GROUP 3: Hospital patient records Project

Part 1: Data Cleaning (SQL and Python)

1. Load the Dataset

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	Name	Age	Gender	Diagnosis	Medication	AdmissionDate	DischargeDate	Doctor	Department	Status
2001	John Doe	45	Male	Hypertension	Med A	2025-01-10	2025-01-15	Dr. Smith	Cardiology	Admitted
2002	Jane Smith	forty-five	Female	Diabetes	Med B	10/01/2025	15/01/2025	Dr. Lee	Endocrinology	admitted
2003	Bob Brown	55	Male	Asthma	Med C	January 10, 2025	January 15, 2025	Dr. Carter	Pulmonology	Under Observation
2004		30	Female	Flu	Med D	2025-02-05	2025-02-10		General Medicine	DISCHARGED
2005	Tom Wilson	62	Male	Heart Disease	Med E	2025-03-01	2025-03-10	Dr. Johnson	Cardiology	Discharged
2006	Susan Clark	49	Female	Kidney Disease	Med F	2025-04-12	2025-04-17	Dr. Patel	Nephrology	Admitted
2007	David Jones	37	Male	Pneumonia	Med G	2025-05-20	2025-05-25	Dr. Martinez	Pulmonology	admitted
2008	Nancy Miller	28	Female	Flu	Med D	2025-06-15	2025-06-20	Dr. Smith	General Medicine	Under Observation
2009	Michael Scott	40	Male	Hypertension	Med A	2025-07-01	2025-07-07	Dr. Smith	Cardiology	Admitted
2010	Pam Beesly	34	Female	Diabetes	Med B	2025-08-10	2025-08-15	Dr. Lee	Endocrinology	Discharged

2. Data Cleaning

SQL Tasks:

1) **Remove duplicate records:**

```sql
SET SQL_SAFE_UPDATES = 0;
# 1-Remove duplicate records.
WITH CTE AS (
    SELECT
        *,
        ROW_NUMBER() OVER (PARTITION BY PatientID ORDER BY AdmissionDate) AS rn
    FROM hospital_patient_records
)
DELETE FROM hospital_patient_records
WHERE PatientID IN (
    SELECT PatientID FROM CTE WHERE rn > 1
);
SET SQL_SAFE_UPDATES = 1;
SELECT * From hospital_patient_records;
```

2) **Standardize the AdmissionDate column to a consistent format:**

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	Name	Age	Gender	Diagnosis	Medication	AdmissionDate	DischargeDate	Doctor	Department	Status
2001	John Doe	45	Male	Hypertension	Med A	2025-01-10	2025-01-15	Dr. Smith	Cardiology	Admitted
2002	Jane Smith	forty-five	Female	Diabetes	Med B	2025-01-10	15/01/2025	Dr. Lee	Endocrinology	admitted
2003	Bob Brown	55	Male	Asthma	Med C	2025-01-10	January 15, 2025	Dr. Carter	Pulmonology	Under Observation
2004		30	Female	Flu	Med D	2025-02-05	2025-02-10		General Medicine	DISCHARGED
2005	Tom Wilson	62	Male	Heart Disease	Med E	2025-03-01	2025-03-10	Dr. Johnson	Cardiology	Discharged
2006	Susan Clark	49	Female	Kidney Disease	Med F	2025-04-12	2025-04-17	Dr. Patel	Nephrology	Admitted
2007	David Jones	37	Male	Pneumonia	Med G	2025-05-20	2025-05-25	Dr. Martinez	Pulmonology	admitted
2008	Nancy Miller	28	Female	Flu	Med D	2025-06-15	2025-06-20	Dr. Smith	General Medicine	Under Observation
2009	Michael Scott	40	Male	Hypertension	Med A	2025-07-01	2025-07-07	Dr. Smith	Cardiology	Admitted
2010	Pam Beesly	34	Female	Diabetes	Med B	2025-08-10	2025-08-15	Dr. Lee	Endocrinology	Discharged

```sql
# 2-Standardize the AdmissionDate column to a consistent format (e.g., YYYY-MM-DD).

SET SQL_SAFE_UPDATES = 0;
UPDATE hospital_patient_records
SET
    AdmissionDate = CASE
        WHEN AdmissionDate LIKE '%/%' THEN STR_TO_DATE(AdmissionDate, '%d/%m/%Y')
        WHEN AdmissionDate LIKE '%,%' THEN STR_TO_DATE(AdmissionDate, '%M %d, %Y')
        ELSE AdmissionDate
    END;
SELECT * From hospital_patient_records;
SET SQL_SAFE_UPDATES = 1;
```

3) **Convert the Age column to a numeric type:**

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	Name	Age	Gender	Diagnosis	Medication	AdmissionDate	DischargeDate	Doctor	Department	Status
2001	John Doe	45	Male	Hypertension	Med A	2025-01-10	2025-01-15	Dr. Smith	Cardiology	Admitted
2002	Jane Smith	45	Female	Diabetes	Med B	2025-01-10	15/01/2025	Dr. Lee	Endocrinology	admitted
2003	Bob Brown	55	Male	Asthma	Med C	2025-01-10	January 15, 2025	Dr. Carter	Pulmonology	Under Observation
2004		30	Female	Flu	Med D	2025-02-05	2025-02-10		General Medicine	DISCHARGED
2005	Tom Wilson	62	Male	Heart Disease	Med E	2025-03-01	2025-03-10	Dr. Johnson	Cardiology	Discharged
2006	Susan Clark	49	Female	Kidney Disease	Med F	2025-04-12	2025-04-17	Dr. Patel	Nephrology	Admitted
2007	David Jones	37	Male	Pneumonia	Med G	2025-05-20	2025-05-25	Dr. Martinez	Pulmonology	admitted
2008	Nancy Miller	28	Female	Flu	Med D	2025-06-15	2025-06-20	Dr. Smith	General Medicine	Under Observation
2009	Michael Scott	40	Male	Hypertension	Med A	2025-07-01	2025-07-07	Dr. Smith	Cardiology	Admitted
2010	Pam Beesly	34	Female	Diabetes	Med B	2025-08-10	2025-08-15	Dr. Lee	Endocrinology	Discharged

```sql
# 3-Convert the Age column to a numeric type.

SET SQL_SAFE_UPDATES = 0;
UPDATE hospital_patient_records
SET Age = CASE
    WHEN Age = 'forty-five' THEN 45
    ELSE CAST(Age AS UNSIGNED)
END;
SET SQL_SAFE_UPDATES = 1;
```

4) Normalize the Status column to lowercase:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	Name	Age	Gender	Diagnosis	Medication	AdmissionDate	DischargeDate	Doctor	Department	Status
2001	John Doe	45	Male	Hypertension	Med A	2025-01-10	2025-01-15	Dr. Smith	Cardiology	admitted
2002	Jane Smith	45	Female	Diabetes	Med B	2025-01-10	15/01/2025	Dr. Lee	Endocrinology	admitted
2003	Bob Brown	55	Male	Asthma	Med C	2025-01-10	January 15, 2025	Dr. Carter	Pulmonology	under observation
2004		30	Female	Flu	Med D	2025-02-05	2025-02-10		General Medicine	discharged
2005	Tom Wilson	62	Male	Heart Disease	Med E	2025-03-01	2025-03-10	Dr. Johnson	Cardiology	discharged
2006	Susan Clark	49	Female	Kidney Disease	Med F	2025-04-12	2025-04-17	Dr. Patel	Nephrology	admitted
2007	David Jones	37	Male	Pneumonia	Med G	2025-05-20	2025-05-25	Dr. Martinez	Pulmonology	admitted
2008	Nancy Miller	28	Female	Flu	Med D	2025-06-15	2025-06-20	Dr. Smith	General Medicine	under observation
2009	Michael Scott	40	Male	Hypertension	Med A	2025-07-01	2025-07-07	Dr. Smith	Cardiology	admitted
2010	Pam Beesly	34	Female	Diabetes	Med B	2025-08-10	2025-08-15	Dr. Lee	Endocrinology	discharged

```
58    # 4-Normalize the Status column to lowercase.
59 •  SET SQL_SAFE_UPDATES = 0;
60 •  UPDATE hospital_patient_records
61    SET Status = LOWER(Status);
62 •  SET SQL_SAFE_UPDATES = 1;
63
```

5) Replace missing values in the Age and Diagnosis columns with default values:

```
63
64    # 5-Replace missing values in the Age and Diagnosis columns with default values (e.g., Unknown).
65
66 •  SET SQL_SAFE_UPDATES = 0;
67 •  UPDATE hospital_patient_records
68    SET
69      Diagnosis = COALESCE(NULLIF(Diagnosis, ''), 'Unknown'),
70      Age = COALESCE(NULLIF(Age, ''), 'Unknown'),
71      Doctor = COALESCE(NULLIF(Doctor, ''), 'Unassigned'),
72      Name = COALESCE(NULLIF(Name, ''), 'Unassigned');
73 •  SET SQL_SAFE_UPDATES = 1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	Name	Age	Gender	Diagnosis	Medication	AdmissionDate	DischargeDate	Doctor	Department	Status
2001	John Doe	45	Male	Hypertension	Med A	2025-01-10	2025-01-15	Dr. Smith	Cardiology	admitted
2002	Jane Smith	45	Female	Diabetes	Med B	2025-01-10	15/01/2025	Dr. Lee	Endocrinology	admitted
2003	Bob Brown	55	Male	Asthma	Med C	2025-01-10	January 15, 2025	Dr. Carter	Pulmonology	under observation
2004	Unassigned	30	Female	Flu	Med D	2025-02-05	2025-02-10	Unassigned	General Medicine	discharged
2005	Tom Wilson	62	Male	Heart Disease	Med E	2025-03-01	2025-03-10	Dr. Johnson	Cardiology	discharged
2006	Susan Clark	49	Female	Kidney Disease	Med F	2025-04-12	2025-04-17	Dr. Patel	Nephrology	admitted
2007	David Jones	37	Male	Pneumonia	Med G	2025-05-20	2025-05-25	Dr. Martinez	Pulmonology	admitted
2008	Nancy Miller	28	Female	Flu	Med D	2025-06-15	2025-06-20	Dr. Smith	General Medicine	under observation
2009	Michael Scott	40	Male	Hypertension	Med A	2025-07-01	2025-07-07	Dr. Smith	Cardiology	admitted
2010	Pam Beesly	34	Female	Diabetes	Med B	2025-08-10	2025-08-15	Dr. Lee	Endocrinology	discharged

Python Tasks Use Pandas to clean the dataset further

Remove leading/trailing spaces

```
# remove leading/trailing spaces
df.columns = df.columns.str.strip()
df.head()
```

✓ 0.0s

	PatientID	Name	Age	Gender	Diagnosis	Medication	AdmissionDate	DischargeDate	Doctor	Department	Status
0	2001	John Doe	45	Male	Hypertension	Med A	1/10/2025	1/15/2025	Dr. Smith	Cardiology	admitted
1	2002	Jane Smith	45	Female	Diabetes	Med B	1/10/2025	15/01/2025	Dr. Lee	Endocrinology	admitted
2	2003	Bob Brown	55	Male	Asthma	Med C	1/10/2025	January 15 2025	Dr. Carter	Pulmonology	under observation
3	2004	Unassigned	30	Female	Flu	Med D	2/5/2025	2/10/2025	Unassigned	General Medicine	discharged
4	2005	Tom Wilson	62	Male	Heart Disease	Med E	3/1/2025	3/10/2025	Dr. Johnson	Cardiology	discharged

Handle missing values

```
# handle missing values
df.isnull().sum()
```

✓ 0.0s

```
PatientID        0
Name             0
Age              0
Gender           0
Diagnosis        0
Medication       0
AdmissionDate    0
DischargeDate    0
Doctor           0
Department       0
Status           0
dtype: int64
```

DischargeDate Format in Row 2

```
# DischargeDate in row 2 has a different format than the rest of the rows
# so we need to convert it to the same format
df['DischargeDate'] = df['DischargeDate'].str.replace('January 15 2025', '1/15/2025')
df.head(15)
```

✓ 0.0s

	PatientID	Name	Age	Gender	Diagnosis	Medication	AdmissionDate	DischargeDate	Doctor	Department	Status
0	2001	John Doe	45	Male	Hypertension	Med A	1/10/2025	1/15/2025	Dr. Smith	Cardiology	admitted
1	2002	Jane Smith	45	Female	Diabetes	Med B	1/10/2025	15/01/2025	Dr. Lee	Endocrinology	admitted
2	2003	Bob Brown	55	Male	Asthma	Med C	1/10/2025	1/15/2025	Dr. Carter	Pulmonology	under observation
3	2004	Unassigned	30	Female	Flu	Med D	2/5/2025	2/10/2025	Unassigned	General Medicine	discharged
4	2005	Tom Wilson	62	Male	Heart Disease	Med E	3/1/2025	3/10/2025	Dr. Johnson	Cardiology	discharged
5	2006	Susan Clark	49	Female	Kidney Disease	Med F	4/12/2025	4/17/2025	Dr. Patel	Nephrology	admitted
6	2007	David Jones	37	Male	Pneumonia	Med G	5/20/2025	5/25/2025	Dr. Martinez	Pulmonology	admitted
7	2008	Nancy Miller	28	Female	Flu	Med D	6/15/2025	6/20/2025	Dr. Smith	General Medicine	under observation
8	2009	Michael Scott	40	Male	Hypertension	Med A	7/1/2025	7/7/2025	Dr. Smith	Cardiology	admitted
9	2010	Pam Beesly	34	Female	Diabetes	Med B	8/10/2025	8/15/2025	Dr. Lee	Endocrinology	discharged

Part 2: Data Exploration and Analysis

1. **What is the average age of patients for each diagnosis?**

```
74    # 1-What is the average age of patients for each diagnosis?
75 •  SELECT Diagnosis, AVG(Age) AS Average_Age
76    FROM cleaned_data_py
77    GROUP BY Diagnosis;
78
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Diagnosis	Average_Age
Hypertension	42.5000
Diabetes	39.5000
Asthma	55.0000
Flu	29.0000
Heart Disease	62.0000
Kidney Disease	49.0000
Pneumonia	37.0000

2. **Which department has the highest number of admitted patients?**

```
79    # 2-Which department has the highest number of admitted patients?
80 •  SELECT
81        Department, COUNT(*) AS Total_Admissions
82    FROM
83        cleaned_data_py
84    WHERE
85        Status = 'admitted'
86    GROUP BY Department
87    ORDER BY Total_Admissions DESC
88    LIMIT 1;
89
90    # 3-How many patients have been discharged per month?
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch r

Department	Total_Admissions
Cardiology	2

3. **How many patients have been discharged per month?**

```
90    # 3-How many patients have been discharged per month?
91 •  SELECT
92        MONTH(DischargeDate) AS Month, COUNT(*) AS Total_Discharged
93    FROM
94        cleaned_data_py
95    WHERE
96        Status = 'discharged'
97    GROUP BY MONTH(DischargeDate)
98    ORDER BY Month;
99
100   # 4-What is the most common diagnosis among patients?
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month	Total_Discharged
NULL	3

4. **What is the most common diagnosis among patients?**

```sql
# 4-What is the most common diagnosis among patients?
SELECT
    Diagnosis, COUNT(*) AS Count
FROM
    cleaned_data_py
GROUP BY Diagnosis
ORDER BY Count DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Diagnosis	Count
Hypertension	2

5. **Which doctor has treated the most patients?**

```sql
# 5-Which doctor has treated the most patients?
SELECT
    Doctor, COUNT(*) AS Total_Patients_Treated
FROM
    cleaned_data_py
GROUP BY Doctor
ORDER BY Total_Patients_Treated DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

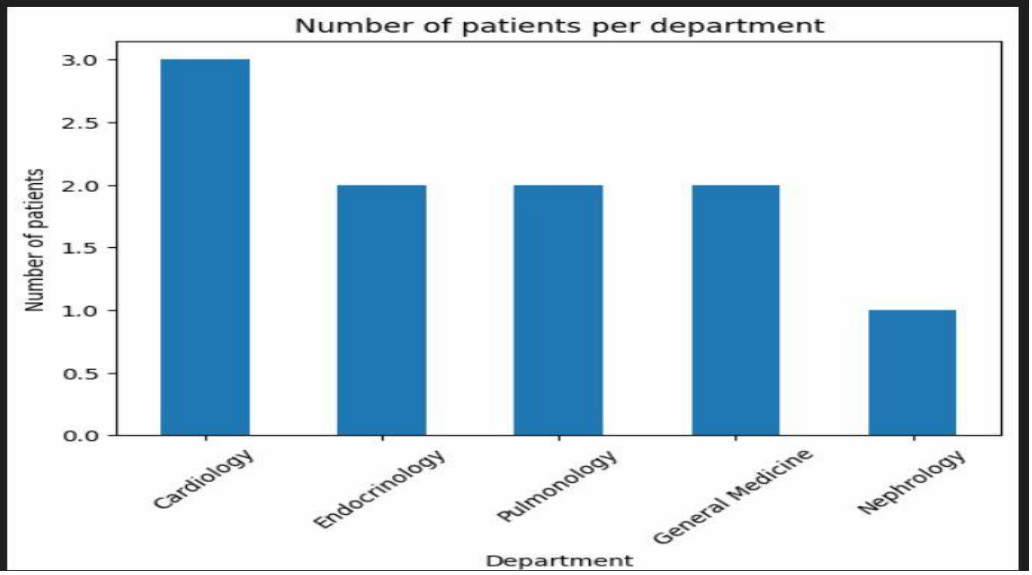
Diagnosis	Count
Hypertension	2

Python Analysis

1. **Visualize the number of patients per department using a bar chart.**

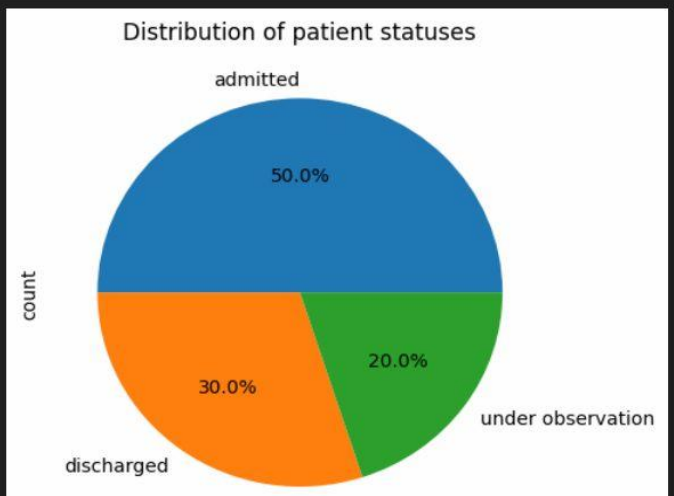
```python
#-Visualize-the-number-of-patients-per-department-using-a-bar-chart.
df['Department'].value_counts().plot(kind='bar')
plt.title('Number of patients per department')
plt.xlabel('Department')
plt.ylabel('Number of patients')
plt.xticks(rotation=45)
plt.show()
```

0.1s



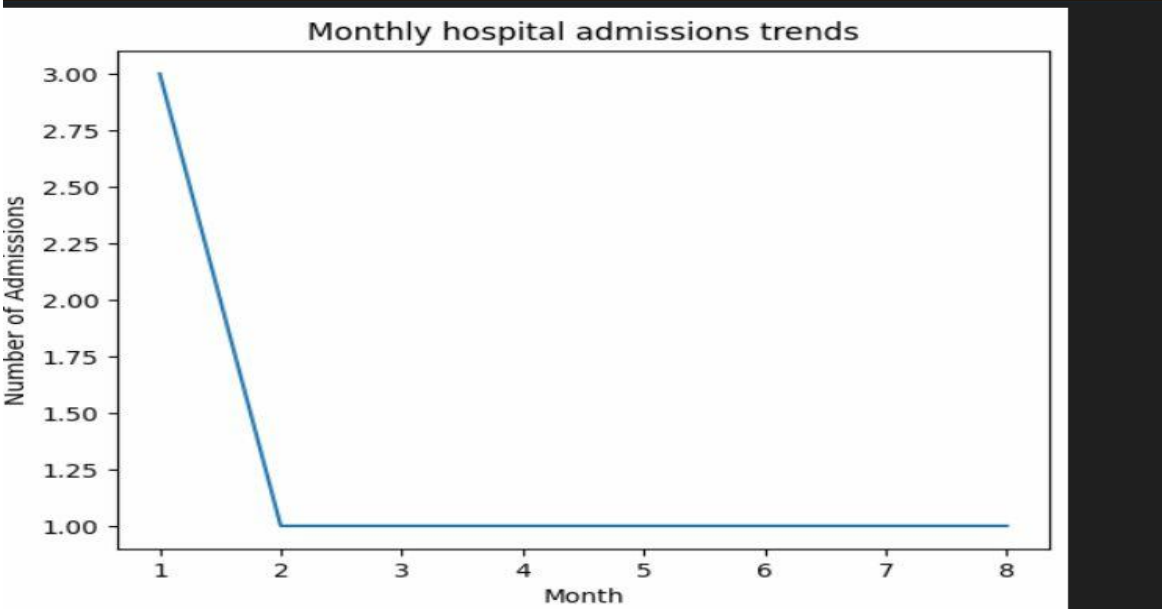
2. Create a pie chart showing the distribution of patient statuses (Admitted, Discharged, Under Observation).

```
# Create a pie chart showing the distribution of patient statuses (Admitted, Discharged, Under Observation).
df['Status'].value_counts().plot(kind='pie', autopct='%1.1f%%')
plt.title('Distribution of patient statuses')
plt.show()

✓ 0.1s
```



3. Generate a line chart showing monthly hospital admissions trends.

```
# Generate a line chart showing monthly hospital admissions trends.
df['AdmissionDate'] = pd.to_datetime(df['AdmissionDate'])
df['AdmissionDate'].dt.month.value_counts().sort_index().plot(kind='line')
plt.title('Monthly hospital admissions trends')
plt.xlabel('Month')
plt.ylabel('Number of Admissions')
plt.show()

0.1s
```



Part 3: Reporting

Recommendations for hospital management:

which departments need more resources:

- Cardiology
- Endocrinology
- Nephrology
- Pulmonology

Common patient demographics

Gender Distribution

		Gender
1	Male	5
2	Female	5

Age Statistics

		Age
3	std	10.88577052853862
4	min	28.0
5	25%	34.75
6	50%	42.5
7	75%	48.0
8	max	62.0

Most Common Departments

		Department
1	Cardiology	3
2	Endocrinology	2
3	Pulmonology	2
4	General Medicine	2
5	Nephrology	1

Most Common Diagnoses

		Diagnosis
2	Diabetes	2
3	Flu	2
4	Asthma	1
5	Heart Disease	1
6	Kidney Disease	1
7	Pneumonia	1

Theoretical Questions:

1. Data Cleaning

- **Common Issues in a Messy Dataset:**
  - **Duplicate Data:** Records that appear more than once.
  - **Missing Data:** Absences of values in one or more fields.
  - **Errors in Data Entry:** Typographical mistakes or incorrect values.
  - **Outliers:** Data points that deviate significantly from the norm.
- **Handling Missing Values:**
  - A common strategy is to replace missing values with the mean (or median/mode) of the column.
  - Other techniques include interpolation, using model-based imputations, or even deletion if appropriate.
- **Importance of Data Type Consistency:**
  - Ensures that calculations and comparisons can be performed accurately.
  - Simplifies the data cleaning process.
  - Reduces the risk of errors during analysis by ensuring that each column holds the expected type of data.

2. SQL Queries

- **Difference Between INNER JOIN and LEFT JOIN:**
  - **INNER JOIN:** Returns only the rows where there is a match on both tables.
  - **LEFT JOIN:** Returns all rows from the left (first) table and the matching rows from the right table; if there is no match, the result will include NULLs for columns from the right table.
- **Using the GROUP BY Clause to Aggregate Data:**
  - The GROUP BY clause is used to group rows that have the same values in one or more columns.
  - It is typically paired with aggregate functions (such as SUM (), AVG (), COUNT (), etc.) to perform calculations on each group.
- **Purpose of the HAVING Clause:**
  - After grouping data using GROUP BY, the HAVING clause is applied to filter groups based on a condition.
  - It serves a similar purpose to the WHERE clause but is used for aggregated data rather than individual rows.

3. Python Analysis

- **Cleaning a Dataset with Mixed Data Types Using Pandas:**
  - **df.convert_dtypes():** Automatically converts columns to the most suitable data types.
  - **df.infer_objects():** Attempts to infer better data types for columns that are currently stored as objects.
  - **df.apply(pd.to_numeric, errors='coerce'):** Converts columns to numeric values, coercing invalid parsing to NaN if necessary.
- **Benefits of Using Visualizations in Data Analysis:**
  1. **Enhanced Understanding:** Quickly identifies trends, patterns, and outliers.
  2. **Improved Decision-Making:** Empowers stakeholders to make data-driven decisions.
  3. **Simplified Communication:** Makes complex data more accessible to non-technical audiences.
  4. **Faster Insights:** Visual summaries can speed up the analysis process.
  5. **Effective Storytelling:** Helps to convey key messages and insights in a compelling manner.