AI Lung Diagnostics

Data Understanding

1. **To Check for the number of rows**

```sql
-- To check for the number of rows
SELECT COUNT(*) AS total_rows FROM lung_cancer_table;
```

2. **To Check for the number of column**

```sql
-- To check for the number of columns
SELECT COUNT(*) AS total_columns
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'lung_cancer_table' AND TABLE_SCHEMA = 'lung_cancer_db';
```

3. **To Check for the name of each column**

```sql
-- To see the name of each column
SELECT COLUMN_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'lung_cancer_table' AND TABLE_SCHEMA = 'lung_cancer_db';
```

4. **To see the first 3 rows in the column**

```sql
-- To see the first 3 rows (all columns)
SELECT *
FROM lung_cancer_table
LIMIT 3;
```

5. **To see the data types of each column**

```sql
--  To see the data types of each column
SELECT COLUMN_NAME, DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'lung_cancer_table' AND TABLE_SCHEMA = 'lung_cancer_db';
```

6. **To see the missing values**

```sql
-- To see the missing values
SELECT
  SUM(CASE WHEN id IS NULL THEN 1 ELSE 0 END) AS missing_id,
  SUM(CASE WHEN age IS NULL THEN 1 ELSE 0 END) AS missing_age,
  SUM(CASE WHEN gender IS NULL THEN 1 ELSE 0 END) AS missing_gender,
  SUM(CASE WHEN diagnosis_date IS NULL THEN 1 ELSE 0 END) AS missing_diagnosis_date,
  SUM(CASE WHEN cancer_stage IS NULL THEN 1 ELSE 0 END) AS missing_cancer_stage,
  SUM(CASE WHEN family_history IS NULL THEN 1 ELSE 0 END) AS missing_family_history,
  SUM(CASE WHEN smoking_status IS NULL THEN 1 ELSE 0 END) AS missing_smoking_status,
  SUM(CASE WHEN bmi IS NULL THEN 1 ELSE 0 END) AS missing_bmi,
  SUM(CASE WHEN cholesterol_level IS NULL THEN 1 ELSE 0 END) AS missing_cholesterol_level,
  SUM(CASE WHEN hypertension IS NULL THEN 1 ELSE 0 END) AS missing_hypertension,
  SUM(CASE WHEN asthma IS NULL THEN 1 ELSE 0 END) AS missing_asthma,
  SUM(CASE WHEN cirrhosis IS NULL THEN 1 ELSE 0 END) AS missing_cirrhosis,
  SUM(CASE WHEN other_cancer IS NULL THEN 1 ELSE 0 END) AS missing_other_cancer,
  SUM(CASE WHEN treatment_type IS NULL THEN 1 ELSE 0 END) AS missing_treatment_type,
  SUM(CASE WHEN end_treatment_date IS NULL THEN 1 ELSE 0 END) AS missing_end_treatment_date,
  SUM(CASE WHEN survived IS NULL THEN 1 ELSE 0 END) AS missing_survived
FROM lung_cancer_table;
```

7. **To check for duplicate rows**

```sql
-- To Check for the duplicate rows in the column
SELECT
  id, age, gender, diagnosis_date, cancer_stage, family_history, smoking_status,
  bmi, cholesterol_level, hypertension, asthma, cirrhosis, other_cancer,
  treatment_type, end_treatment_date, survived,
  COUNT(*) AS duplicate_count
FROM lung_cancer_table
GROUP BY
  id, age, gender, diagnosis_date, cancer_stage, family_history, smoking_status,
  bmi, cholesterol_level, hypertension, asthma, cirrhosis, other_cancer,
  treatment_type, end_treatment_date, survived
HAVING duplicate_count > 1;
```

Data Exploration

1. **For Continous Columns**
   - **Age**

```sql
-- Continous columns
-- Age
SELECT
  MIN(age) AS min_age,
  MAX(age) AS max_age,
  AVG(age) AS avg_age,
  STDDEV(age) AS stddev_age,
  COUNT(age) AS count_age
FROM lung_cancer_table;
```

   - **BMI**

```sql
-- BMI column
SELECT
  MIN(bmi) AS min_bmi,
  MAX(bmi) AS max_bmi,
  AVG(bmi) AS avg_bmi,
  STDDEV(bmi) AS stddev_bmi,
  COUNT(bmi) AS count_bmi
FROM lung_cancer_table;
```

   - **Cholestrol level**

```sql
-- Cholestrol Level column
SELECT
  MIN(cholesterol_level) AS min_cholesterol,
  MAX(cholesterol_level) AS max_cholesterol,
  AVG(cholesterol_level) AS avg_cholesterol,
  STDDEV(cholesterol_level) AS stddev_cholesterol,
  COUNT(cholesterol_level) AS count_cholesterol
FROM lung_cancer_table;
```

- **Outlier Detection**

```sql
-- Continous Column Spot simple outiers

SELECT *
FROM lung_cancer_table
WHERE
  age < 0 OR age > 120
  OR bmi < 10 OR bmi > 50
  OR cholesterol_level < 100 OR cholesterol_level > 300;
```

- **Outlier Range Check**

```sql
-- Continuous Columns – Outlier Range Check

SELECT
MIN(age) AS min_age, MAX(age) AS max_age,
MIN(bmi) AS min_bmi, MAX(bmi) AS max_bmi,
MIN(cholesterol_level) AS min_chol, MAX(cholesterol_level) AS max_chol
FROM lung_cancer_table;
```

- **Check for the first 5 rows in the tabel**

```sql
/*
This query shows the first 5 rows of the lung_cancer_table
to check column names and sample data values.
*/

SELECT *
FROM lung_cancer_table
LIMIT 5;
```

- **Range Check**
- **Age column**

```sql
-- Range Check for continous column
-- Range Check – age
SELECT
  MIN(age) AS min_age,
  MAX(age) AS max_age
FROM lung_cancer_table;
```

- **BMI Column**

```sql
--  Range Check – bmi
SELECT
  MIN(bmi) AS min_bmi,
  MAX(bmi) AS max_bmi
FROM lung_cancer_table;
```

- **Cholestrol Column**

```sql
-- Range Check – cholesterol_level
SELECT
  MIN(cholesterol_level) AS min_cholesterol,
  MAX(cholesterol_level) AS max_cholesterol
FROM lung_cancer_table;
```

- **Check distribution shape (binned counts)**
- **Age**

```sql
-- Check distribution shape (binned counts)
-- bin age into decades
SELECT
  FLOOR(age/10)*10 AS age_group,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY age_group
ORDER BY age_group;
```

- **BMI**

```sql
-- Same idea for bmi (custom bins)
SELECT
  FLOOR(bmi/5)*5 AS bmi_group,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY bmi_group
ORDER BY bmi_group;
```

- **Cholestrol level**

```sql
-- Same idea for cholesterol_level
SELECT
  FLOOR(cholesterol_level/50)*50 AS cholesterol_group,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY cholesterol_group
ORDER BY cholesterol_group;
```

2. **For Categorical Column**
   - **Unique value**

```sql
-- Data Exploration for Categorical column
-- See unique values in each categorical column

SELECT DISTINCT gender FROM lung_cancer_table;
SELECT DISTINCT cancer_stage FROM lung_cancer_table;
SELECT DISTINCT smoking_status FROM lung_cancer_table;
SELECT DISTINCT family_history FROM lung_cancer_table;
SELECT DISTINCT hypertension FROM lung_cancer_table;
SELECT DISTINCT asthma FROM lung_cancer_table;
SELECT DISTINCT other_cancer FROM lung_cancer_table;
SELECT DISTINCT treatment_type FROM lung_cancer_table;
SELECT DISTINCT survived FROM lung_cancer_table;
```

   - **Get counts**

```sql
-- Get counts/frequency for each category

SELECT gender, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY gender
ORDER BY count DESC;
```

```sql
SELECT cancer_stage, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY cancer_stage
ORDER BY count DESC;
```

```sql
SELECT smoking_status, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY smoking_status
ORDER BY count DESC;
```

```sql
SELECT family_history, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY family_history
ORDER BY count DESC;
```

```sql
SELECT hypertension, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY hypertension
ORDER BY count DESC;
```

```sql
SELECT asthma, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY asthma
ORDER BY count DESC;
```

```sql
SELECT other_cancer, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY other_cancer
ORDER BY count DESC;
```

```sql
SELECT treatment_type, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY treatment_type
ORDER BY count DESC;
```

```sql
SELECT survived, COUNT(*) AS count
FROM lung_cancer_table
GROUP BY survived
ORDER BY count DESC;
```

- **Explore relationship between categorical column**
- **Survival by gender**

```sql
-- Survival by Gender
SELECT
  gender,
  survived,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY gender, survived
ORDER BY gender, survived;
```

- **Survival by Cancer stage**

```sql
-- Survival by Cancer Stage
SELECT
  cancer_stage,
  survived,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY cancer_stage, survived
ORDER BY cancer_stage, survived;
```

- **Survival by Smoking status**

```sql
-- Survival by Smoking Status
SELECT
  smoking_status,
  survived,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY smoking_status, survived
ORDER BY smoking_status, survived;
```

- **Survival by Family History**

```sql
-- Survival by Family History
SELECT
  family_history,
  survived,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY family_history, survived
ORDER BY family_history, survived;
```

- **Survival by Treatement type**

```sql
-- Survival by Treatment Type
SELECT
  treatment_type,
  survived,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY treatment_type, survived
ORDER BY treatment_type, survived;
```

- **Survival by Hypertension**

```sql
-- Survival by Hypertension
SELECT
  hypertension,
  survived,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY hypertension, survived
ORDER BY hypertension, survived;
```

- **Survival count by asthma**

```sql
-- Survival by Asthma
SELECT
  asthma,
  survived,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY asthma, survived
ORDER BY asthma, survived;
```

- **Survival count by other_cancer**

```sql
-- Survival by Other Cancer
SELECT
  other_cancer,
  survived,
  COUNT(*) AS count
FROM lung_cancer_table
GROUP BY other_cancer, survived
ORDER BY other_cancer, survived;
```

Data Cleaning

```sql
-- Data Cleaning
-- No missing values in the Data set
-- Fix the data types
ALTER TABLE lung_cancer_table MODIFY COLUMN hypertension VARCHAR(5);
ALTER TABLE lung_cancer_table MODIFY COLUMN asthma VARCHAR(5);
ALTER TABLE lung_cancer_table MODIFY COLUMN other_cancer VARCHAR(5);
ALTER TABLE lung_cancer_table MODIFY COLUMN survived VARCHAR(5);

-- Disabling the safeupdates
SET SQL_SAFE_UPDATES = 0;

-- Hypertension
UPDATE lung_cancer_table SET hypertension = 'Yes' WHERE hypertension = '1';
UPDATE lung_cancer_table SET hypertension = 'No' WHERE hypertension = '0';


-- Asthma
SELECT COUNT(*) FROM lung_cancer_table WHERE asthma = '0';

-- Disabling the safeupdates
SET SQL_SAFE_UPDATES = 0;

UPDATE lung_cancer_table
SET asthma = 'No'
WHERE asthma = '0'
LIMIT 1000;
```

```sql
-- Check how many are left to be converted to No
SELECT COUNT(*) FROM lung_cancer_table WHERE asthma = '0';
UPDATE lung_cancer_table SET asthma = 'Yes' WHERE asthma = '1';
UPDATE lung_cancer_table SET asthma = 'No' WHERE asthma = '0';

-- Other Cancer
SELECT COUNT(*) FROM lung_cancer_table WHERE other_cancer = '1';
SELECT COUNT(*) FROM lung_cancer_table WHERE other_cancer = '0';

UPDATE lung_cancer_table
SET other_cancer = 'No'
WHERE other_cancer = '0'
LIMIT 10000;

SELECT COUNT(*) FROM lung_cancer_table WHERE other_cancer = '0';

UPDATE lung_cancer_table SET other_cancer = 'Yes' WHERE other_cancer = '1';
UPDATE lung_cancer_table SET other_cancer = 'No' WHERE other_cancer = '0';

-- Survived
UPDATE lung_cancer_table SET survived = 'Yes' WHERE survived = '1';
UPDATE lung_cancer_table SET survived = 'No' WHERE survived = '0';
```

Author – Niranjan
Date – 30-June-2025