

Graphical Data Representation System using Python and MySQL

Credits:
Raj Rishi Rana
Satyam Kumar

CERTIFICATE



This is to certify that Mr. Satyam Kumar of std. XII Navy Children School, Mumbai has put in sincere and dedicated efforts towards the project. His work has been highly appreciative.

Sign of Head of Department

Sign of Principal

School Stamp

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher Mrs. Sapna Malik as well as our Principal Dr.(Mrs) Nandana Khare who gave me the golden opportunity to do this wonderful project in computer science on the topic “Graphical Data Representation System using Python and MySQL”, which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to them. Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limit of time frame.

Table of Contents

- Cover Page
- Certificate
- Acknowledgement
- Index
- Software Specification
- Project Description
- Implementation
- Output
- Bibliography
- Future Scope

Software Specification

- **Programming Language:**

- **Front-end : Python 3.13**

Python is a high-level, versatile programming language known for its simplicity and readability, making it a popular choice for beginners and experts alike. Developed in the late 1980s by Guido van Rossum, Python emphasizes clear, concise syntax, allowing developers to focus on problem-solving rather than complex code structures. Its extensive standard library supports tasks ranging from web development and data analysis to artificial intelligence and scientific computing. Python's flexibility and adaptability stem from its compatibility with various libraries and frameworks, such as Django for web development, NumPy and Pandas for data manipulation, and TensorFlow for machine learning. With a large, supportive community and continuous development, Python has become a preferred language for a wide range of applications across industries, known for accelerating development and enhancing productivity.

- **Back-end : MySQL 8.0**

MySQL is a widely used open-source relational database management system (RDBMS) known for its speed, reliability, and flexibility. Developed in the mid-1990s by Swedish developers, MySQL quickly gained popularity due to its ease of use and powerful capabilities, especially in web applications and data management. MySQL is based on Structured Query Language (SQL), which allows users to create, manipulate, and manage databases effectively. It supports a wide range of platforms, including Windows, Linux, and macOS, and integrates seamlessly with popular programming languages like Python, PHP, and Java. MySQL is highly scalable, capable of handling large data volumes, and includes robust features for data security, transaction processing, and concurrency. Its popularity as the backbone of numerous web applications, including social media sites and e-commerce platforms, is supported by its active community and ongoing development under Oracle Corporation.

- **Modules Used:**
 - `tkinter`: For creating the graphical user interface.
 - `matplotlib.pyplot`: For plotting graphs.
 - `mysql.connector`: For connecting and querying MySQL databases.
 - `numpy`: For array handling.
 - `math`: For mathematical operations.
- **Database:** MySQL
- **IDE Used:** PyCharm (JetBrains IDEs)
- **Operating System:** Linux

Project Description

This project, "**Graphical Data Representation System using Python and MySQL**" aims to simplify **data visualization** for mathematical equations and database tables. The program uses a Tkinter GUI, where users can input either a mathematical equation or the name of a MySQL database and table. The program processes the input and generates a graphical representation of the data.

- **Objective:** To provide an easy-to-use tool for visualizing data and mathematical functions.
- **User Input Options:**
 - **Mathematical Equation:** The user enters a formula, and the program calculates values using `numpy` and plots the result.
 - **Database Table:** The user provides the name of a MySQL database and table, and the program retrieves data and plots it.

This system leverages Python's data processing and graphical capabilities to create a user-friendly experience for data visualization.

Implementation

1. Importing modules

```
1  import tkinter as tk
2  import mysql.connector as ml
3  import matplotlib.pyplot as plt
4  import numpy as np
5  from math import *
```

2. Initializing

```
8  # creating tkinter window
9  root = tk.Tk()
10 root.geometry("300x300")
11 root.title("Visualizer")
12
13 # setting up connection between python and MySQL
14 cn = ml.connect(host='localhost', user='root', passwd='1234')
15 curr = cn.cursor()
```


3. Defining functions

```
17 def start():
18     """Asking user what he wants to plot: Equation or Database"""
19     global what
20     what = tk.StringVar()
21     what.set("What do you want to plot?")
22     drop = tk.OptionMenu(root, what, value="Mathematical Equation", *values="MySQL database")
23     drop.pack()
24
25     tk.Button(root, text='Next', command=eqn_or_db).pack()
```

```
27 def eqn_or_db():
28     # e1 is error1 which is encountered when user didn't select any option from the first drop-down menu
29     e1 = tk.Label(root, text="Please select an option from the drop down menu !!!")
30     if what.get()=="Mathematical Equation":
31         try:
32             e1.destroy()
33         except: pass
34
35         inp_eqn()
36     elif what.get()=="MySQL database":
37         try:
38             e1.destroy()
39         except: pass
40
41         fetch_dbs()
42     else:
43         e1.pack()
44
```




```

45 def inp_eqn(): rajrish5541f
46     """This function is meant to input Mathematical Equation."""
47
48     # creating global variables
49     global eq, eq_st, eq_end
50     tk.Label(root, text='Enter equation:').pack()
51
52     # eq is the textbox where user will enter the equation
53     eq = tk.Entry(root, width=25)
54     eq.pack()
55     tk.Label(root, text='Define domain:').pack()
56
57     # domain is a tkinter frame which contains 2 labels and 2 textboxes
58     # where user will define the domain of the function
59     domain = tk.Frame(root)
60     tk.Label(domain, text='From: ').grid(row=0, column=0)
61     eq_st = tk.Entry(domain, width=5)
62     eq_st.grid(row=0, column=1)
63     tk.Label(domain, text='To: ').grid(row=0, column=2)
64     eq_end = tk.Entry(domain, width=5)
65     eq_end.grid(row=0, column=3)
66     domain.pack()
67
68     tk.Button(root, text='Plot', command=plot_eqn).pack()
69


```

```

70 def plot_eqn(): rajrish5541f
71     """This function is meant to plot the equation retrieved by the inp_eqn() function."""
72
73     # setting up variables which store the equation and domain
74     eqn = eq.get()
75     eqn_st = int(eq_st.get())
76     eqn_end = int(eq_end.get())
77
78     # xval is the numpy array of all the x-coordinates
79     xval = np.arange((eqn_st), eqn_end, 0.1)
80     x = []
81
82     # There is possibility that the function is not defined for some values of x
83     # Hence a for loop is required to separate out those x-coordinates
84     # this loop will split xval array into smaller arrays which hold the x-coordinates where function is defined
85     for i in xval:
86         try:
87             y = eval(eqn.replace(_old: 'x', _new: f"({i})"))
88             x.append(i)
89         except:
90             mlt.plot(*args: x, [eval(eqn.replace(_old: 'x', _new: f"({i})")) for i in x], 'b-')
91             x=[]
92     mlt.plot(*args: x, [eval(eqn.replace(_old: 'x', _new: f"({i})")) for i in x], 'b-', label=eqn)
93
94     mlt.legend()
95     # Finally showing up the graph
96     mlt.show()
97

```

```

98 def fetch_dbs(): rajrish5541f
99     """This function is meant to only fetch names of databases and return the name of chosen database"""
100
101     # setting up global variable
102     global dbname
103
104     # executing MySQL query
105     curr.execute('show databases;')
106
107     # creating a tkinter StringVar which will hold the name of chosen database
108     dbname = tk.StringVar()
109     dbname.set("Select database name")
110
111     # fetching the records
112     db1 = curr.fetchall()
113     dbs = [] # this will hold names of databases
114
115     # this for loop will create the list which will hold the names of all the databases
116     for i in db1:
117         dbs.append(i[0])
118     drop = tk.OptionMenu(root, dbname, *values: *dbs)
119     drop.pack()
120
121     # once databases are fetched, tables will be fetched
122     tk.Button(root, text='Next', command=fetch_tbs).pack()
123

```

```

124 def fetch_tbs(): 🐘rajrish5541f
125     """This function is used to fetch table names"""
126
127     # Declaring global variables
128     global tname
129
130     # creating a tkinter StringVar which will hold the name of chosen table
131     tname = tk.StringVar()
132     tname.set('Select table name')
133
134     # executing MySQL queries
135     curr.execute('use {};'.format(dbname.get()))
136     curr.execute('show tables;')
137     drop = tk.OptionMenu(root, tname, *values:[i[0] for i in curr.fetchall()])
138     drop.pack()
139
140     # if table names are fetched corretly then column names will be fetched
141     tk.Button(root, text='Next', command=fetch_clms).pack()
142

```

```

143 def fetch_clms(): 🐘rajrish5541f
144     """This function will fetch column names"""
145
146     # declaring global variables
147     global x,y,tname
148     tname = tname.get()
149
150     # executing MySQL query
151     curr.execute('describe {};'.format(tname.get()))
152     all_clms = curr.fetchall()
153     desirable = ['int', 'decimal', 'numeric', 'float', 'double']
154
155     # useable_clms is a list of only those column names which can be plotted, i.e which are numeric
156     useable_clms=[]
157     for i in desirable:
158         for j in all_clms:
159             if i in j[1]:
160                 useable_clms.append(j[0])
161
162     # select_clms is a tkinter frame which allows user to choose
163     # which columns to plot on x-axis and which one on y-axis
164     select_clms = tk.Frame(root)
165     x = tk.StringVar()
166     x.set('X-axis')
167     y = tk.StringVar()
168     y.set('Y-axis')
169     drop1 = tk.OptionMenu(select_clms, x, *values:*useable_clms)
170     drop1.grid(row=0,column=0)
171     drop2 = tk.OptionMenu(select_clms, y, *values:*useable_clms)
172     drop2.grid(row=0,column=1)
173     select_clms.pack()
174
175     # Once we get column names, we will plot the graph
176     tk.Button(root, text='Plot', command=plot_db).pack()
177

```

```

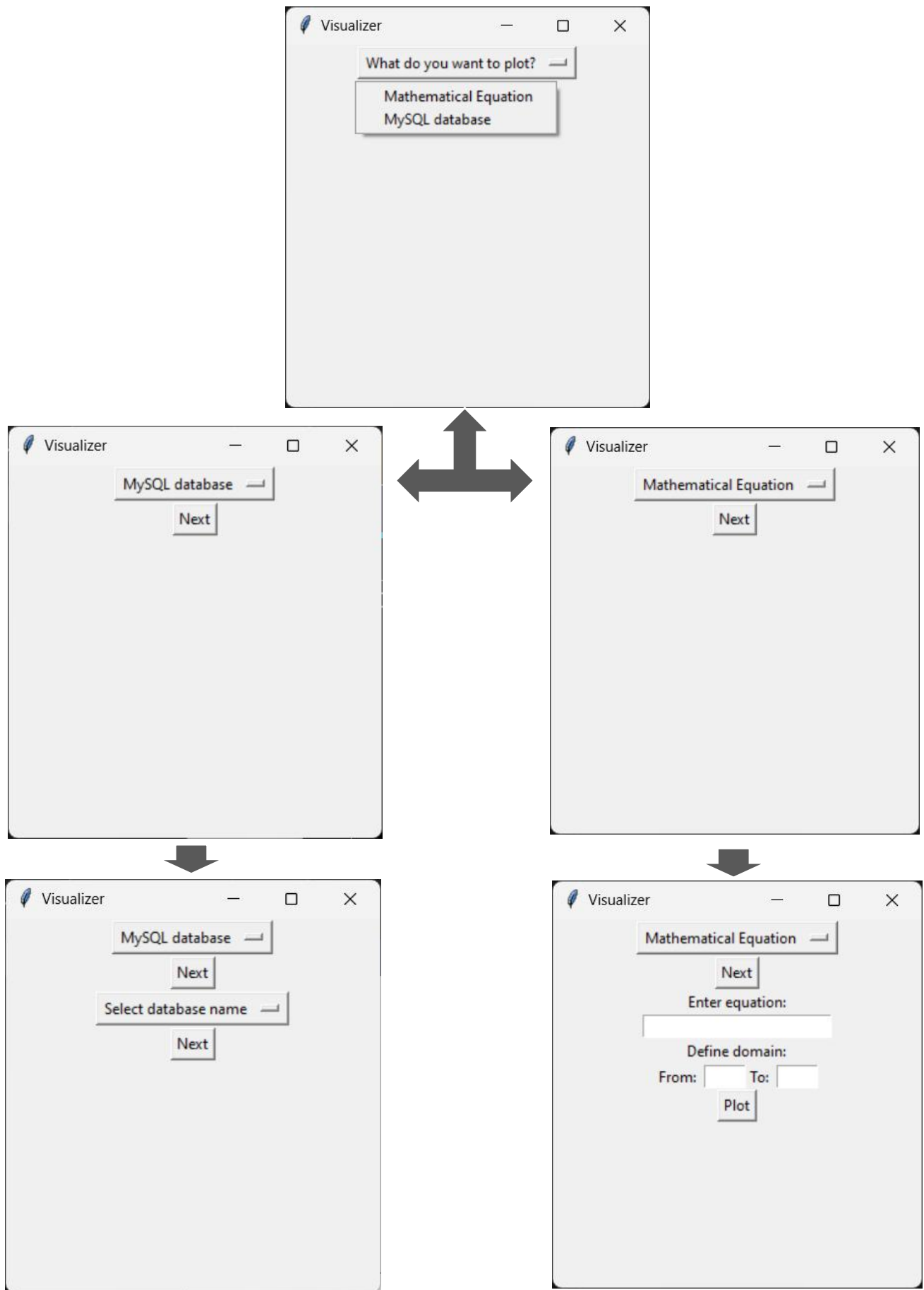
178 def plot_db():  🐞rajrish5541f
179     """This function is used to plot the graph of the db"""
180
181     # declaring variables
182     xaxis = x.get()
183     yaxis = y.get()
184
185     # executing query to get the records
186     curr.execute('select {},{} from {} order by {}'.format(*args:xaxis,yaxis,tname,xaxis))
187     entries = curr.fetchall()
188     xval, yval = [i[0] for i in entries], [i[1] for i in entries]
189
190     # This will plot the absolute coordinates
191     mlt.plot(*args:xval, yval, marker='o', linestyle='')
192
193     curr.execute('select {},{} from {} group by {} order by {}'.format(*args:xaxis,f'avg({yaxis})',tname,xaxis,xaxis))
194     entries = curr.fetchall()
195     xval, yval = [i[0] for i in entries], [i[1] for i in entries]
196
197     # This will plot the line of average values
198     mlt.plot(*args:xval,yval,'r--', label='Average')
199
200     # setting x and y labels for the graph
201     mlt.xlabel(xaxis)
202     mlt.ylabel(yaxis)
203
204     mlt.legend()
205     # showing the graph
206     mlt.show()
207
208 start()
209
210 root.mainloop()
211
212 '''
213 @Credits:
214     ● RAJ RISHI RANA aka rajrish5541f
215     ● SATYAM KUMAR
216 '''
217 🌟

```

Source code available at github.com -

[CS-11-12/Projects/Project Synopsis](https://github.com/rajrish5541f/CS-11-12/Projects/Project%20Synopsis) at main · rajrish5541f/CS-11-12

Output



Visualizer

MySQL database

Next

Select database name

- information_schema
- mysql
- performance_schema
- raj
- sakila
- sys
- world

Visualizer

Mathematical Equation

Next

Enter equation:

x^2

Define domain:

From: -10 To: 10

Plot

Visualizer

MySQL database

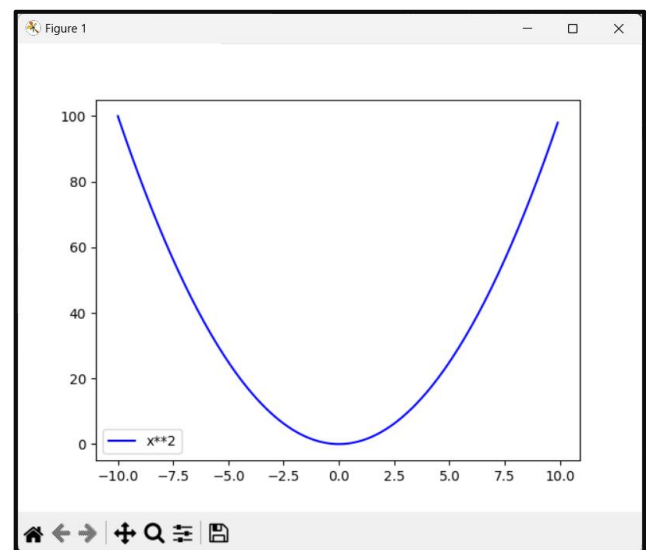
Next

raj

Next

Select table name

- employee
- stud



Visualizer

MySQL database

Next

raj

Next

employee

Next

X-axis

Y-axis

Plot

Visualizer

MySQL database

Next

raj

Next

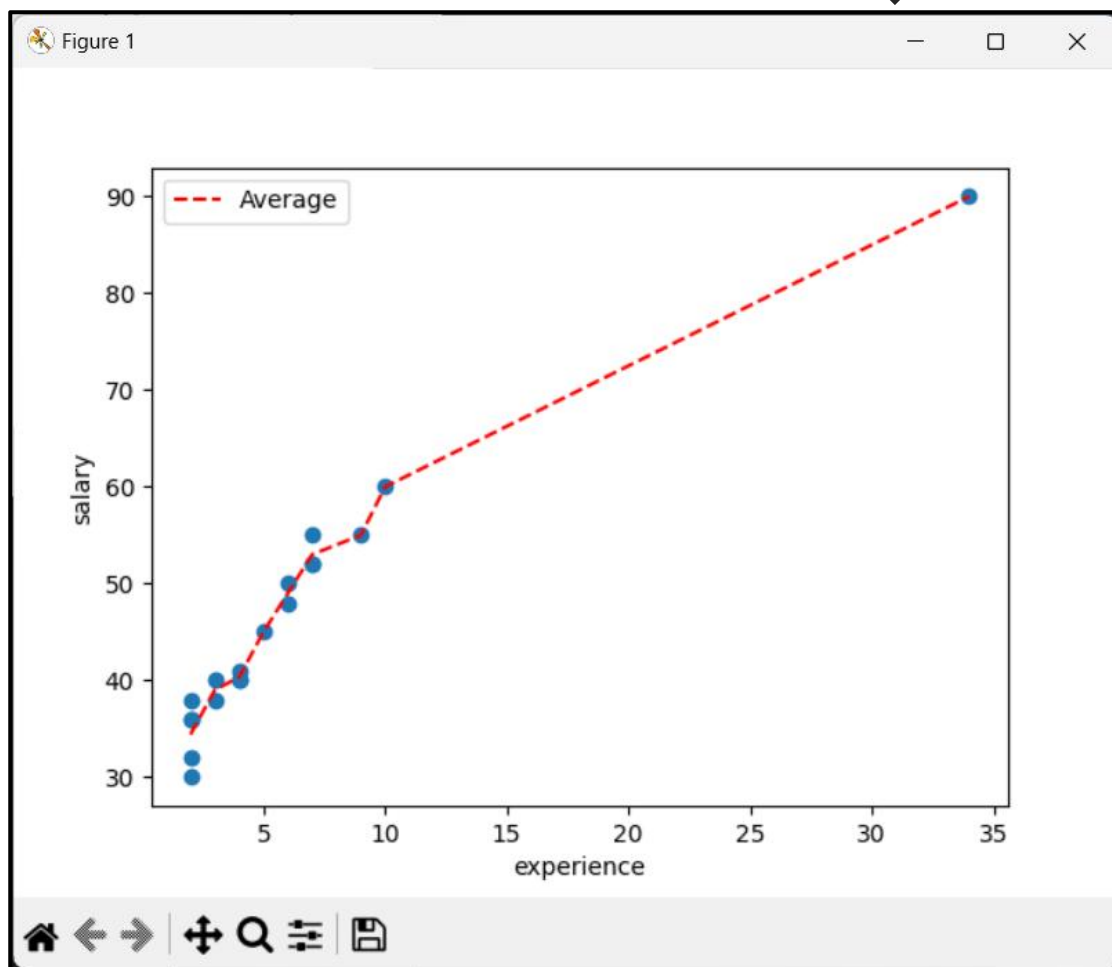
employee

Next

experience

salary

Plot



Bibliography

- **Python Documentation:** <https://docs.python.org/>
- **Matplotlib Documentation:** <https://matplotlib.org/>
- **MySQL Documentation:** <https://dev.mysql.com/>
- **Stack Overflow:** <https://stackoverflow.com/>
- **Youtube Handles:**
 - <https://youtube.com/@freecodecamp/>
 - <https://youtube.com/@SurfsharkAcademy/>
 - <https://youtube.com/@Codemycom/>
 - <https://youtube.com/@CodeWithHarry/>
 - <https://youtube.com/@CodingWithLewis/>

Future Scope

- **Enhanced Visualization Options:** Adding support for more types of plots like bar charts, pie charts, and histograms for varied data analysis.
- **Extended Database Support:** Expanding to support additional databases like PostgreSQL, SQLite, or cloud databases.
- **Complex Mathematical Expressions:** Allowing users to input more complex expressions with multiple variables.
- **Data Analysis Tools:** Incorporating basic data analysis functions (e.g., mean, median, mode) for database data to provide deeper insights.