

PHASE # 4 – COMPENSATION

- QUESTION # 1 – WHAT IS THE TOTAL SALARY FOR THE COMPANY?
- QUESTION # 2 – HOW MANY ACTIVE EMPLOYEES ARE THERE?
- QUESTION # 3 – WHAT IS THE AVERAGE TENURE OF AN EMPLOYEE?
- QUESTION # 4 – WHAT JOB CODES HAVE THE MOST EMPLOYEES WHOSE SALARY IS NOT ABOVE THE MEDIAN?
- QUESTION # 5 – WHAT JOB FAMILY HAS THE MOST EMPLOYEES?
- QUESTION # 6 – WHAT ARE THE DEMOGRAPHICS OF EMPLOYEES?

REQUIREMENTS:

All visuals must be able to filter on Level and Job Family
Level must exclude, “Executive”
Separate Hourly and Salary employees

SQL code – exporting employee data

```
USE portfolio;
SELECT e.employee_id, e.sex, e.race,
TIMESTAMPDIFF(YEAR, birthdate, CURDATE()) AS current_age, -- Add EEs current age
TIMESTAMPDIFF(YEAR, hire_date, CURDATE()) AS years_worked, -- Add years worked
e.position_number, p.position_name, jf.job_family_name, jc.job_code, jc.job_code_name,
jc.salary_min, jc.salary_25th_percentile, jc.salary_median_50th_percentile,
jc.salary_75th_percentile, jc.salary_max, jf.job_family_name, e.salary, wt.wagetype_name,
jc.weekly_hours

FROM employees e
INNER JOIN position p
    ON e.position_number = p.position_number
LEFT JOIN job_family jf
    ON p.job_family_code = jf.job_family_code
LEFT JOIN job_code jc
    ON p.job_code = jc.job_code
LEFT JOIN wagetype wt
    ON p.wagetype_code = wt.wagetype_code

WHERE e.position_number != '99999999' -- exclude position 99999999 employees
    AND wt.wagetype_name != 'Hourly'; -- exclude hourly employees*
```

*-changed to ‘Salary’ to exclude Salary employees

Power BI – Measures for Data Pane

DAX

Create Range Min Measure

```
Range Min = RELATED('PHASE 4 - Compensation SUM'[salary_min])
```

Create Range Median Measure

```
Range Median = RELATED('PHASE 4 - Compensation
SUM'[salary_median_50th_percentile])
```

Create Range Max Measure

```
Range Max = RELATED('PHASE 4 - Compensation SUM'[salary_max])
```

Create Average Salary Measure

```
Average Salary = AVERAGE('PHASE 4 - Compensation salaries'[salary])
```

Code by PBI Page:

SUMMARY

DAX

Chart: Market Ranges by Job Code

- Create the Avg Sal > Median visual calculation

```
Avg Sal > Median = [Average Salary] - [Market Median]
```

Chart: Job Codes Information

- Create % of Job Codes visual calculation

```
Percent of Job Codes = DIVIDE(SUM([Total Job Code]), CALCULATE(SUM([Total Job Code]), ALL([Job Family Name])))
```

- Create % of EEs visual calculation

```
Percent of EEs = DIVIDE(SUM([Total Employees]), CALCULATE(SUM([Total Employees]), ALL([Job Family Name])))
```

SALARY & HOURLY PAY RANGES

DAX – used in both pages

Chart: Job Code Information

- Create Salaries Above Range Measure

```
Salaries Above Range Count =  
COUNTROWS(FILTER('PHASE 4 - Compensation salaries',  
'PHASE 4 - Compensation salaries'[salary] > CALCULATE(MIN('PHASE 4 - Compensation salaries'[Range Max]))))
```

- Create Salaries Below Range Measure

```
Salaries Below Range Count =  
COUNTROWS(FILTER('PHASE 4 - Compensation salaries',  
'PHASE 4 - Compensation salaries'[salary] < CALCULATE(MIN('PHASE 4 - Compensation salaries'[Range Min]))))
```

Chart: Salary by Job Family

Python

```
# The following code to create a dataframe and remove duplicated rows is always executed and  
acts as a preamble for your script:
```

```
# dataset = pandas.DataFrame(job_family_name, salary_min, salary_25th_percentile,  
salary_median_50th_percentile, salary_75th_percentile, salary_max, salary, wagetype_name)  
# dataset = dataset.drop_duplicates()
```

Paste or type your script code here:

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import pandas as pd
```

```
# 'dataset' is the pandas DataFrame created automatically by Power BI
# Ensure you have a 'Category' field for the x-axis and a 'Value' field for the y-axis in the Values
section

#Adjust figure size
plt.figure(figsize=(18, 12))

#BOX AND WHISKER
#Filter box and whisker plot
filtered_data = dataset[dataset['wagetype_name'] != 'Hourly']
#Sort box and whisker by Job Family
sorted_categories = sorted(dataset['job_family_name'].unique())
# Create the box plot using seaborn
ax = sns.boxplot(x='job_family_name', y='salary', data=filtered_data, order=sorted_categories,
showfliers=False, color='#AAA662')
#Adjust label size
ax.tick_params(axis='x', labelsize=14)
ax.tick_params(axis='y', labelsize=16)

#SCATTER
#Filter data on scatter plot
filtered_data = dataset[dataset['wagetype_name'] != 'Hourly']
#Sort scatter plot by Job Family
sorted_categories = sorted(dataset['job_family_name'].unique())
#Create and overlay the scatter plot (stripplot)
#Use 'jitter=True' to spread points for better visibility and 'dodge=True' to ensure they align
correctly with the box plots
sns.stripplot(x='job_family_name', y='salary', data=filtered_data, order=sorted_categories,
color='3', jitter=True, size=6, ax=ax)

#Create title and labels
plt.title('Salary by Job Family', loc='center', fontsize=26)
plt.xlabel('')
plt.ylabel('Rate', fontsize=14)

#Rotate x-axis labels
plt.xticks(rotation=45)
plt.tight_layout(pad=2)

plt.show()
```

Values in **Bold** were changed for the chart on Salary Page