```
In [1]:   # Import dependencies
          from sqlalchemy import create_engine
          from db_pw import id, pw

          import pandas as pd
          import matplotlib.pyplot as plt
```

```
In [2]:   # Create an engine to read data from the database
          engine = create_engine(f"postgresql://{id}:{pw}@localhost/employee")
```

```
In [3]:   # Get salary data
          salary_df = pd.read_sql("salaries", engine)
          salary_df.head()
```

Out[3]:

	emp_no	salary
0	10001	60117
1	10002	65828
2	10003	40006
3	10004	40054
4	10005	78228

```
In [4]:   # Plot a histogram to see the most common salary range.
          bins = 10
          salary_df["salary"].hist(bins = bins, color = "skyblue", ec = "black")
          plt.xlabel("Salary ($)")
          plt.ylabel("Counts")
          plt.title("Histogram of Salary")
          plt.xticks(rotation = 30)
          plt.show()
```



```
In [5]:   # Calculate quartiles
          mean = salary_df["salary"].mean()
          std = salary_df["salary"].std(ddof = 0)
          lowerq = salary_df["salary"].quantile([0.25, 0.5, 0.75])[0.25]
          med = salary_df["salary"].quantile([0.25, 0.5, 0.75])[0.5]
          upperq = salary_df["salary"].quantile([0.25, 0.5, 0.75])[0.75]
          iqr = upperq - lowerq
          lower_bound = lowerq - (1.5 * iqr)
          upper_bound = upperq + (1.5 * iqr)
```

```
In [6]:  # Plot a boxplot
         plt.figure(figsize = (8,2))
         salary_df["salary"].plot.box(vert = False, sym = "+", widths=0.5)
         plt.xlabel("Salary($)")
         plt.show()
         print("Mean: ", round(mean, 2))
         print("Std. Deviation: ", round(std, 2))
         print("Median: ", med)
         print("Q1: ", lowerq)
         print("Q3: ", upperq)
         print("Lower Bound: ", lower_bound)
         print("Upper Bound: ", upper_bound)
```
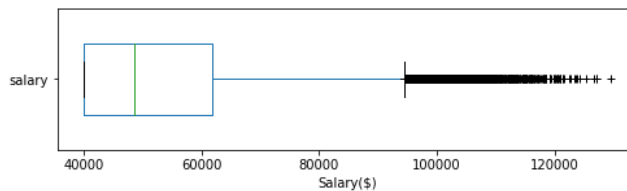


```
Mean:  52970.73
Std. Deviation:  14301.45
Median:  48681.0
Q1:  40000.0
Q3:  61758.0
Lower Bound:  7363.0
Upper Bound:  94395.0
```

```
In [7]:  # Read title and employee data
         title_df = pd.read_sql("titles", engine)
         employees_df = pd.read_sql("employees", engine)
```

```
In [8]:  # Merge all data
         merge_df = pd.merge(employees_df, title_df, left_on = "emp_title", right_on = "title_id", how = "inner")
         merge_df = pd.merge(merge_df, salary_df, on = "emp_no", how = "inner")
```

```
In [9]:  merge_df.head()
```

Out[9]:

	emp_no	emp_title	birth_date	first_name	last_name	sex	hire_date	title_id	title	salary
0	473302	s0001	1953-07-25	Hideyuki	Zallocco	M	1990-04-28	s0001	Staff	40000
1	421786	s0001	1957-09-28	Xiong	Verhoeff	M	1987-11-26	s0001	Staff	40000
2	273487	s0001	1957-04-14	Christoph	Parfitt	M	1991-06-28	s0001	Staff	56087
3	246449	s0001	1958-03-23	Subbu	Bultermann	F	1988-03-25	s0001	Staff	87084
4	48085	s0001	1964-01-19	Venkatesan	Gilg	M	1993-06-28	s0001	Staff	63016

```
In [10]:  # Calculate average salaries by title
          group_df = merge_df.loc[:, ["title", "salary"]].groupby("title").mean()
```

```python
# Plot a bar chart to reflect average salaries by title
group_df.plot.bar(color = "skyblue", label = "Avg. Salary ($)")
plt.title("Avg. Salary by title")
plt.xlabel("Title")
plt.ylabel("Avg. Salary ($)")
plt.xticks(rotation = 30)
plt.legend(loc = "best")
plt.show()
```
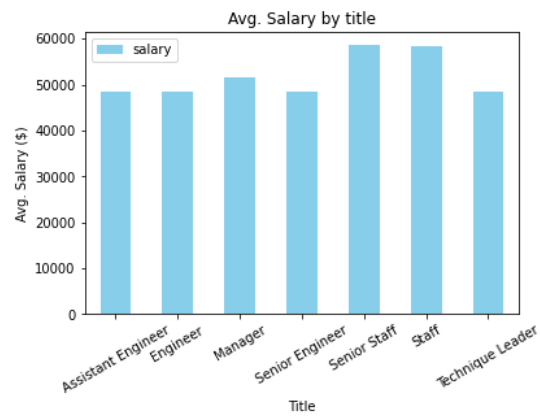
```python
# Search for my ID number, 499942. Oh, it's April foolsday!
merge_df.loc[merge_df.loc[:, "emp_no"] == 499942]
```

	emp_no	emp_title	birth_date	first_name	last_name	sex	hire_date	title_id	title	salary
287532	499942	e0004	1963-01-10	April	Foolsday	F	1997-02-10	e0004	Technique Leader	40000