IBM Data Analyst Capstone Project

Analysis on Emerging Technology Skills and Trends

Oscar Hidalgo Gutiérrez

02 June 2024

OUTLINE


Executive Summary


Introduction


Methodology


Results

Visualization – Charts
Dashboard


Discussion

Findings & Implications


Conclusion


Appendix

EXECUTIVE SUMMARY

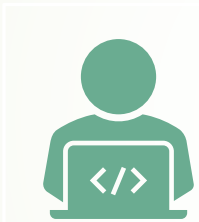
- Staying ahead in the global IT market requires adapting to rapidly evolving technologies. This report leverages data analytics to shed light on current and forecasted skill demands within the realm of programming languages, databases, and additional tech domains. It also examines the demographic profiles of tech industry professionals.
- Data was sourced from a Stack Overflow survey, the IBM website, and Github job listings. It underwent collection, cleansing, exploratory analysis, and was then displayed on dashboards.
- The analysis reveals that JavaScript remains the most favored programming language with expectations to maintain its popularity. Currently, MySQL sees the most usage, whereas PostgreSQL is expected to experience increased demand going forward.
- Moreover, the bulk of survey participants are male, based in the USA, and are typically 28 years old.

INTRODUCTION

- This report employs data analytics to elucidate both current and forthcoming trends concerning the demand for skills in programming languages, databases, platforms, and web frameworks.
- The research focused on the following questions:
    1. What are the most sought-after programming languages currently?
    2. Which database skills are in high demand?
    3. Which are the leading IDEs and web frameworks?
- The intended readership of this study includes IT professionals, HR managers, and anyone interested in the IT industry, aiming to understand the most vital IT skills today and their future relevance.

METHODOLOGY

Data was collected in a variety of formats, such as job availability for different technologies and locations, utilizing the Github jobs API in Python.



Programming language names and their associated annual salaries were extracted via web scraping from the IBM website. Additionally, a dataset from the 2019 Stack Overflow developer survey was downloaded and stored.

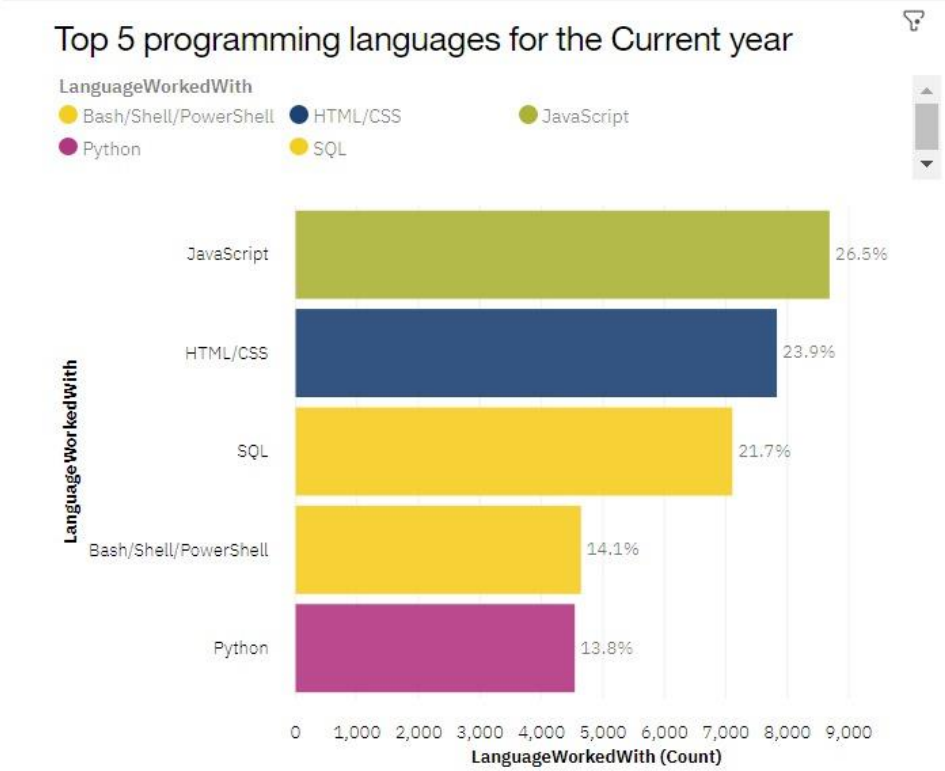
The data was cleaned and analyzed using Python. An exploratory data analysis (EDA) was conducted to evaluate the distribution, detect any outliers, and identify correlations between different columns within the dataset.

Visualization of the data was achieved through the creation of charts, graphs, and dashboards using Python and Cognos analytics tools. All Python analyses were performed in Jupyter Notebook within Visual Studio.

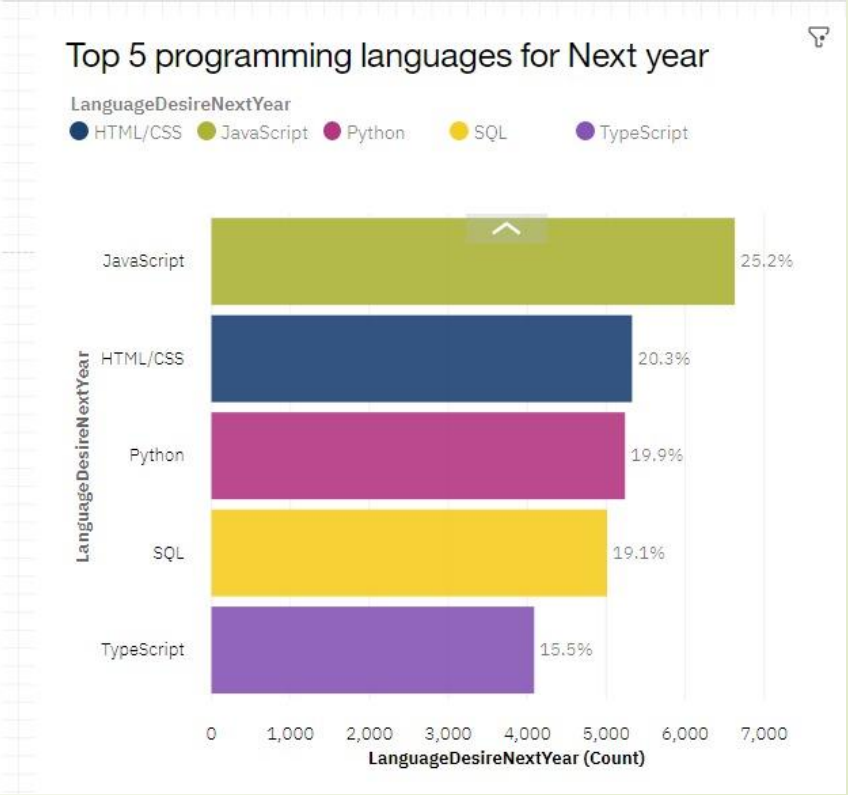
RESULTS

PROGRAMMING LANGUAGE TRENDS

- Current year



- Next year



PROGRAMMING LANGUAGE TRENDS FINDINGS & IMPLICATIONS

Findings

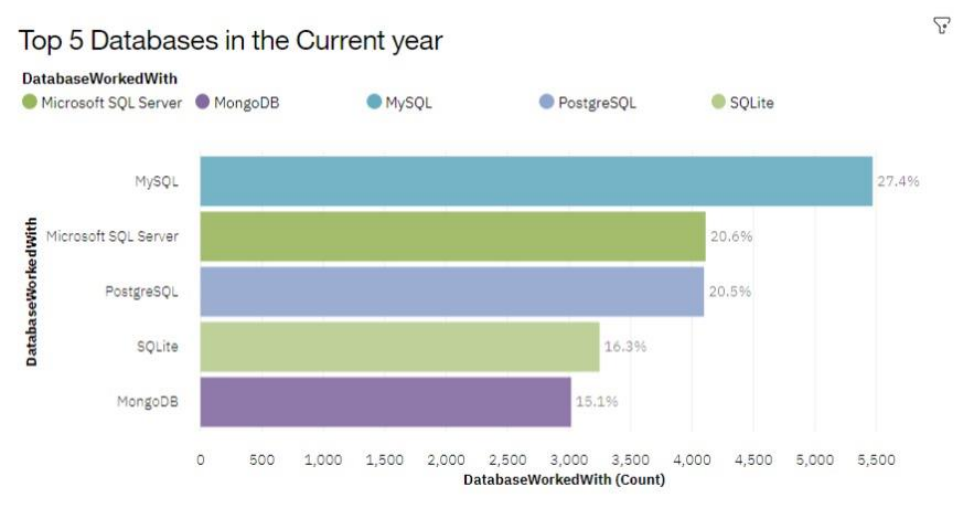
- Currently, the most widely used programming languages include JavaScript, HTML/CSS, SQL, Shell languages, and Python.
- For the coming years, JavaScript, HTML/CSS, Python, SQL, and TypeScript are expected to be the most prevalent languages.
- Python is projected to be in higher demand than SQL next year.

Implications

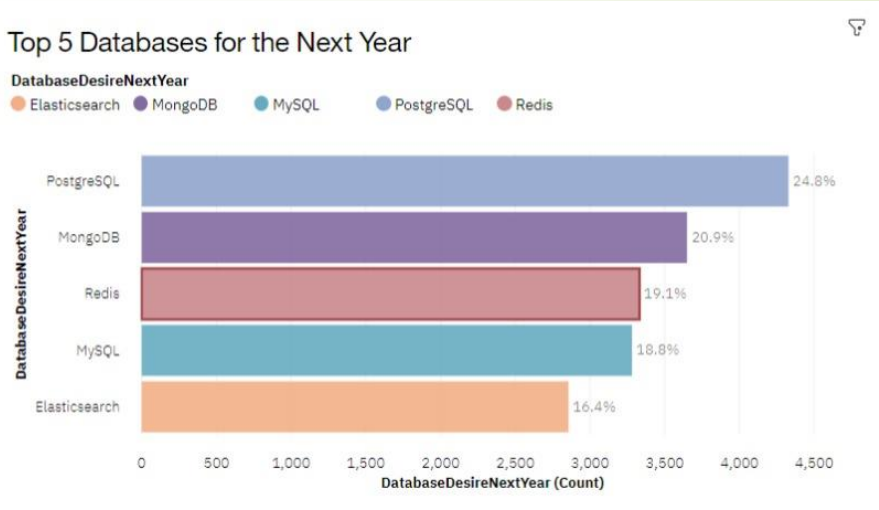
- The significant use of JavaScript and HTML for web development indicates a high demand for web development skills, particularly as TypeScript's popularity continues to grow.
- Python is increasingly popular due to rising demands in artificial intelligence (AI) and machine learning (ML) skills.
- SQL remains crucial for data professionals and is essential for those aiming to work as data analysts, scientists, or business analysts, highlighting the importance of SQL skills.

DATABASE TRENDS

- Current year



- Next year



DATABASE TRENDS FINDINGS & IMPLICATIONS

Findings

- Currently, MySQL, Microsoft SQL Server, PostgreSQL, SQLite, and MongoDB rank as the top five most utilized databases.
- However, PostgreSQL, MongoDB, Redis, MySQL, and Elasticsense are expected to rise in popularity in the coming years.
- Redis and Elasticsense, relatively new in the market, are poised for increased traction within the IT industry.

Implications

- SQL remains a critical tool for data specialists to monitor.
- The preference for open-source databases continues among companies.
- Oracle SQL, however, does not appear in the top five and is gradually losing its relevance.

DASHBOARD

DASHBOARD TAB 1

All tabs

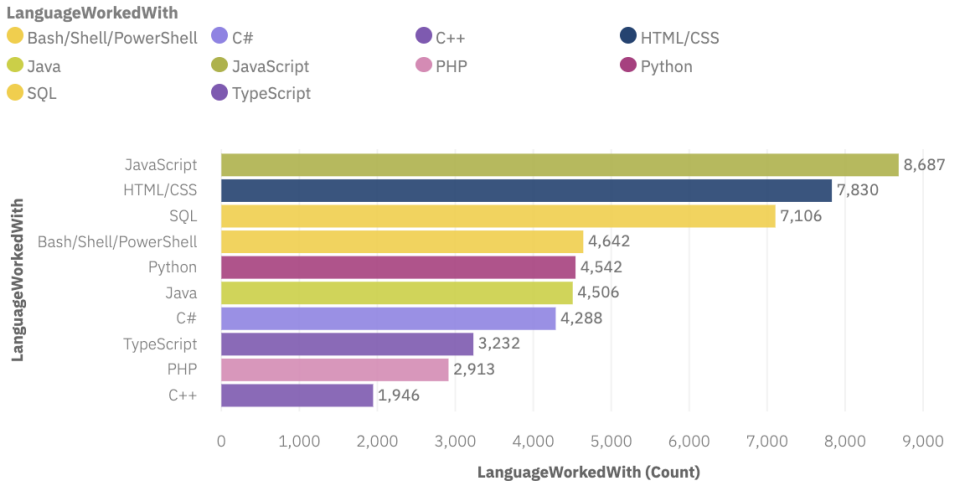
Drag and drop data here to filter all tabs.

This tab

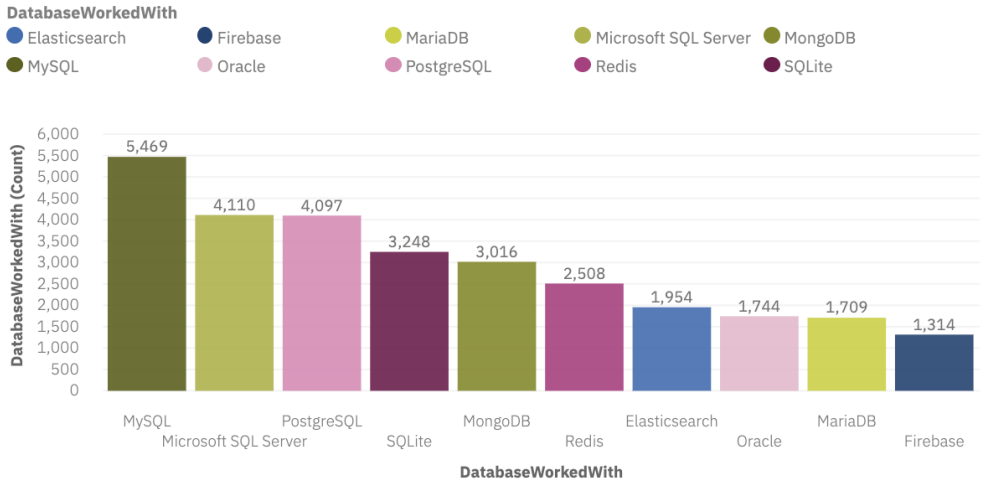
Drag and drop data here to filter this tab.

Current Technology Usage | Future Technology Trend | Demographics

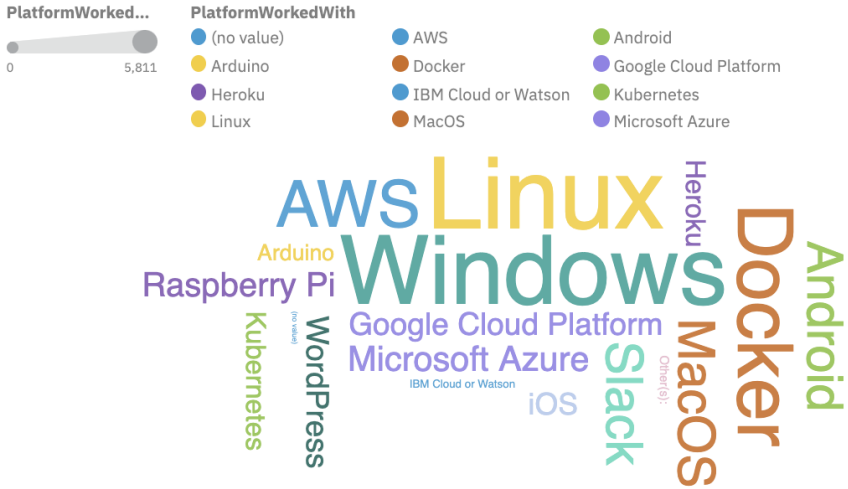
Top 10 LanguageWorkedWith



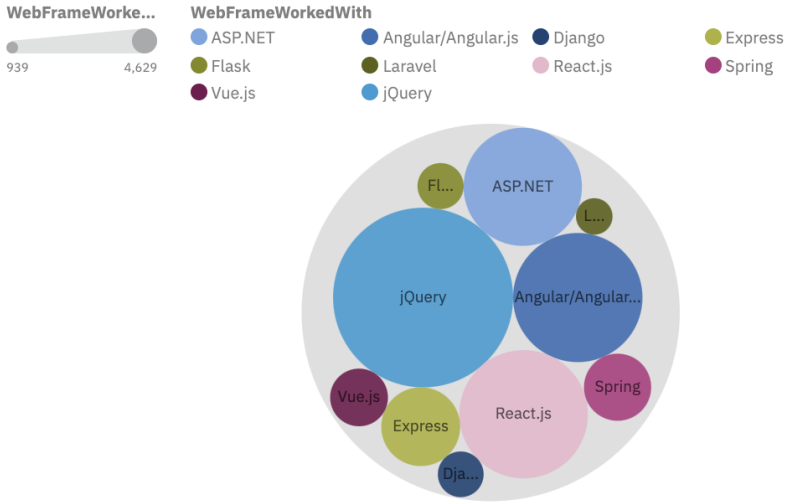
Top 10 DatabaseWorkedWith



PlatformWorkedWith



Top 10 WebFrameWorkedWith



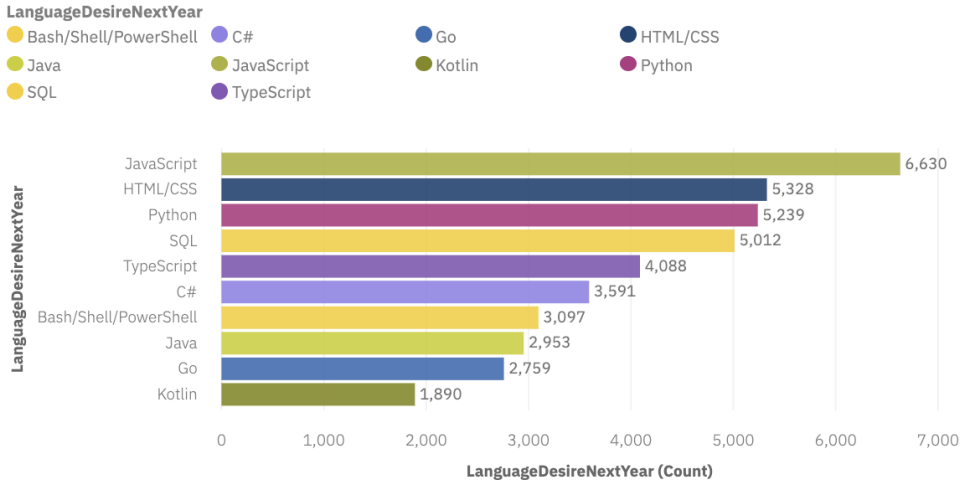
DASHBOARD TAB 2

All tabs — Drag and drop data here to filter all tabs.

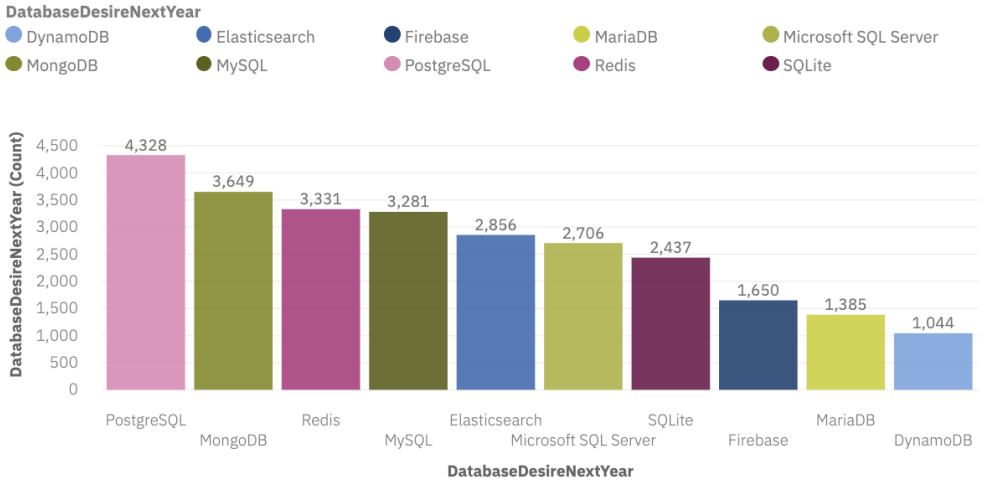
This tab — Drag and drop data here to filter this tab.

Current Technology Usage | Future Technology Trend | Demographics

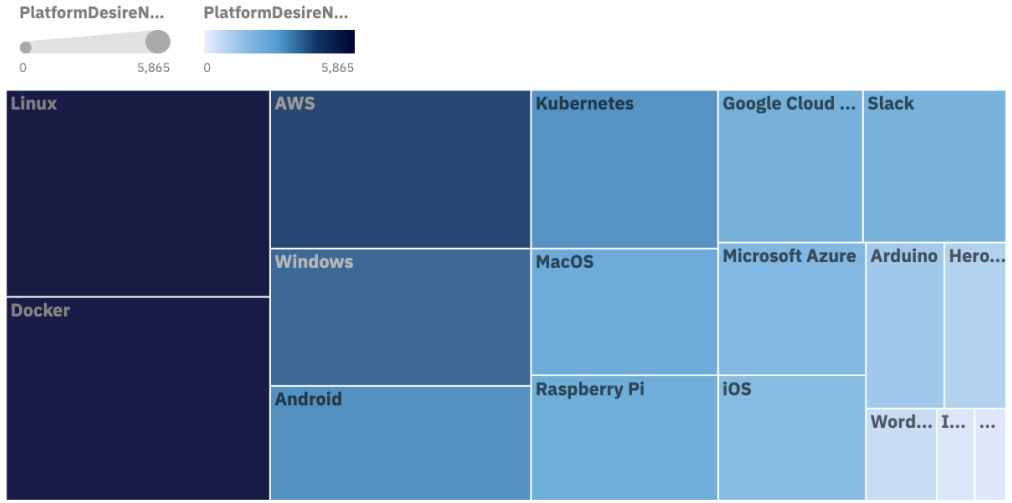
Top 10 LanguageDesireNextYear



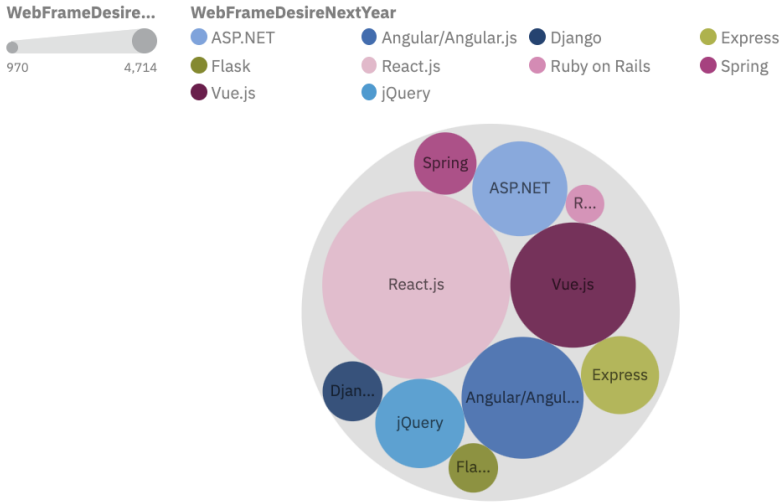
Top 10 DatabaseDesireNextYear



PlatformDesireNextYear



Top 10 WebFrameDesireNextYear



DASHBOARD TAB 2

All tabs

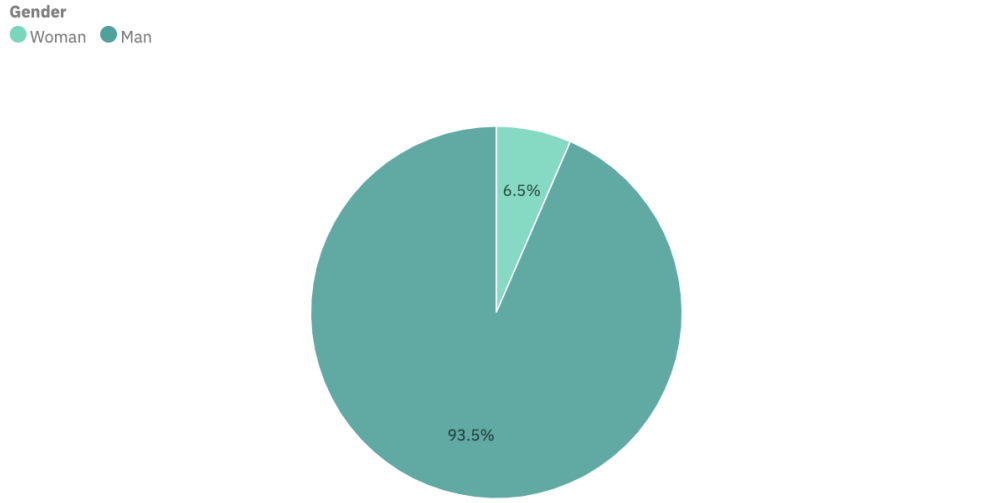
Drag and drop data here to filter all tabs.

This tab

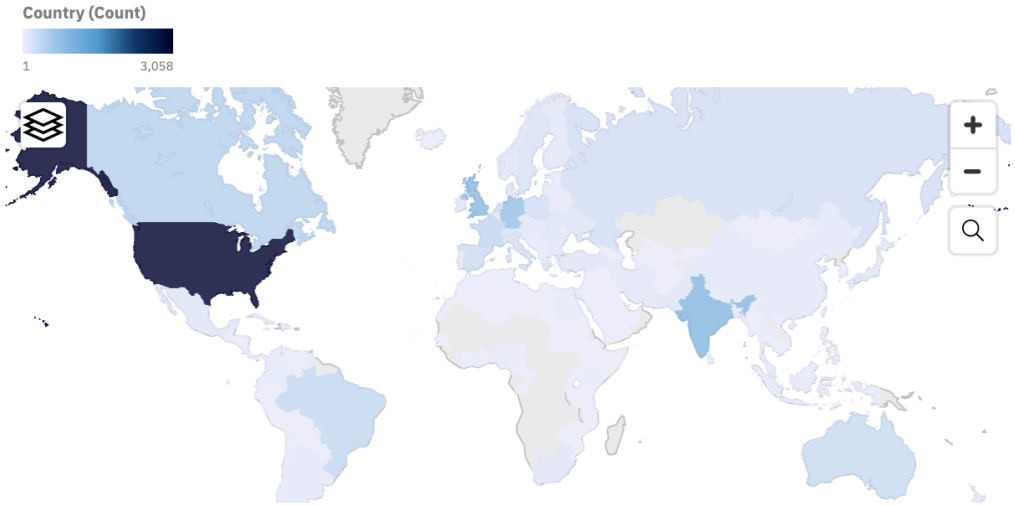
Gender
Man, Woman 2

Current Technology Usage | Future Technology Trend | Demographics

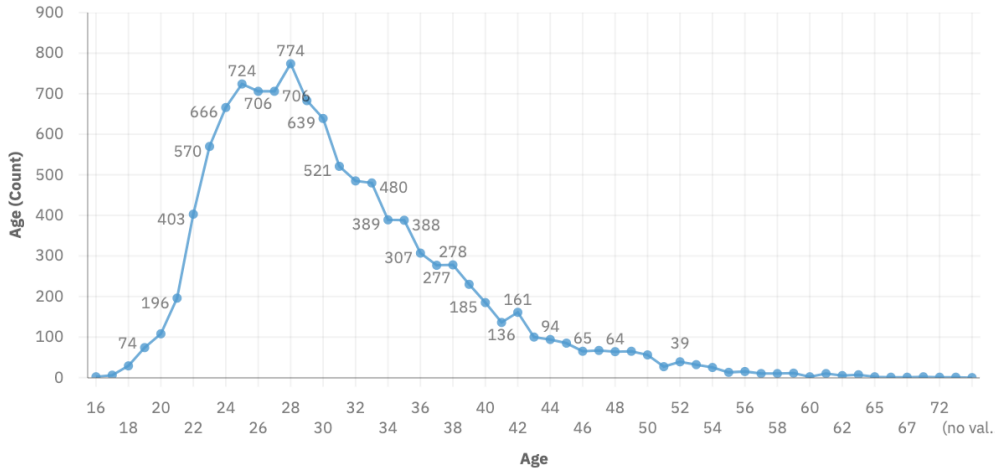
Respondent classified by Gender



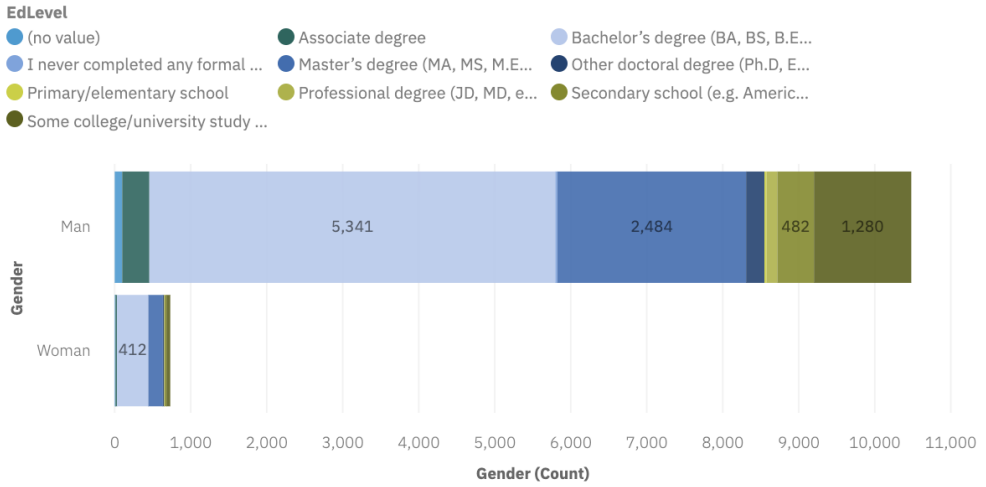
Respondent Count for Countries



Respondent Count by Age



Respondent Count by Gender, classified by Formal Education Level



DISCUSSION

- Enhancing skills within the technology sector.
- Strategies to narrow the significant gender disparity in technology.
- Necessity of graduate or postgraduate degrees in tech careers.
- Rising popularity and demand for mobile development, especially with Kotlin's growth.
- Expansion of technology education and resources in less developed areas of Southeast Asia, South America, Africa, and some European regions.
- Future relevance of Oracle SQL in the tech industry.

OVERALL FINDINGS & IMPLICATIONS

Findings

- Most people in the IT field have a Bachelors' degree.
- Web development languages are the most popular and on-demand tools in the IT field currently.
- The Tech sector is filled with majorly young people under 40 years of age.
- Most respondents want to learn Postgre SQL and React JS next year.

Implications

- It is important for data professionals to develop proficiencies in NoSQL in addition to SQL databases.
- Web development is still a very lucrative skill.
- Less developed countries need more access to tech trainings and education.

CONCLUSION

IT IS ESSENTIAL TO STAY UPDATED IN THE TECH SECTOR AS TRENDS CONTINUOUSLY EVOLVE. ADDITIONALLY, IT'S CRUCIAL TO CLOSE THE GENDER GAP AND EXPAND EDUCATIONAL OPPORTUNITIES IN LESS DEVELOPED REGIONS.

ASSESSING THE NECESSITY OF ADVANCED DEGREES TO MEET JOB MARKET DEMANDS IS ALSO VITAL. AS THE DEMAND FOR MOBILE DEVELOPMENT AND SKILLS IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING INCREASES, PROFESSIONALS NEED TO ADAPT TO NEW TOOLS WHILE EVALUATING THE ONGOING RELEVANCE OF ESTABLISHED LANGUAGES LIKE SQL.

APPENDIX

In [19]:
```
norm_annual_comp = []

for i in range(len(df)):

    if df['CompFreq'].iloc[i] == 'Weekly':
        norm_annual_comp.append(df['CompTotal'].iloc[i]*52)
    elif df['CompFreq'].iloc[i] == 'Monthly':
        norm_annual_comp.append(df['CompTotal'].iloc[i]*12)
    else:
        norm_annual_comp.append(df['CompTotal'].iloc[i]*1)

df['NormalizedAnnualCompensation']= norm_annual_comp
df.head()
```

Out[19]:

	Respondent	MainBranch	Hobbyist	OpenSourcer	OpenSource	Employment	Country	Student	EdLevel	UndergradMajor
0	4	I am a developer by profession	No	Never	The quality of OSS and closed source software ...	Employed full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...
1	9	I am a developer by profession	Yes	Once a month or more often	The quality of OSS and closed source software ...	Employed full-time	New Zealand	No	Some college/university study without earning ...	Computer science, computer engineering, or sof...
2	13	I am a developer by profession	Yes	Less than once a month but more than once per ...	OSS is, on average, of HIGHER quality than pro...	Employed full-time	United States	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...
3	16	I am a developer by profession	Yes	Never	The quality of OSS and closed source software ...	Employed full-time	United Kingdom	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	NaN
4	17	I am a developer by profession	Yes	Less than once a month but more than once per ...	The quality of OSS and closed source software ...	Employed full-time	Australia	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...

5 rows × 86 columns

What is the median **NormalizedAnnualCompensation**?

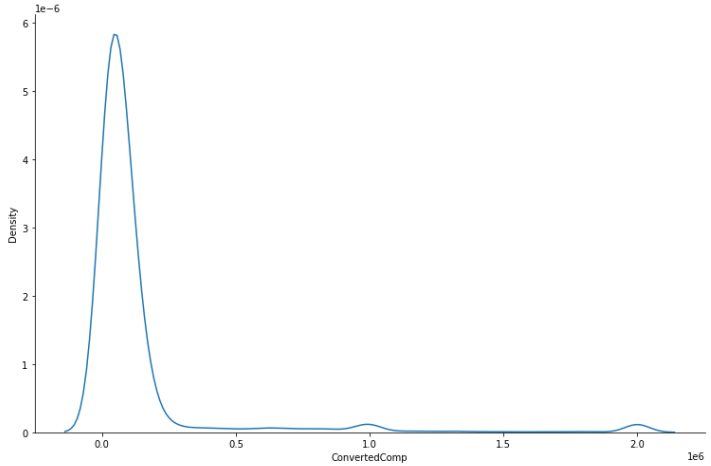
In [20]:
```
df['NormalizedAnnualCompensation'].describe()
```

Out[20]:
```
count    1.058900e+04
mean     6.133295e+06
std      9.838157e+07
min      0.000000e+00
25%      5.200000e+04
50%      1.000000e+05
75%      3.600000e+05
max      8.400000e+09
Name: NormalizedAnnualCompensation, dtype: float64
```

APPENDIX

In [8]:
```
sns.displot(df['ConvertedComp'], kind="kde", height=7, aspect = 1.5)
```

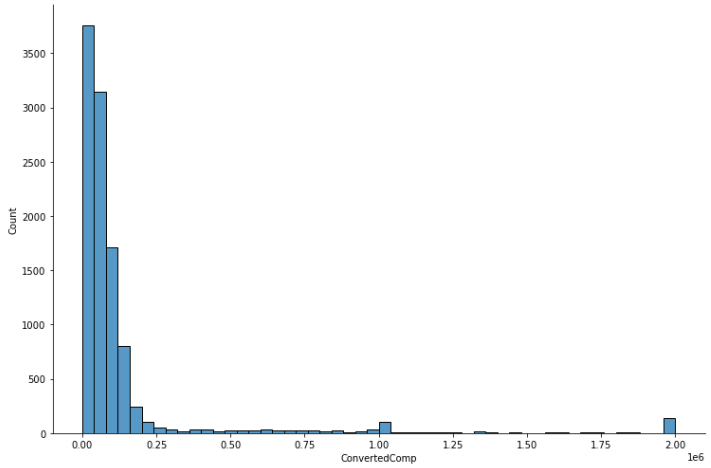
Out[8]:
```
<seaborn.axisgrid.FacetGrid at 0x1dc6da1e580>
```



We plot the histogram for the column `ConvertedComp`.

In [9]:
```
sns.displot(df['ConvertedComp'],bins=50, height=7, aspect = 1.5)
```

Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x1dc71f40310>
```



What is the median of the column `ConvertedComp`?

In [10]:
```
df['ConvertedComp'].describe()
```

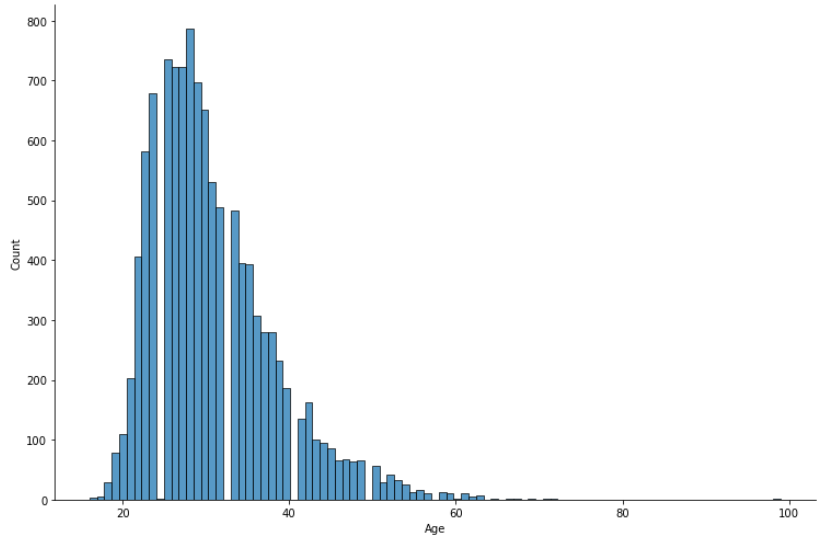
Out[10]:
```
count    1.058200e+04
mean     1.315967e+05
std      2.947865e+05
min      0.000000e+00
25%      2.686800e+04
50%      5.774500e+04
75%      1.000000e+05
max      2.000000e+06
Name: ConvertedComp, dtype: float64
```

We can see the median is 57745 as it is equal to the 50th percentile

APPENDIX

In [21]:
```
sns.displot(df['Age'], height=7, aspect = 1.5)
```

Out[21]:
```
<seaborn.axisgrid.FacetGrid at 0x1dc7315b550>
```

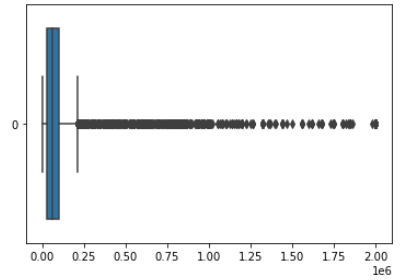


Findin Outliers

We find out if outliers exist in the column `ConvertedComp` using a box plot

In [22]:
```
sns.boxplot(data=df['ConvertedComp'], orient = 'h')
```

Out[22]:
```
<AxesSubplot:>
```



We can find out the Inter Quartile Range for the column `ConvertedComp`.

In [23]:
```
q1 = df['ConvertedComp'].quantile(0.25)
q3 = df['ConvertedComp'].quantile(0.75)

print('Q1', q1)
print('Q3', q3)

IQR = q3 - q1
print('IQR', IQR)
```

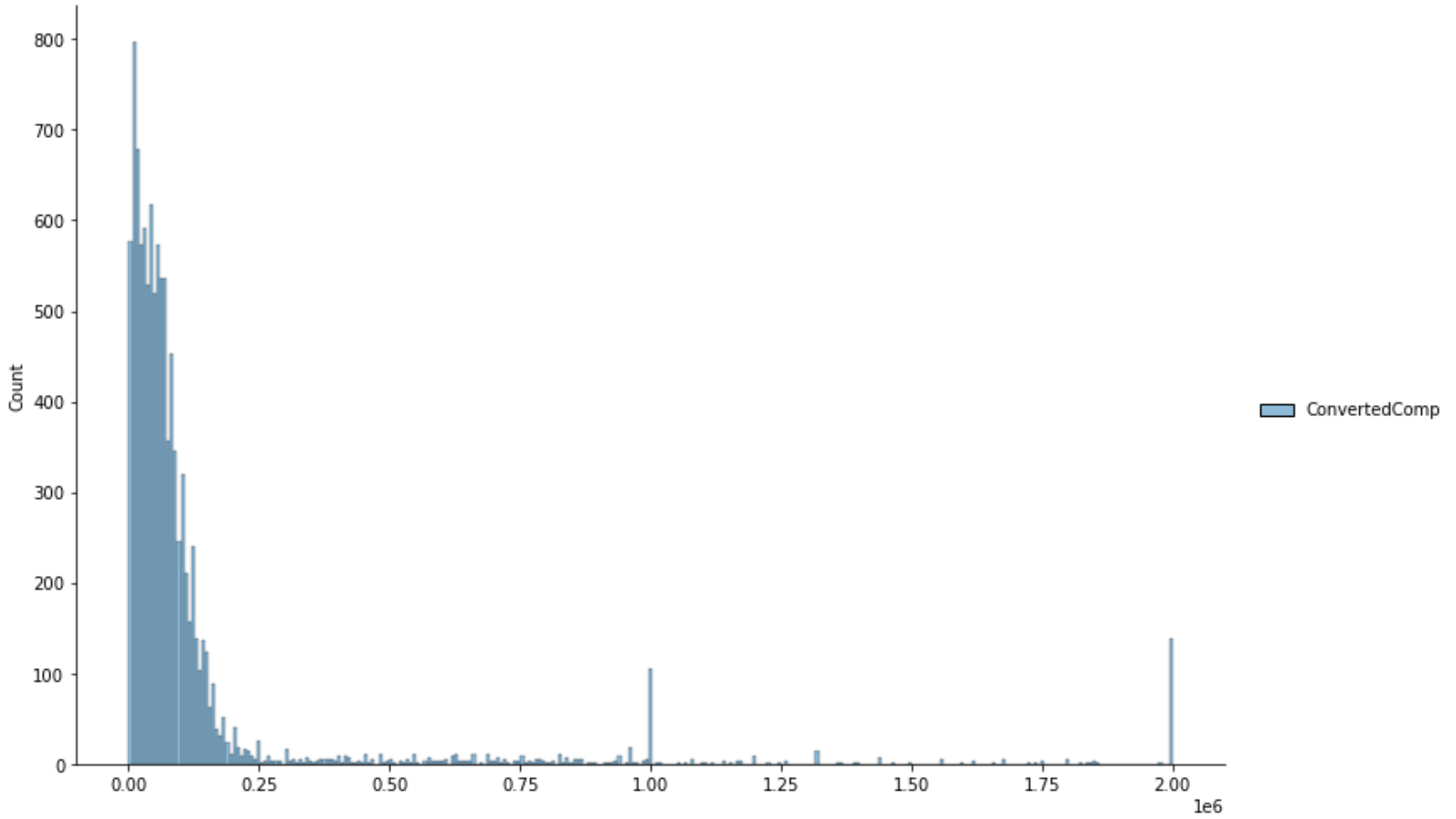
```
Q1 26868.0
Q3 100000.0
IQR 73132.0
```

APPENDIX

In [9]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [10]:
```
QUERY = """
SELECT ConvertedComp
FROM master

"""
df = pd.read_sql_query(QUERY,conn)

sns.displot(df, height = 7, aspect = 1.5)
```

Out[10]: <seaborn.axisgrid.FacetGrid at 0x20871f85460>



APPENDIX

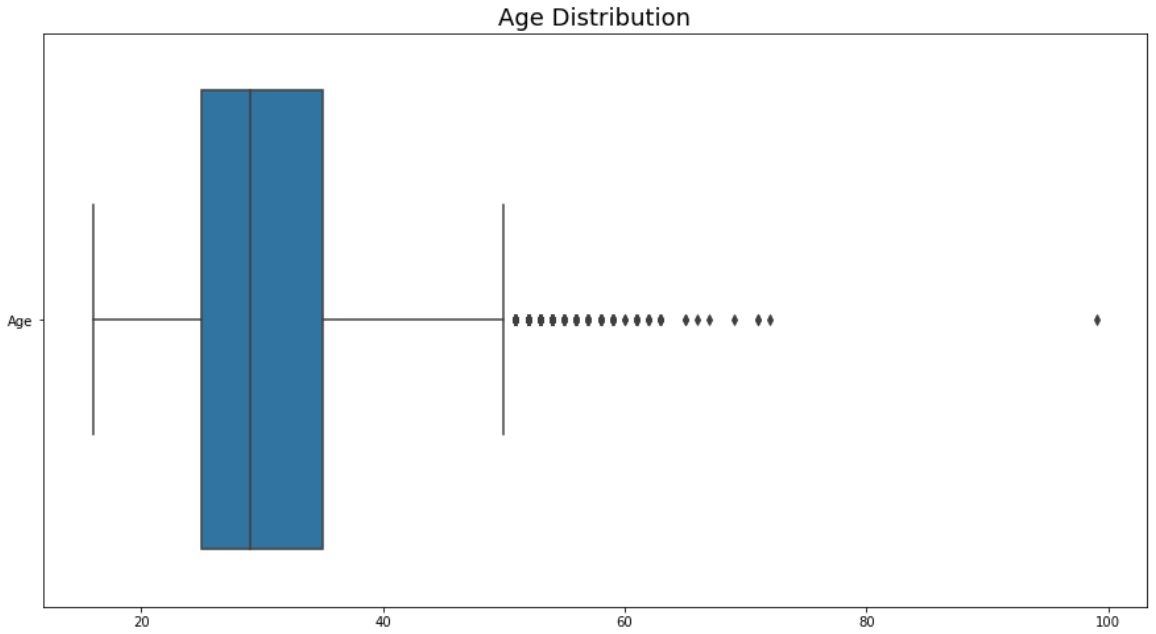
Box Plots

We plot a box plot of `Age.`

In [11]:

```
QUERY = """
SELECT Age
FROM master

"""
df = pd.read_sql_query(QUERY,conn)

print(df)

plt.rcParams['figure.figsize'] = [15,8]

ax = sns.boxplot(data = df, orient = 'h')
ax.set_title('Age Distribution', size = 18)

plt.show()
```

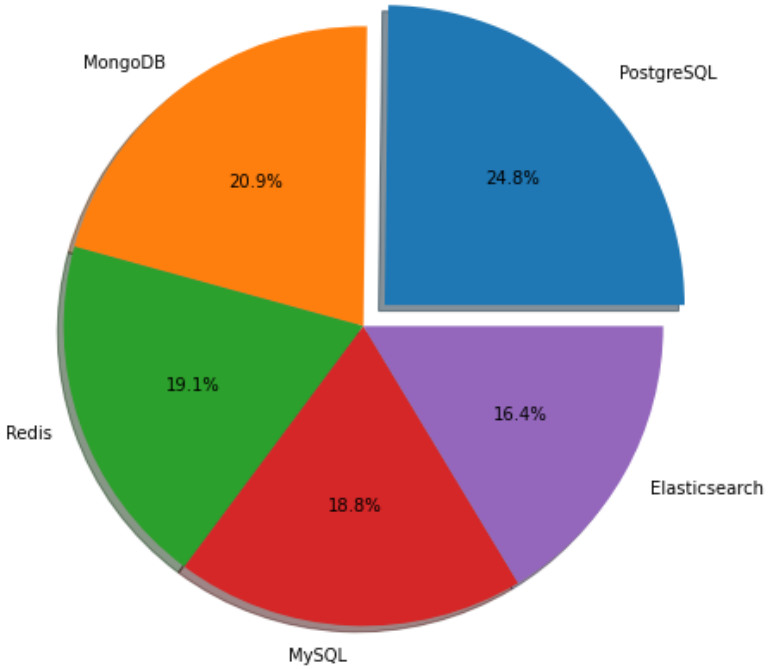
```
       Age
0      22.0
1      23.0
2      28.0
3      26.0
4      29.0
...     ...
11393  36.0
11394  25.0
11395  34.0
11396  25.0
11397  30.0

[11398 rows x 1 columns]
```



APPENDIX

In [26]:

```
QUERY = """
SELECT *, COUNT(*) AS count
FROM DatabaseDesireNextYear
GROUP BY DatabaseDesireNextYear
ORDER BY count DESC
LIMIT 5

"""
df = pd.read_sql_query(QUERY,conn)

data = df['count']
labels = df['DatabaseDesireNextYear']

plt.rcParams['figure.figsize'] = [8,8]

fig, ax = plt.subplots()
ax.pie(data, labels = labels, autopct='%1.1f%%', explode = (0.1, 0, 0, 0, 0), shadow = True)
plt.show()
```



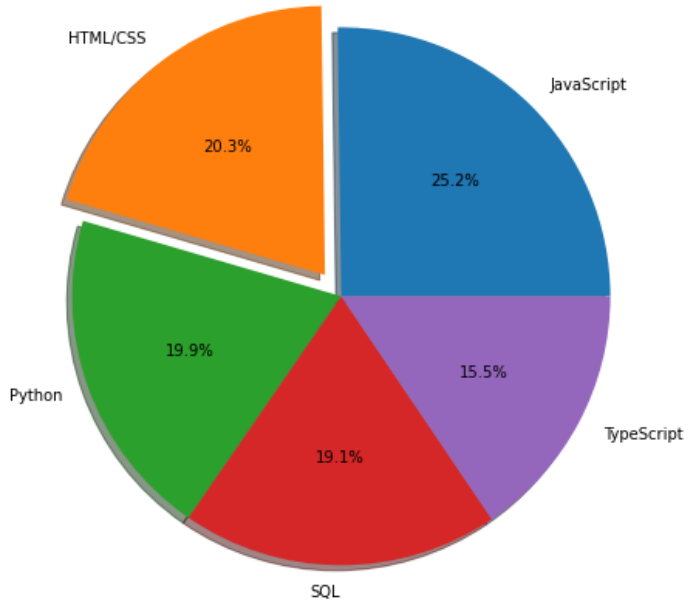
In the list of most popular languages respondents wish to learn next year, what is the rank of Python?

APPENDIX

In [24]:

```
QUERY = """
SELECT *, COUNT(*) AS count
FROM LanguageDesireNextYear
GROUP BY LanguageDesireNextYear
ORDER BY count DESC
LIMIT 5

"""
df = pd.read_sql_query(QUERY,conn)

print(df)

data = df['count']
labels = df['LanguageDesireNextYear']

plt.rcParams['figure.figsize'] = [8,8]

fig, ax = plt.subplots()
ax.pie(data, labels = labels, autopct='%1.1f%%', explode = (0, 0.1, 0, 0, 0), shadow = True)
plt.show()
```

```
   Respondent LanguageDesireNextYear  count
0           4              JavaScript   6630
1           9                HTML/CSS   5328
2          20                  Python   5239
3           4                     SQL   5012
4           9              TypeScript   4088
```

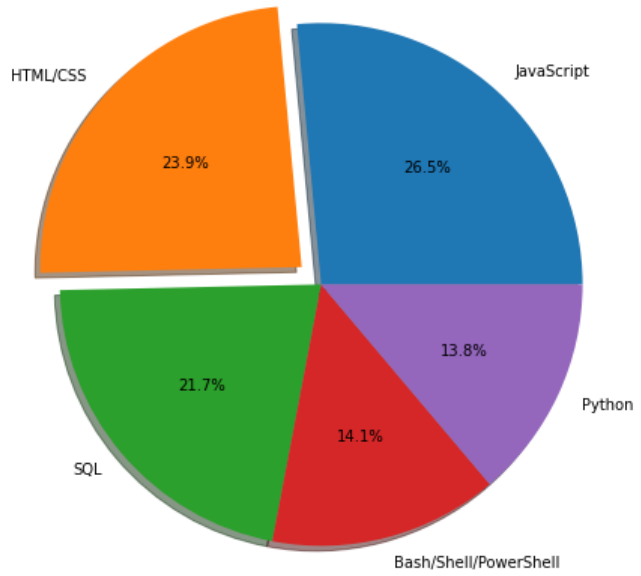


APPENDIX

In [29]:

```
# your code goes here

QUERY = """
SELECT *, COUNT(*) AS count
FROM LanguageWorkedWith
GROUP BY LanguageWorkedWith
ORDER BY count DESC
LIMIT 5

"""
df = pd.read_sql_query(QUERY,conn)

print(df)

data = df['count']
labels = df['LanguageWorkedWith']

plt.rcParams['figure.figsize'] = [8,8]

fig, ax = plt.subplots()
ax.pie(data, labels = labels, autopct='%1.1f%%', explode = (0, 0.1, 0, 0, 0), shadow = True)
plt.show()
```

	Respondent	LanguageWorkedWith	count
0	9	JavaScript	8687
1	9	HTML/CSS	7830
2	4	SQL	7106
3	9	Bash/Shell/PowerShell	4642
4	4	Python	4542



APPENDIX

Visualizing comparison of data

Line Chart

We plot the median `ConvertedComp` for all ages from 45 to 60.

In [43]:

```
QUERY = """
SELECT ConvertedComp, Age
FROM master
WHERE Age BETWEEN 45 AND 60

"""
df = pd.read_sql_query(QUERY,conn)

medians = df.groupby('Age')['ConvertedComp'].median()

data = pd.DataFrame(list(zip(medians.index, medians.values)),
                    columns =['Age', 'MedConvertedComp'])

plt.rcParams['figure.figsize'] = [8,8]

sns.lineplot(data=data, x='Age', y = 'MedConvertedComp')

plt.show()
```