept_name;

* mysql+mysqldb://root:***@127.0.0.1/employees
9 rows affected.

```

```

Out[9]:

```

department	min_salary	max_salary	difference
Customer Service	38836	144866	106030
Development	38850	144434	105584
Finance	38812	134662	95850
Human Resources	38735	123674	84939
Marketing	39127	143644	104517
Production	38836	132552	93716
Quality Management	38928	122376	83448
Research	38851	124181	85330
Sales	39427	158220	118793

Here we can see the largest differences in salary is in the sales department. With the smallest difference being in quality management. The sales department has the highest salary in the company with a salary of 158,220.

How many current employees make over 100,000 a year? Part B which department has the most salaries over 100,000?

The database has employees setup on a contract bases so there is a from and to date of employment. Some employees have indefinite contracts. To make some sense of this we will assume the year is 2003 and we are looking at the current employees on indefinite contracts.

```

In [10]: %%sql
#How many current employees make over $100,000 a year?
SELECT

COUNT(e.emp_no) AS total_employees_equal_or_above_100K

```

```

FROM
employees AS e

JOIN
titles AS t ON t.emp_no = e.emp_no

JOIN
dept_emp AS de ON e.emp_no = de.emp_no

JOIN
departments AS d ON de.dept_no = d.dept_no

JOIN
salaries AS s ON e.emp_no = s.emp_no

WHERE s.salary >= 100000 AND YEAR(s.to_date) > 2002;

```

```

* mysql+mysqldb://root:***@127.0.0.1/employees
1 rows affected.

```

Out[10]: **total_employees_equal_or_above_100K**

11458

There is 11,458 employees currently making over 100,000 a year.

In [11]: **%%sql**

```

#Part B which department has the most salaries over 100,000?

```

```

SELECT

d.dept_name AS department,
COUNT(s.salary) AS count_salary_100K_and_over

FROM
employees AS e

JOIN
titles AS t ON t.emp_no = e.emp_no

JOIN
dept_emp AS de ON e.emp_no = de.emp_no

JOIN
departments AS d ON de.dept_no = d.dept_no

JOIN
salaries AS s ON e.emp_no = s.emp_no

WHERE s.salary >= 100000 AND YEAR(s.to_date) > 2002
GROUP BY d.dept_name;

```

```

* mysql+mysqldb://root:***@127.0.0.1/employees
9 rows affected.

```

Out[11]: **department count_salary_100K_and_over**

Development	968
Sales	6572
Finance	917
Production	769
Quality Management	113

Marketing	1293
Customer Service	452
Human Resources	85
Research	289

In [12]:

```
%%sql

#What is the historical growth of salaries each year in the company?

SELECT
year,
salary,
salary - LAG(salary) OVER w AS salary_growth

FROM
(SELECT

YEAR(from_date) as year,
SUM(salary) as salary

FROM
salaries

GROUP BY year
ORDER BY year) AS ss

WINDOW w AS (ORDER BY year)
ORDER BY year

* mysql+mysqldb://root:***@127.0.0.1/employees
18 rows affected.
```

Out[12]:

year	salary	salary_growth
1985	330187637	None
1986	693344812	363157175
1987	1064354371	371009559
1988	1455739634	391385263
1989	1847710988	391971354
1990	2249696863	401985875
1991	2649056889	399360026
1992	3074852712	425795823
1993	3483602574	408749862
1994	3889568129	405965555
1995	4297018475	407450346
1996	4719800054	422781579
1997	5122272444	402472390
1998	5515874590	393602146
1999	5893872958	377998368
2000	5956812513	62939555
2001	5948475900	-8336613

As we can see above the total salary growth in the company increased from 1986-2000. Then the salaries started to decrease starting in 2001.

```
In [24]: %%sql
#Rank the manager salaries from highest to lowest. How many department managers make ove

SELECT

q1.*,
RANK() OVER w AS salary_rank

FROM
(SELECT

dm.emp_no AS dept_manager_emp_no,
d.dept_name AS department,
MAX(s.salary) AS highest_salary

FROM
dept_manager as dm

JOIN
salaries AS s ON dm.emp_no = s.emp_no

JOIN
departments as d ON dm.dept_no = d.dept_no

GROUP BY dm.emp_no, department
ORDER BY dm.emp_no) as q1
WINDOW w AS (ORDER BY q1.highest_salary desc)

* mysql+mysqldb://root:***@127.0.0.1/employees
24 rows affected.
```

```
Out[24]:
```

dept_manager_emp_no	department	highest_salary	salary_rank
110022	Marketing	108407	1
110039	Marketing	106491	2
111400	Research	103244	3
111133	Sales	101987	4
111035	Sales	95873	5
110725	Quality Management	93193	6
110085	Finance	88443	7
110114	Finance	83457	8
111534	Research	79393	9
110183	Human Resources	79229	10
110800	Quality Management	77626	11
110344	Production	75121	12
110567	Development	74510	13
110511	Development	74056	14
111692	Customer Service	73953	15

110854	Quality Management	72876	16
110303	Production	72583	17
110765	Quality Management	72432	18
111877	Customer Service	71406	19
110386	Production	66995	20
110228	Human Resources	65400	21
111939	Customer Service	58968	22
111784	Customer Service	58189	23
110420	Production	56654	24

Above we can see there are 4 managers making above 100K a year. Both marketing managers, one research, and one sales manager.

Conclusion:

In this project I used my knowledge of the data manipulation language to answer the questions of a fictitious company.

- I uploaded a database in MySQL
- Connected jupyter notebooks to MySQL so I can present the input and outputs together.
- Thought of what questions I can create to ask myself that will showcase my knowledge using the data manipulation language.