SQL Task

Assignment 11

1. Retrieve the total sales for each employee:

- **Clean SalseAmount**

```
SET SQL_SAFE_UPDATES=0;
update sales_data
set SalesAmount = cast(replace(replace(SalesAmount, '$',''),',','') as decimal(10,2));
Select * From sales_data;
```

- **Output**

TransactionID	CustomerID	SalesAmount	PurchaseDate	EmployeeID
T00001	C0001	2824.00	01-11-2022	1
T00002	C0002	1409.00	19-11-2023	2
T00003	C0003	5506.00	09-10-2023	3
T00004	C0004	5012.00	04-02-2022	3
T00005	C0005	4657.00	28-10-2024	7
T00006	C0006	3286.00	06-03-2021	10
T00007	C0007	2679.00	26-09-2022	2
T00008	C0008	9935.00	09-03-2020	7
T00009	C0009	2424.00	03-04-2023	7
T00010	C0010	7912.00	18-02-2023	10
T00011	C0011	1520.00	06-12-2022	8
T00012	C0012	1488.00	15-10-2023	9

- **Query1:**

```
select e.name as employeeName,
sum(s.SalesAmount) as TotalSales
from employees e
left join sales_data s on e.employeesID = s.employeeID
group by e.employeesID,e.name;
```

- **Output:**

	employeeName	TotalSales
▶	John Doe	49588
	Jane Smith	74641
	Alice Brown	55560
	Bob Johnson	66747
	Charlie Lee	47589

2. Identify the top-performing employee in terms of sales:

- **Query2:**

```
select e.name as employeeName, sum(s.SalesAmount) as TotalSales
from employees e
left join sales_data s on s.EmployeeID= e.employeesID
group by e.name
order by TotalSales desc
limit 1;
```

- **Output:**

	employeeName	TotalSales
▶	Jane Smith	74641

3. Find departments with the highest total salaries:

- **Query3:**

```
select d.departmentName, sum(e.salary) as TotalSalaries
from employees e
left join departments d on e.departmentID = d.departmentID
group by d.departmentName
order by TotalSalaries desc;
```

- **Output:**

departmentName	TotalSalaries
Marketing	125000.00
Sales	120000.00
HR	55000.00

4. List each employee and their supervisor's name:

- **Query4:**

```
set sql_safe_updates=0;
update employees set supervisorID =3 where employeesID =1;
update employees set supervisorID =3 where employeesID =4;
```

- **Output:**

(Missing data)

5. Retrieve customers with purchases above $10,000:

- **Query5:**

```
select customerID, SalesAmount as Purchases
from sales_data
having Purchases>10000;
```

- **Output**

customerID	Purchases

- **second answer**

```
select customerID, SalesAmount as Purchases
from sales_data
having Purchases>1000;
```

- **Output**

	customerID	Purchases
▶	C0001	2824.00
	C0002	1409.00
	C0003	5506.00
	C0004	5012.00
	C0005	4657.00

Python Task

1. Import Sales Data as csv file

- **Query**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df_Sales=pd.read_csv('/content/sales_data.csv')
df_Sales.head()
```

- **output**

	TransactionID	CustomerID	SalesAmount	PurchaseDate	EmployeeID
0	T00001	C0001	$2,824	01-11-2022	1
1	T00002	C0002	$1,409	19-11-2023	2
2	T00003	C0003	$5,506	09-10-2023	3
3	T00004	C0004	$5,012	04-02-2022	3
4	T00005	C0005	$4,657	28-10-2024	7

2. Inserting Data of Employee Table

- **Query**

```
data_EMP= {'EmployesID':["1","2","3","4","5"],
        'name':["Jhon doe","Jhan Smith","Alice Brown","bob jhonson", "charlie lie"],
        'DepID':['101','102','101','103','102'],
        'Salary':[50000,60000,70000,55000,65000],
            'suprvisorID':['3','','','3','']}
df=pd.DataFrame(data_EMP)
df
```

- **Output**

	EmployesID	name	DepID	Salary	suprvisorID
0	1	Jhon doe	101	50000	3
1	2	Jhan Smith	102	60000	
2	3	Alice Brown	101	70000	
3	4	bob jhonson	103	55000	3
4	5	charlie lie	102	65000	

3. Cleaning Sales Data

#1

- **Query**

```
df_Sales['SalesAmount']=df_Sales['SalesAmount'].replace ({'\$':'',',':''}, regex=True )
df_Sales['SalesAmount']=pd.to_numeric(df_Sales['SalesAmount'], errors='coerce')
df_Sales['SalesAmount'].dtype
df_Sales['SalesAmount'].fillna(0, inplace= True )
df_Sales['SalesAmount']
df_Sales.head()
```

- **Output**

	TransactionID	CustomerID	SalesAmount	PurchaseDate	EmployeeID
0	T00001	C0001	2824	01-11-2022	1
1	T00002	C0002	1409	19-11-2023	2
2	T00003	C0003	5506	09-10-2023	3
3	T00004	C0004	5012	04-02-2022	3
4	T00005	C0005	4657	28-10-2024	7

#2

- **Query**

```
df_Sales['PurchaseDate']= pd.to_datetime(df_Sales['PurchaseDate'], errors='coerce')
df_Sales['PurchaseDate']
```

- **Output**

	PurchaseDate
0	2022-01-11
1	NaT
2	2023-09-10
3	2022-04-02
4	NaT
...	...
95	NaT
96	NaT
97	NaT
98	NaT
99	NaT

4. Cleaning Employee Data

- **1# Query**

```
# Converting emp id to numeric
data_EMP['EmployesID'] = pd.to_numeric(data_EMP['EmployesID'], errors='coerce')
print(data_EMP['EmployesID'].dtype)
```

- **Output**

```
int64
```

#2

- **Query**

```
# converting Sup ID to int
data_EMP = pd.DataFrame(data_EMP)
data_EMP['suprvisorID'] = pd.to_numeric(data_EMP['suprvisorID'], errors='coerce')
data_EMP['suprvisorID'] = data_EMP['suprvisorID'].fillna(0).astype(int)
data_EMP
```

- **Output**

	EmployesID	name	DepID	Salary	suprvisorID
0	1	Jhon doe	101	50000	3
1	2	Jhan Smith	102	60000	0
2	3	Alice Brown	101	70000	0
3	4	bob jhonson	103	55000	3
4	5	charlie lie	102	65000	0

5. Sorting Employee names by salary

- **Query**

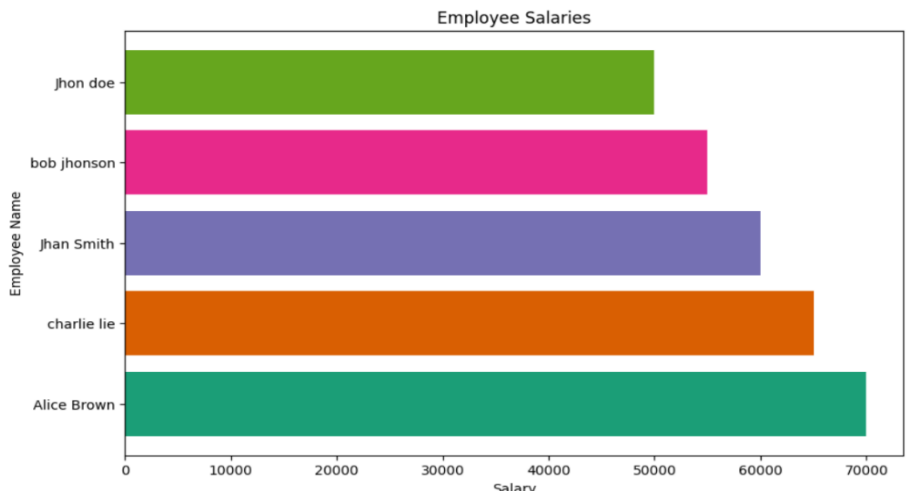
```
# Sort data by Salary
data_EMP_sorted = data_EMP.sort_values(by='Salary', ascending=False)

# Create horizontal bar chart
plt.figure(figsize=(10, 6))
plt.barh(data_EMP_sorted['name'], data_EMP_sorted['Salary'], color=sns.color_palette('Dark2'))

# Labels and title
plt.xlabel("Salary")
plt.ylabel("Employee Name")
plt.title("Employee Salaries")

# Show the plot
plt.show()
```

- **Output**



6. Save the cleaned Data

- **Query**

```
# Save cleaned sales dataset
df_Sales.to_csv("cleaned_sales.csv", index=False)

# Save cleaned employee dataset
data_EMP.to_csv("cleaned_employees.csv", index=False)
```