

# Python Application

## Group 18

Janani Karthikeyan

+1-413-557-9761

[karthikeyan.j@northeastern.edu](mailto:karthikeyan.j@northeastern.edu)

Sneha Manjunath

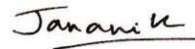
+1-857-891-3226

[chakrabhavi.s@northeastern.edu](mailto:chakrabhavi.s@northeastern.edu)

Percentage of Effort Contributed by Janani: 50%

Percentage of Effort Contributed by Sneha: 50%

Signature of Student 1:



Signature of Student 2:



Submission Date: 26 November 2023

```
In [1]: !pip install mysql-connector-python
```

```
Collecting mysql-connector-python
  Obtaining dependency information for mysql-connector-python from https://files.pythonhosted.org/packages/a7/84/b63f11124f808b6f1e3389072bc36cc907929d7574e85f94bf8f18117fe4/mysql\_connector\_python-8.2.0-cp311-cp311-win\_amd64.whl.metadata (https://files.pythonhosted.org/packages/a7/84/b63f11124f808b6f1e3389072bc36cc907929d7574e85f94bf8f18117fe4/mysql\_connector\_python-8.2.0-cp311-cp311-win\_amd64.whl.metadata)
  Downloading mysql_connector_python-8.2.0-cp311-cp311-win_amd64.whl.metadata (2.1 kB)
Collecting protobuf<=4.21.12,>=4.21.1 (from mysql-connector-python)
  Downloading protobuf-4.21.12-cp310-abi3-win_amd64.whl (527 kB)
----- 0.0/527.0 kB ? eta -:--:--
----- 71.7/527.0 kB 2.0 MB/s eta 0:00:01
----- 501.8/527.0 kB 6.3 MB/s eta 0:00:01
----- 527.0/527.0 kB 4.7 MB/s eta 0:00:00
Downloading mysql_connector_python-8.2.0-cp311-cp311-win_amd64.whl (14.2 MB)
----- 0.0/14.2 MB ? eta -:--:--
-- 1.0/14.2 MB 32.7 MB/s eta 0:00:01
----- 2.0/14.2 MB 25.7 MB/s eta 0:00:01
----- 2.4/14.2 MB 19.3 MB/s eta 0:00:01
----- 3.4/14.2 MB 19.5 MB/s eta 0:00:01
----- 4.2/14.2 MB 19.3 MB/s eta 0:00:01
----- 5.2/14.2 MB 19.7 MB/s eta 0:00:01
----- 6.2/14.2 MB 19.7 MB/s eta 0:00:01
----- 7.4/14.2 MB 20.5 MB/s eta 0:00:01
----- 8.5/14.2 MB 20.8 MB/s eta 0:00:01
----- 9.9/14.2 MB 21.9 MB/s eta 0:00:01
----- 11.0/14.2 MB 21.8 MB/s eta 0:00:01
----- 12.2/14.2 MB 21.8 MB/s eta 0:00:01
----- 13.1/14.2 MB 23.4 MB/s eta 0:00:01
----- 14.2/14.2 MB 23.4 MB/s eta 0:00:01
----- 14.2/14.2 MB 22.6 MB/s eta 0:00:01
----- 14.2/14.2 MB 19.2 MB/s eta 0:00:00
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.2.0 protobuf-4.21.12
```

```
In [30]: pip install pandas sqlalchemy matplotlib
```

```
Requirement already satisfied: pandas in c:\users\man_s\anaconda3\lib\site-packages (1.5.3)
Requirement already satisfied: sqlalchemy in c:\users\man_s\anaconda3\lib\site-packages (1.4.39)
Requirement already satisfied: matplotlib in c:\users\man_s\anaconda3\lib\site-packages (3.7.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\man_s\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\man_s\anaconda3\lib\site-packages (from pandas) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in c:\users\man_s\anaconda3\lib\site-packages (from pandas) (1.24.3)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\man_s\anaconda3\lib\site-packages (from sqlalchemy) (2.0.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\man_s\anaconda3\lib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\man_s\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\man_s\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\man_s\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\man_s\anaconda3\lib\site-packages (from matplotlib) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\man_s\anaconda3\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\man_s\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: six>=1.5 in c:\users\man_s\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
In [35]: pip install mysqlclient
```

```
Collecting mysqlclient
  Obtaining dependency information for mysqlclient from https://files.pythonhosted.org/packages/6d/48/571d99aa4634719c92d516c7bada1091ab85cebbd02fcf83f34b19f04c87/mysqlclient-2.2.0-cp311-cp311-win\_amd64.whl.metadata (https://files.pythonhosted.org/packages/6d/48/571d99aa4634719c92d516c7bada1091ab85cebbd02fcf83f34b19f04c87/mysqlclient-2.2.0-cp311-cp311-win\_amd64.whl.metadata)
  Downloading mysqlclient-2.2.0-cp311-cp311-win_amd64.whl.metadata (4.5 kB)
Downloading mysqlclient-2.2.0-cp311-cp311-win_amd64.whl (199 kB)
----- 0.0/200.0 kB ? eta -:--:--
----- 163.8/200.0 kB 3.3 MB/s eta 0:00:01
----- 200.0/200.0 kB 3.0 MB/s eta 0:00:00
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.2.0
Note: you may need to restart the kernel to use updated packages.
```

```

In [209]: #Example Query
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt

host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

def execute_query(query):
    cursor.execute(query)
    data = cursor.fetchall()
    return data

query = "SELECT * FROM organization"
data = execute_query(query)

columns = [desc[0] for desc in cursor.description]
df_ = pd.DataFrame(data, columns=columns)

print("EXAMPLE QUERY- ORGANIZATION TABLE OUTPUT:")
print()
print(df_organization)

cursor.close()
conn.close()

```

EXAMPLE QUERY- ORGANIZATION TABLE OUTPUT:

	ORGANIZATION_ID	O_NAME	O_LOCATION	O_CONTACT
0	SP961	Stellar Systems	Tokyo	309-645-1256
1	SP962	Galactic Enterprises	Mumbai	611-532-0829
2	SP963	Cosmic Ventures	Mumbai	852-367-0844
3	SP964	Celestial Innovations	Dubai	334-357-3610
4	SP965	Nebula Corporation	New York	737-123-5285
5	SP966	AstroTech	Dubai	842-404-0867
6	SP967	Starbound Solutions	Rio de Janeiro	389-308-4833
7	SP968	Lunar Enterprises	Rio de Janeiro	113-739-7324
8	SP969	Solar Nexus	Mumbai	408-146-8508
9	SP970	Interstellar Holdings	Mumbai	780-691-8487
10	SP971	Orion Industries	London	396-604-4017
11	SP972	Nova Enterprises	Tokyo	245-250-2192
12	SP973	Cosmos Corporation	New York	817-750-7309
13	SP974	Astro Dynamics	Dubai	764-143-7956
14	SP975	Galaxy Group	Tokyo	849-863-2692
15	SP976	Comet Enterprises	Dubai	172-120-1968
16	SP977	Saturn Solutions	Tokyo	786-965-4374
17	SP978	Andromeda Innovations	New York	202-433-5356
18	SP979	Milky Way Enterprises	Tokyo	744-497-0441
19	SP980	Solaris Corporation	Toronto	985-218-4721
20	SP981	Supernova Systems	Sydney	626-404-1661
21	SP982	Cosmic Connections	Paris	990-541-7190
22	SP983	Aurora Enterprises	Rio de Janeiro	873-238-8696
23	SP984	Pulsar Innovations	Tokyo	139-394-1417
24	SP985	Nebula Nexus	Sydney	395-177-3133
25	SP986	AstroTech Solutions	Dubai	202-363-5941
26	SP987	Starlight Holdings	London	449-547-9344
27	SP988	Celestial Innovations	Tokyo	689-426-3619
28	SP989	Galactic Nexus	Berlin	850-139-7035
29	SP990	Stellar Solutions	Berlin	805-556-9241

```

In [246]: #Query 1 - Bar Graph
host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT R.ROCKET_ID, R.R_NAME, SD.DEBRIS_ID, SUM(SD.D_MASS) AS TOTAL_D_MASS
FROM ROCKET R, SPACE_DEBRIS SD, PRESENT_IN P
WHERE R.ROCKET_ID = P.ROCKET_ID
AND P.DEBRIS_ID = SD.DEBRIS_ID
GROUP BY R.ROCKET_ID, R.R_NAME, SD.DEBRIS_ID;
"""

cursor.execute(query)
data = cursor.fetchall()

columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)

cursor.close()
conn.close()

plt.figure(figsize=(6, 4))
num_bars = len(df['DEBRIS_ID'])
cmap = get_cmap('Reds', len(df['ROCKET_ID'].unique()))
for i, (rocket_id, group_df) in enumerate(df.groupby('ROCKET_ID')):
    plt.bar(
        group_df['DEBRIS_ID'],
        group_df['TOTAL_D_MASS'],
        color = cmap(i),
        label=f"Rocket {rocket_id}"
    )

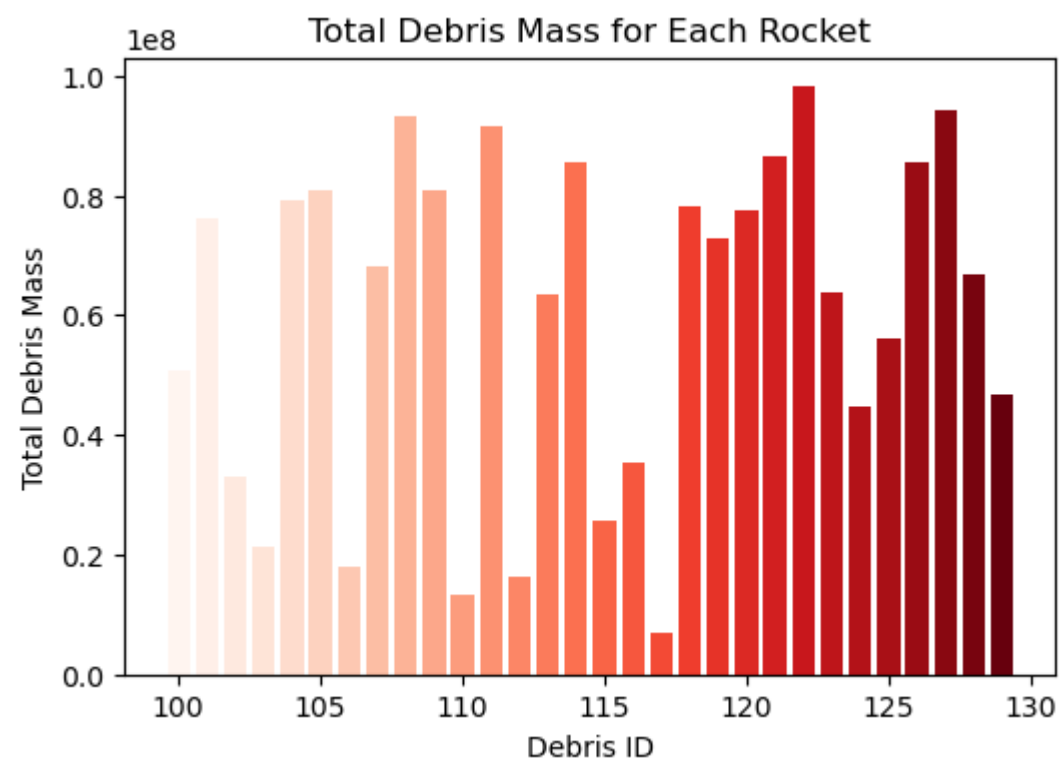
print("QUERY 1- BAR GRAPH OUTPUT:")
print()
print(df.to_string(index=False))

plt.xlabel('Debris ID')
plt.ylabel('Total Debris Mass')
plt.title('Total Debris Mass for Each Rocket')
plt.show()

```

QUERY 1- BAR GRAPH OUTPUT:

ROCKET_ID	R_NAME	DEBRIS_ID	TOTAL_D_MASS
1	Falcon 9	100	50627782.50
2	Atlas V	101	76311808.23
3	Delta IV	102	32977925.00
4	Soyuz	103	21346950.03
5	Long March 5	104	79321433.33
6	Proton-M	105	80919564.80
7	Ariane 5	106	17986562.98
8	H-IIA	107	68251580.56
9	GSLV Mk III	108	93130006.20
10	Vega	109	80973187.00
11	Antares	110	13123221.97
12	Electron	111	91455960.28
13	Minotaur	112	16255089.94
14	Pegasus	113	63556502.30
15	Starship	114	85563013.50
16	New Shepard	115	25787049.13
17	LauncherOne	116	35403329.54
18	Angara	117	6830670.02
19	Kuaizhou	118	78228495.64
20	Hyperbola-1	119	72915968.43
21	SLS	120	77411867.47
22	GSLV Mk II	121	86502044.13
23	Zenit	122	98140185.29
24	CZ-3B	123	63647112.24
25	CZ-5	124	44822320.03
26	CZ-7	125	56226598.80
27	CZ-11	126	85660326.93
28	CZ-2F	127	94237485.71
29	CZ-6	128	66692179.85
30	CZ-4B	129	46825808.11



```

In [200]: #Query 2 - Pie Chart
host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT S.S_STATUS, COUNT(S.SATELLITE_ID) AS SatelliteCount
FROM SATELLITE S
GROUP BY S.S_STATUS;
"""

cursor.execute(query)
data = cursor.fetchall()

columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)

cursor.close()
conn.close()

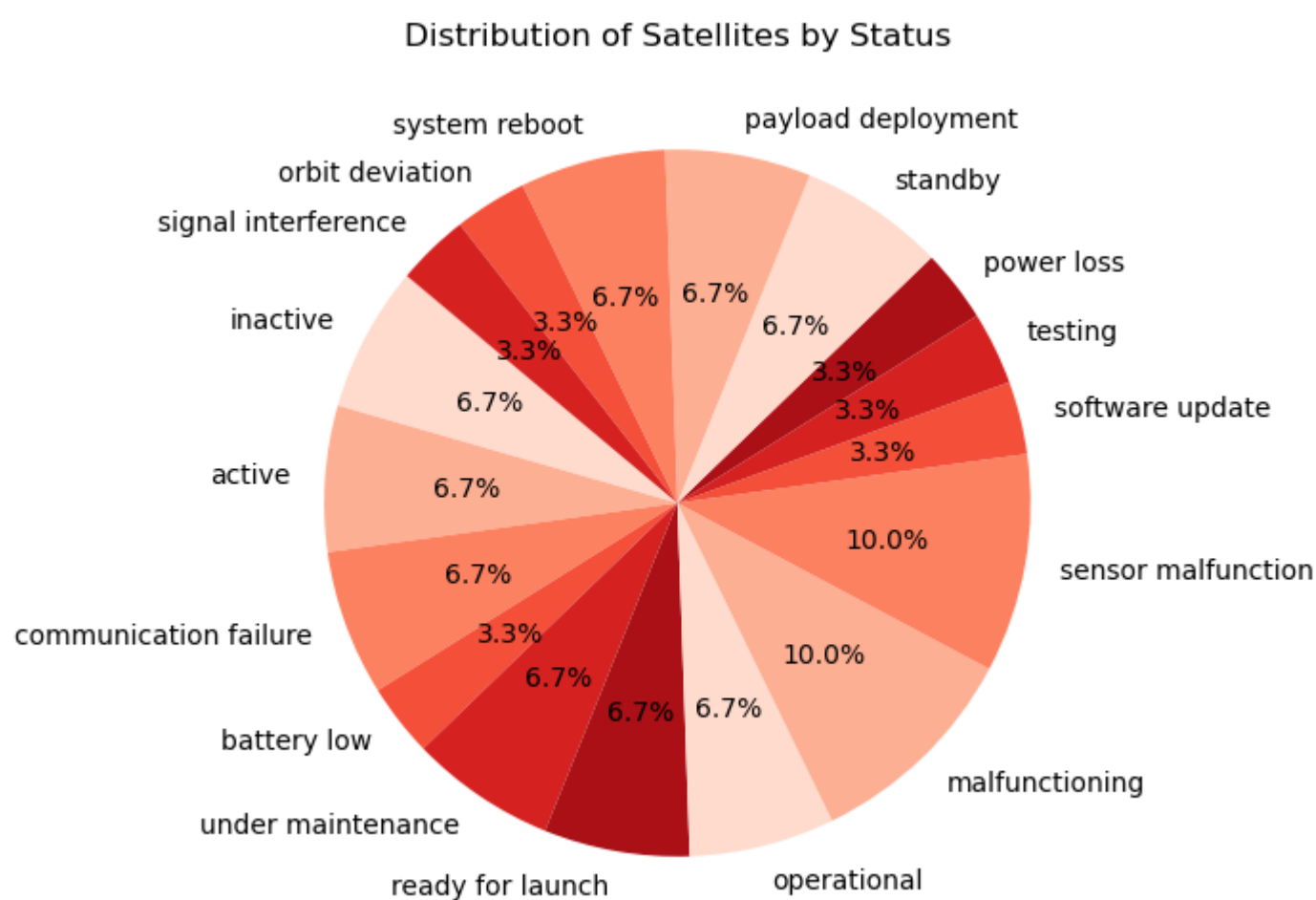
print("QUERY 2- PIE CHART OUTPUT:")
print()
print(df.to_string(index=False))

plt.figure(figsize=(6, 6))
plt.pie(df['SatelliteCount'], labels=df['S_STATUS'], autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Satellites by Status')
plt.show()

```

QUERY 2- PIE CHART OUTPUT:

S_STATUS	SatelliteCount
inactive	2
active	2
communication failure	2
battery low	1
under maintenance	2
ready for launch	2
operational	2
malfunctioning	3
sensor malfunction	3
software update	1
testing	1
power loss	1
standby	2
payload deployment	2
system reboot	2
orbit deviation	1
signal interference	1



```

In [245]: #Query 3 - Scatter Plot
host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT SEN_TYPE, AVG(SEN_DETECTION_RANGE) AS AvgDetectionRange
FROM SENSOR
GROUP BY SEN_TYPE;
"""

cursor.execute(query)
data = cursor.fetchall()

columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)

cursor.close()
conn.close()

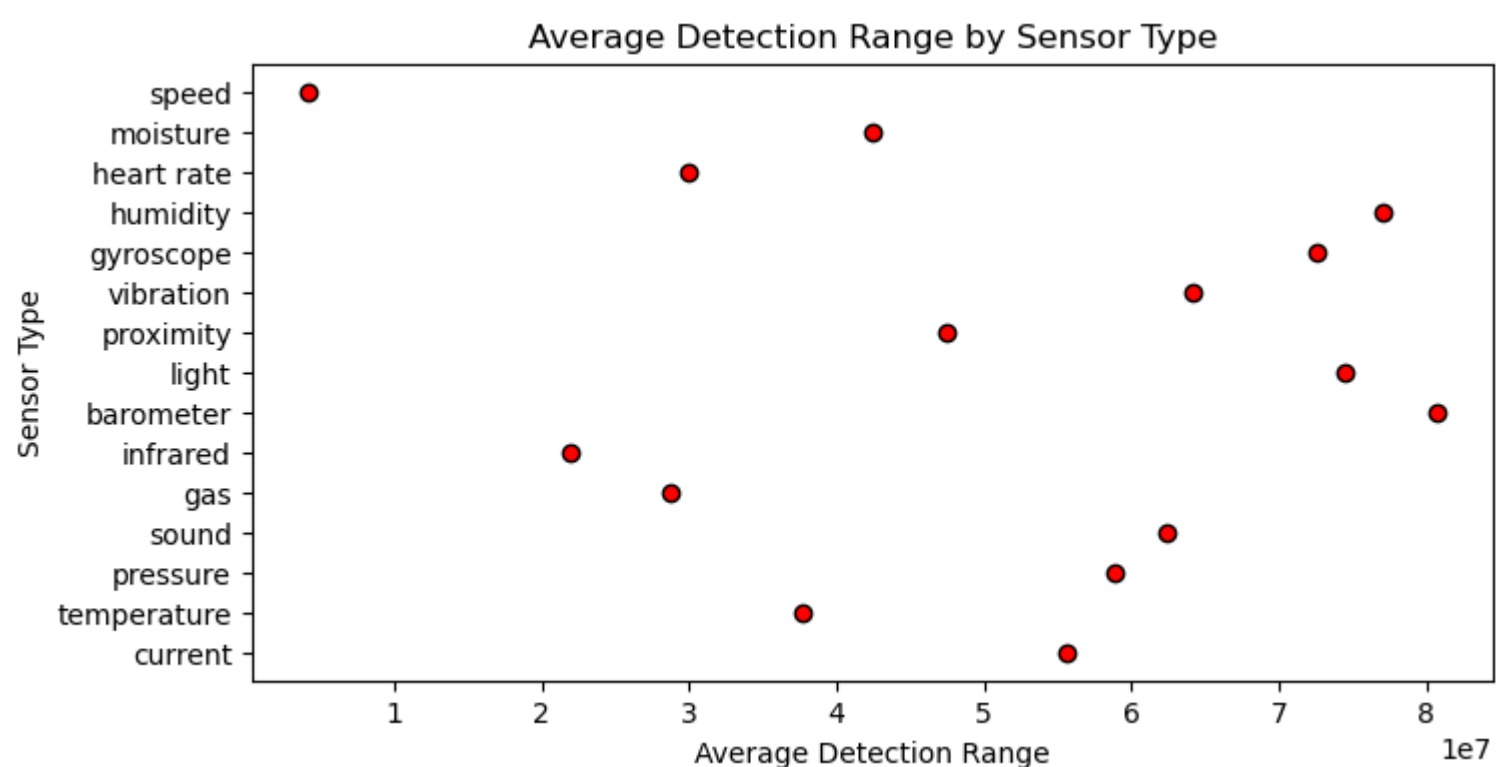
print("QUERY 3- SCATTER PLOT OUTPUT:")
print()
print(df.to_string(index=False))

plt.figure(figsize=(8, 4))
plt.scatter(df['AvgDetectionRange'], df['SEN_TYPE'], color='red', edgecolor='black') # Exchange x and y Labels
plt.xlabel('Average Detection Range')
plt.ylabel('Sensor Type')
plt.title('Average Detection Range by Sensor Type')
plt.show()

```

QUERY 3- SCATTER PLOT OUTPUT:

SEN_TYPE	AvgDetectionRange
current	55657661.215000
temperature	37674382.630000
pressure	58866465.065000
sound	62333420.543333
gas	28760070.063333
infrared	21966670.910000
barometer	80663383.075000
light	74416404.700000
proximity	47407092.070000
vibration	64098621.020000
gyroscope	72619183.020000
humidity	77097882.510000
heart rate	29935489.130000
moisture	42408760.090000
speed	4139282.560000



```
In [238]: #Query 4 - Bar Graph
host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
    SELECT R.ROCKET_ID, R.R_NAME
    FROM ROCKET R
    WHERE EXISTS (
        SELECT 1
        FROM PRESENT_IN P
        JOIN SENSOR S ON P.ROCKET_ID = R.ROCKET_ID
        WHERE S.SEN_DETECTION_FREQUENCY IS NOT NULL)
    LIMIT 15;
"""

cursor.execute(query)
data = cursor.fetchall()

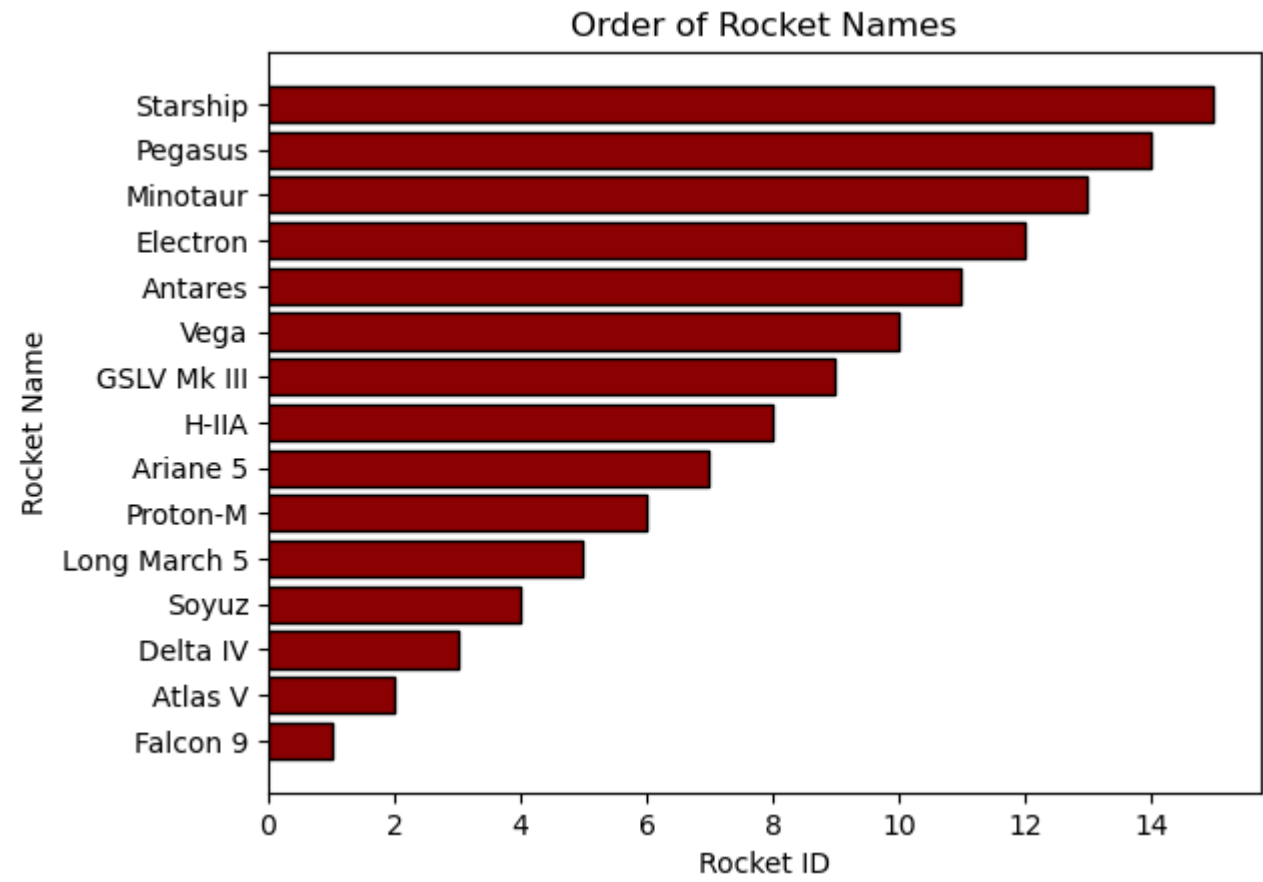
df = pd.DataFrame(data, columns=["ROCKET_ID", "R_NAME"])
plt.barh(df["R_NAME"], df["ROCKET_ID"],color='darkred',edgecolor='black')
plt.xlabel("Rocket ID")
plt.ylabel("Rocket Name")
plt.title("Order of Rocket Names")

print("QUERY 4- BAR GRAPH OUTPUT:")
print()
print(df.to_string(index=False))
plt.show()

cursor.close()
conn.close()
```

QUERY 4- BAR GRAPH OUTPUT:

ROCKET_ID	R_NAME
1	Falcon 9
2	Atlas V
3	Delta IV
4	Soyuz
5	Long March 5
6	Proton-M
7	Ariane 5
8	H-IIA
9	GSLV Mk III
10	Vega
11	Antares
12	Electron
13	Minotaur
14	Pegasus
15	Starship





```

In [244]: #Query 5 - Scatter Plot
host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
    SELECT R_NAME AS NEW_NAME, R_LAUNCH_DATE AS NEW_LAUNCH_DATE
    FROM ROCKET
    WHERE R_NAME IN ('Long March 5', 'Proton M', 'Soyuz', 'Electron','Minotaur', 'Pegasus')
    UNION
    SELECT S_NAME, S_LAUNCHED_DATE
    FROM SATELLITE
    WHERE SATELLITE_ID IN (
        SELECT DISTINCT S.SATELLITE_ID
        FROM SATELLITE S
        INNER JOIN LAUNCH_FACILITY F ON S.S_LAUNCHED_DATE = F.F_LAUNCHED_DATE
        WHERE F.F_LOCATION IN ('France', 'Turkey'))
    ORDER BY NEW_LAUNCH_DATE;
    """

cursor.execute(query)
data = cursor.fetchall()

df = pd.DataFrame(data, columns=["NEW_NAME", "NEW_LAUNCH_DATE"])
plt.figure(figsize=(8,4))
plt.scatter(df["NEW_LAUNCH_DATE"], df["NEW_NAME"], color='red',edgecolor='black')
plt.xlabel("Launch Date")
plt.ylabel("Rocket or Satellite Name")
plt.title("Scatter Plot of Rocket/Satellite Names by Launch Date")

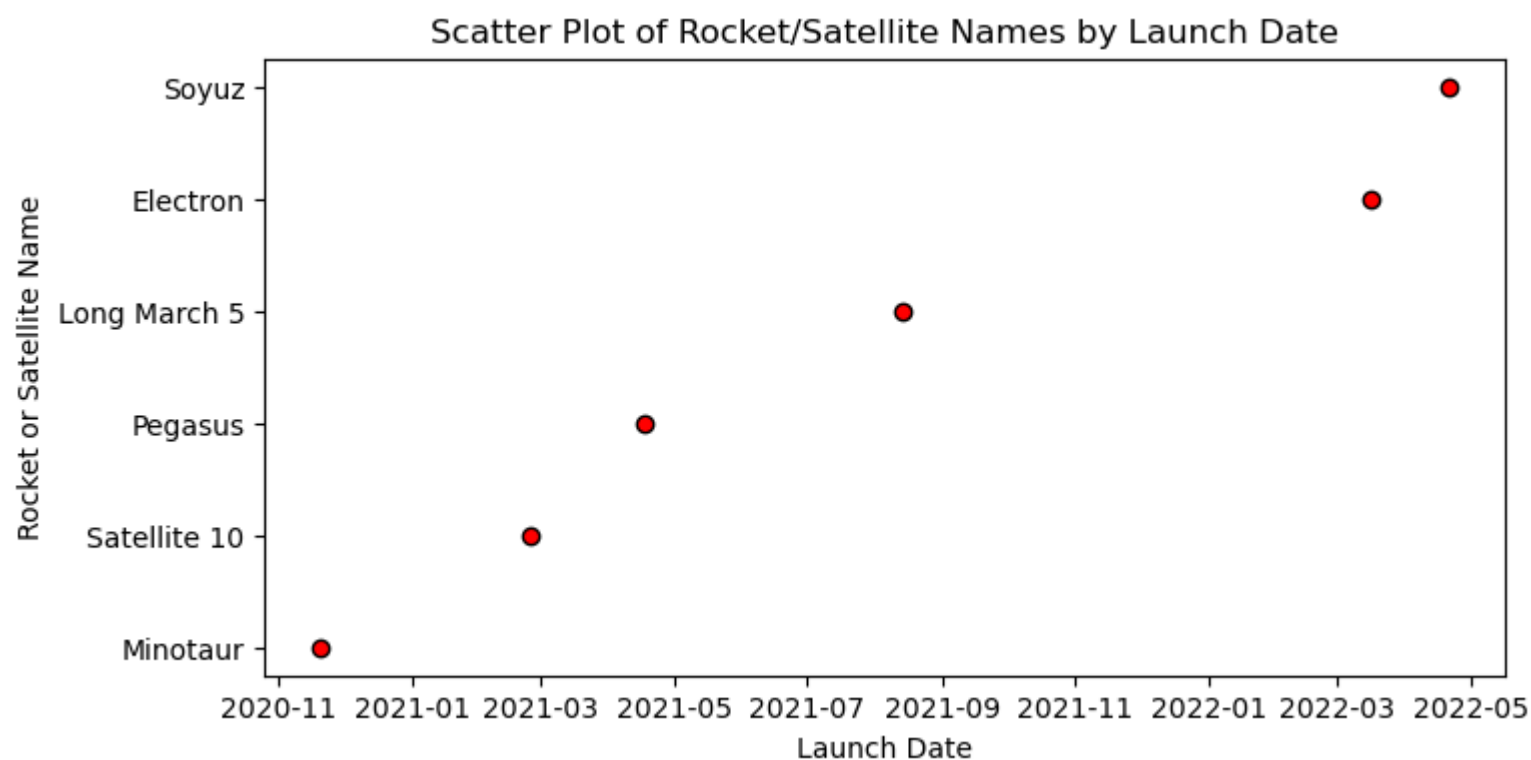
print("QUERY 5- SCATTER PLOT OUTPUT:")
print()
print(df.to_string(index=False))
plt.show()

cursor.close()
conn.close()

```

QUERY 5- SCATTER PLOT OUTPUT:

NEW_NAME	NEW_LAUNCH_DATE
Minotaur	2020-11-20
Satellite 10	2021-02-24
Pegasus	2021-04-17
Long March 5	2021-08-14
Electron	2022-03-16
Soyuz	2022-04-21



```

In [247]: #Query 6 - Bar Graph
host = "localhost"
port = 3305
user = "root"
password = "root"
database = "dma_project"

conn = mysql.connector.connect(host=host, port=port, user=user, password=password, database=database)
cursor = conn.cursor()

query = """
SELECT R_NAME AS NEW_NAME, R_LAUNCH_DATE AS NEW_LAUNCH_DATE
FROM ROCKET
WHERE R_NAME IN ('Long March 5', 'Proton M', 'Soyuz', 'Electron', 'Minotaur', 'Pegasus')
UNION
SELECT S_NAME, S_LAUNCHED_DATE
FROM SATELLITE
WHERE SATELLITE_ID IN (
    SELECT DISTINCT S.SATELLITE_ID
    FROM SATELLITE S
    INNER JOIN LAUNCH_FACILITY F ON S.S_LAUNCHED_DATE = F.F_LAUNCHED_DATE
    WHERE F.F_LOCATION IN ('Turkey', 'France', 'Russia', 'China', 'Germany', 'Brazil'))
ORDER BY NEW_LAUNCH_DATE;
"""

cursor.execute(query)
data = cursor.fetchall()

df = pd.DataFrame(data, columns=["NEW_NAME", "NEW_LAUNCH_DATE"])
df['NEW_LAUNCH_DATE'] = pd.to_datetime(df['NEW_LAUNCH_DATE'])
df['LaunchYear'] = df['NEW_LAUNCH_DATE'].dt.year
launch_counts = df['LaunchYear'].value_counts().sort_index()
plt.bar(launch_counts.index, launch_counts, color='darkred', edgecolor='black')
plt.xlabel("Launch Year")
plt.ylabel("Number of Launches")
plt.title("Number of Launches Per Year")

print("QUERY 6- BAR GRAPH OUTPUT:")
print()
print(df.to_string(index=False))
plt.show()

cursor.close()
conn.close()

```

QUERY 6- BAR GRAPH OUTPUT:

NEW_NAME	NEW_LAUNCH_DATE	LaunchYear
Minotaur	2020-11-20	2020
Satellite 10	2021-02-24	2021
Pegasus	2021-04-17	2021
Long March 5	2021-08-14	2021
Electron	2022-03-16	2022
Soyuz	2022-04-21	2022

