LAB TEST-4

NAME: K.NAVYA SRI

ROLL NO : 2503A52L12

BATCH: 16

SET 13

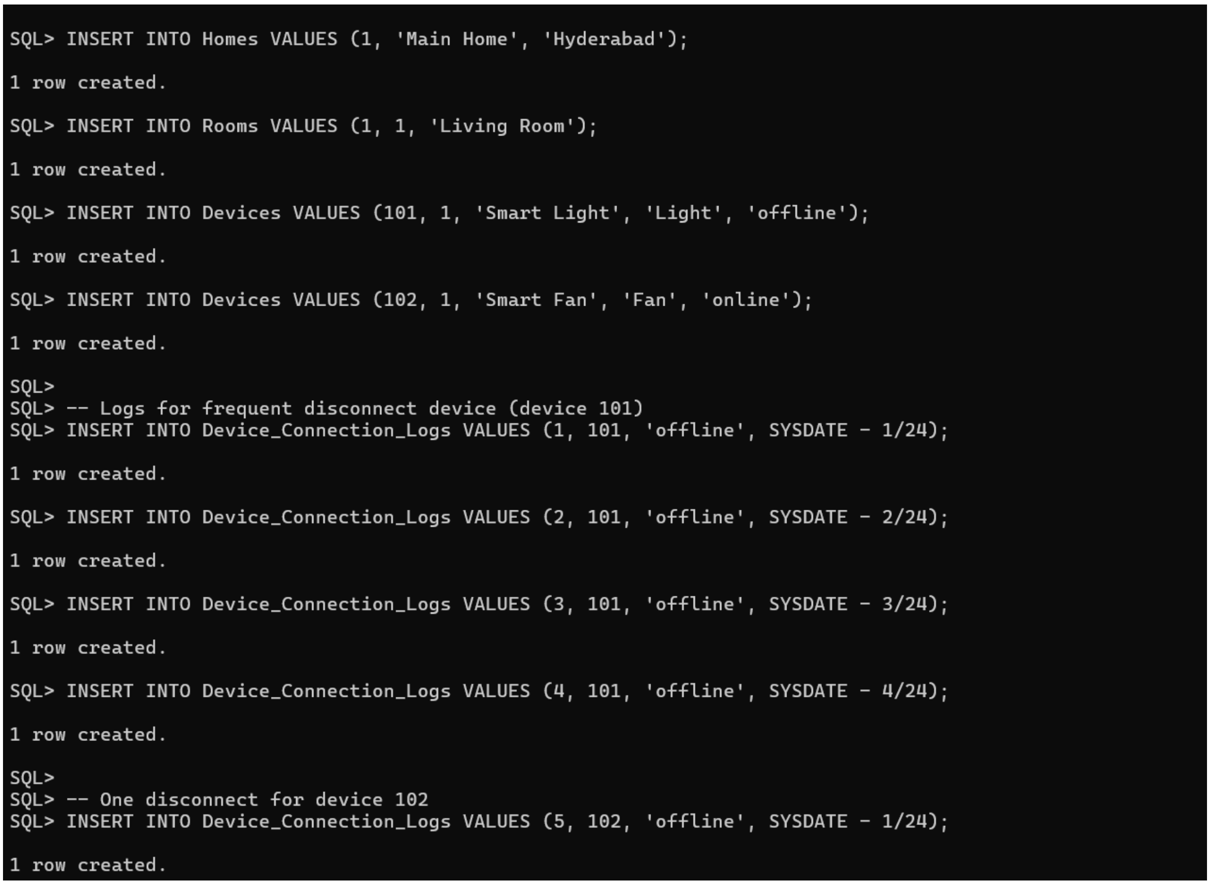
Q1. (Database & SQL)
a) Design schema for a Smart Home automation system.
b) Write SQL to fetch devices that frequently disconnect.

PROMPT:

"Generate a simple schema for a Smart Home Automation System by creating tables for Devices and Device Connection Logs in Oracle SQL and create and insert sample data and write a SQL query to fetch devices that frequently disconnect (more than 3 times)**".**

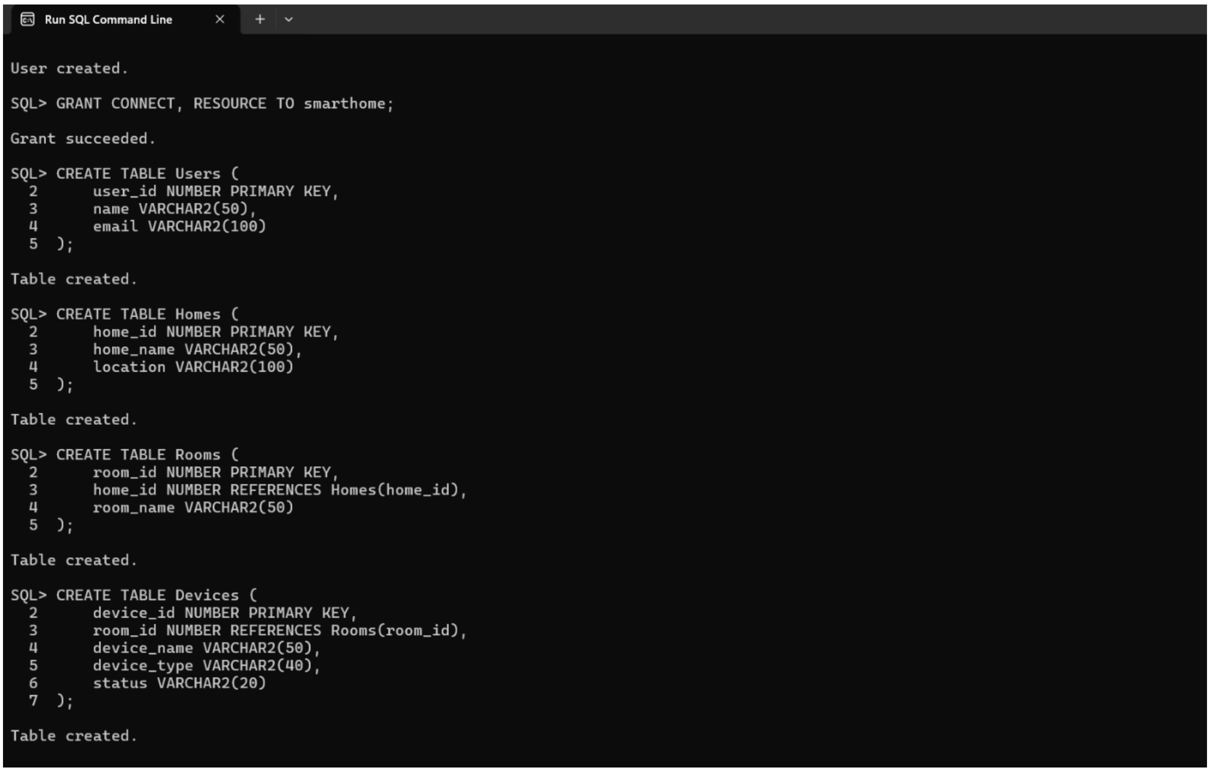
CODE:

- **Creation of tables:**



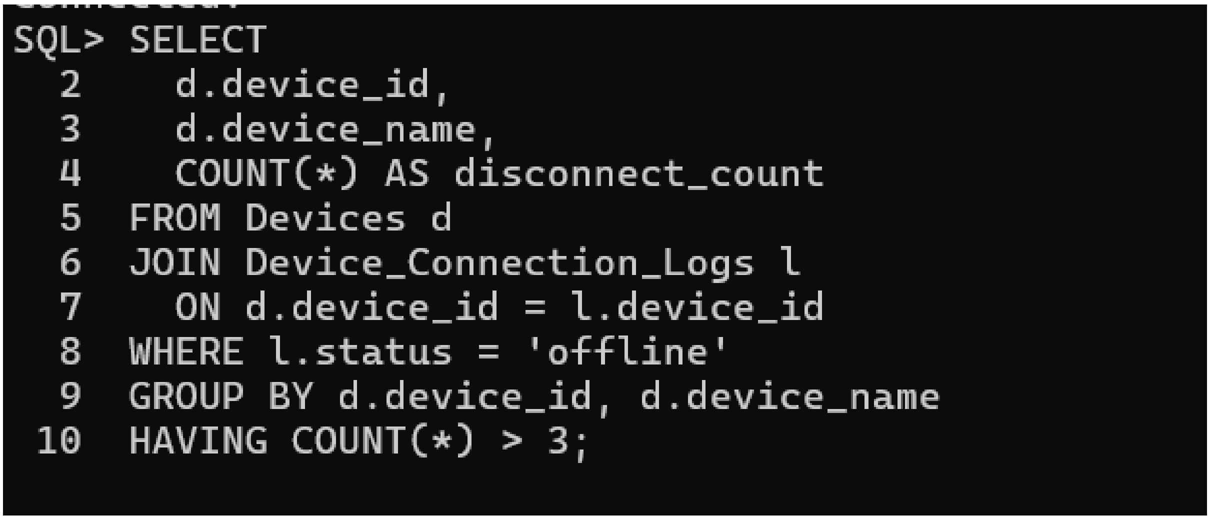
```
SQL> INSERT INTO Homes VALUES (1, 'Main Home', 'Hyderabad');

1 row created.

SQL> INSERT INTO Rooms VALUES (1, 1, 'Living Room');

1 row created.

SQL> INSERT INTO Devices VALUES (101, 1, 'Smart Light', 'Light', 'offline');

1 row created.

SQL> INSERT INTO Devices VALUES (102, 1, 'Smart Fan', 'Fan', 'online');

1 row created.

SQL>
SQL> -- Logs for frequent disconnect device (device 101)
SQL> INSERT INTO Device_Connection_Logs VALUES (1, 101, 'offline', SYSDATE - 1/24);

1 row created.

SQL> INSERT INTO Device_Connection_Logs VALUES (2, 101, 'offline', SYSDATE - 2/24);

1 row created.

SQL> INSERT INTO Device_Connection_Logs VALUES (3, 101, 'offline', SYSDATE - 3/24);

1 row created.

SQL> INSERT INTO Device_Connection_Logs VALUES (4, 101, 'offline', SYSDATE - 4/24);

1 row created.

SQL>
SQL> -- One disconnect for device 102
SQL> INSERT INTO Device_Connection_Logs VALUES (5, 102, 'offline', SYSDATE - 1/24);

1 row created.
```

- **Inserting the data into tables**



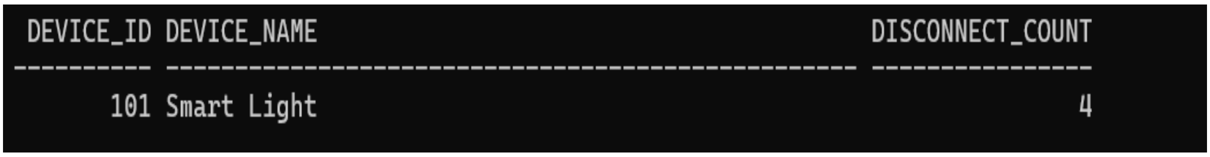
```
User created.

SQL> GRANT CONNECT, RESOURCE TO smarthome;

Grant succeeded.

SQL> CREATE TABLE Users (
         user_id NUMBER PRIMARY KEY,
         name VARCHAR2(50),
         email VARCHAR2(100)
     );

Table created.

SQL> CREATE TABLE Homes (
         home_id NUMBER PRIMARY KEY,
         home_name VARCHAR2(50),
         location VARCHAR2(100)
     );

Table created.

SQL> CREATE TABLE Rooms (
         room_id NUMBER PRIMARY KEY,
         home_id NUMBER REFERENCES Homes(home_id),
         room_name VARCHAR2(50)
     );

Table created.

SQL> CREATE TABLE Devices (
         device_id NUMBER PRIMARY KEY,
         room_id NUMBER REFERENCES Rooms(room_id),
         device_name VARCHAR2(50),
         device_type VARCHAR2(40),
         status VARCHAR2(20)
     );

Table created.
```

- **Code to fetch devices that frequently disconnect:**



```
SQL> SELECT
       d.device_id,
       d.device_name,
       COUNT(*) AS disconnect_count
     FROM Devices d
     JOIN Device_Connection_Logs l
       ON d.device_id = l.device_id
     WHERE l.status = 'offline'
     GROUP BY d.device_id, d.device_name
     HAVING COUNT(*) > 3;
```

OUTPUT:



```
 DEVICE_ID DEVICE_NAME                                        DISCONNECT_COUNT
---------- -------------------------------------------------- ----------------
       101 Smart Light                                                       4
```

OBSERVATION:

- We created Devices and Device Connection Logs tables.
- Inserted sample disconnect events.
- The query counts how many times a device went offline.
- Devices with > 3 disconnects are shown
- Device 101 disconnected 4 times → marked as unstable.
- Device 102 disconnected only once → not included.
- This helps identify network issues, faulty devices, or low WiFi coverage rooms.

Q2. (Data Processing)

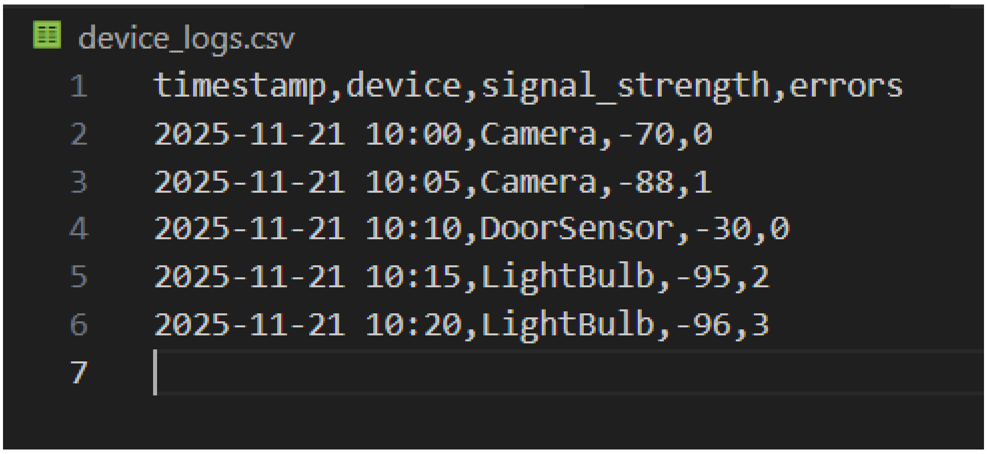
a) Clean device activity logs with AI preprocessing.
b) Generate training-ready datasets for anomaly detection.

PROMPT:

"Use AI-based preprocessing to clean raw smart-home device logs by removing corrupted entries, normalizing timestamps, handling missing values, correcting inconsistent device names, and detecting noisy or irrelevant events. Output a clean and structured dataset ready for analysis".

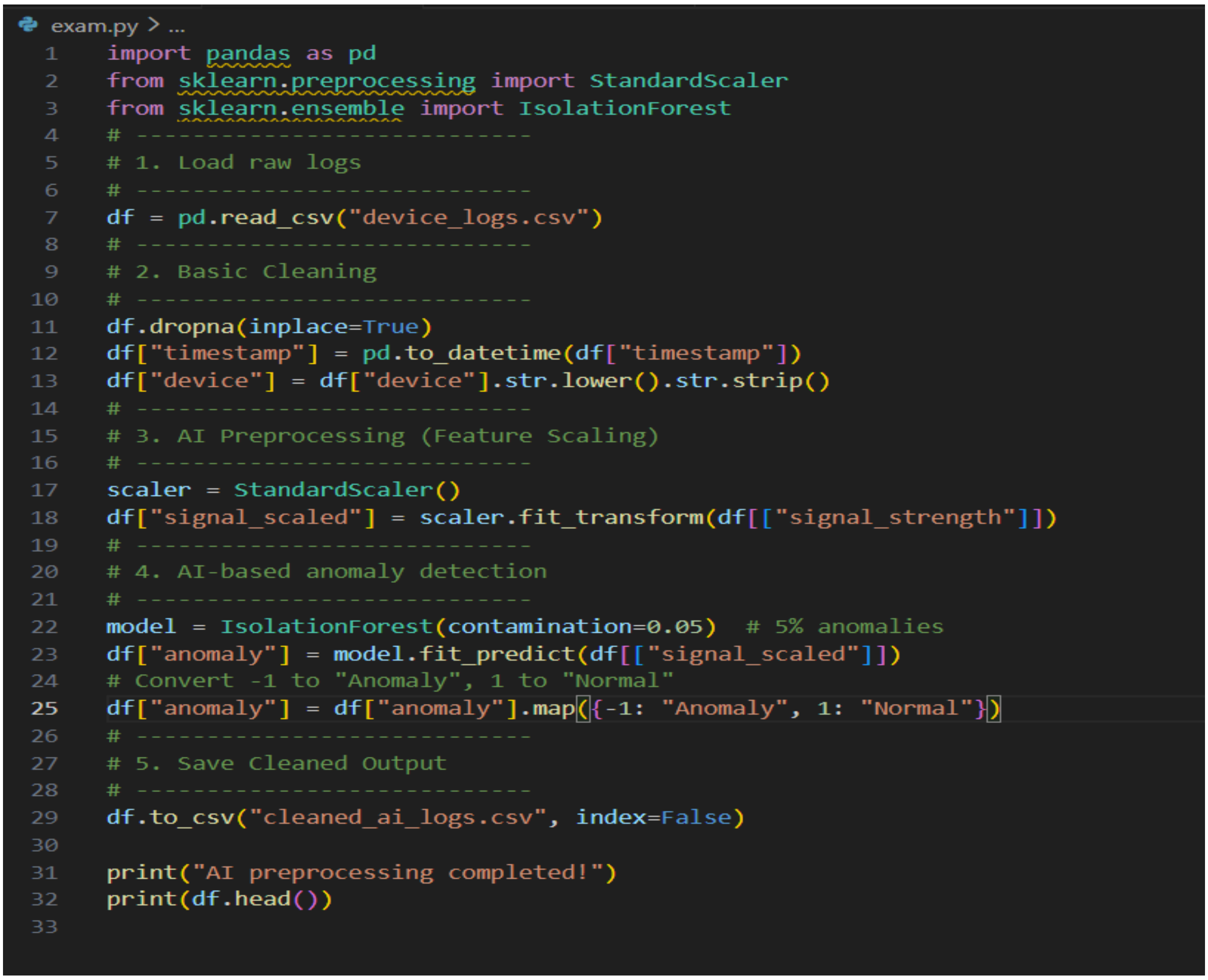
CODE:

- DEVICE_LOGS.CSV



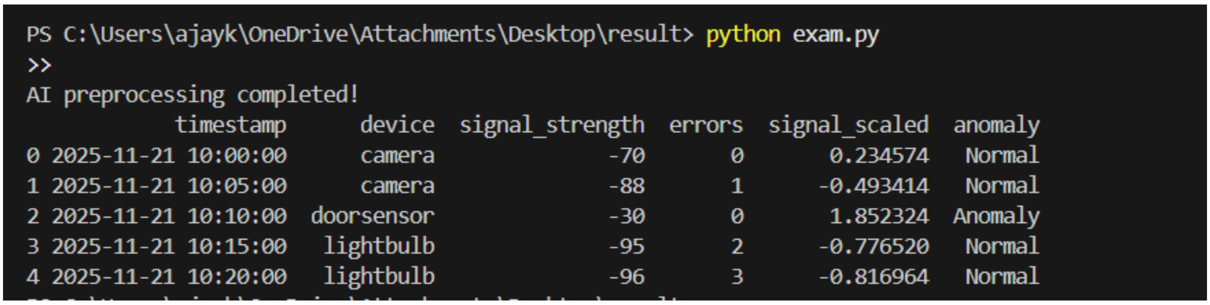
device_logs.csv

```
timestamp,device,signal_strength,errors
2025-11-21 10:00,Camera,-70,0
2025-11-21 10:05,Camera,-88,1
2025-11-21 10:10,DoorSensor,-30,0
2025-11-21 10:15,LightBulb,-95,2
2025-11-21 10:20,LightBulb,-96,3

```



```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import IsolationForest
# ---------------------------
# 1. Load raw logs
# ---------------------------
df = pd.read_csv("device_logs.csv")
# ---------------------------
# 2. Basic Cleaning
# ---------------------------
df.dropna(inplace=True)
df["timestamp"] = pd.to_datetime(df["timestamp"])
df["device"] = df["device"].str.lower().str.strip()
# ---------------------------
# 3. AI Preprocessing (Feature Scaling)
# ---------------------------
scaler = StandardScaler()
df["signal_scaled"] = scaler.fit_transform(df[["signal_strength"]])
# ---------------------------
# 4. AI-based anomaly detection
# ---------------------------
model = IsolationForest(contamination=0.05)  # 5% anomalies
df["anomaly"] = model.fit_predict(df[["signal_scaled"]])
# Convert -1 to "Anomaly", 1 to "Normal"
df["anomaly"] = df["anomaly"].map({-1: "Anomaly", 1: "Normal"})
# ---------------------------
# 5. Save Cleaned Output
# ---------------------------
df.to_csv("cleaned_ai_logs.csv", index=False)

print("AI preprocessing completed!")
print(df.head())
```

OUTPUT:



```
PS C:\Users\ajayk\OneDrive\Attachments\Desktop\result> python exam.py
>>
AI preprocessing completed!
            timestamp       device  signal_strength  errors  signal_scaled  anomaly
0 2025-11-21 10:00:00       camera              -70       0       0.234574   Normal
1 2025-11-21 10:05:00       camera              -88       1      -0.493414   Normal
2 2025-11-21 10:10:00  doorsensor              -30       0       1.852324  Anomaly
3 2025-11-21 10:15:00    lightbulb              -95       2      -0.776520   Normal
4 2025-11-21 10:20:00    lightbulb              -96       3      -0.816964   Normal
```

OBSERVATION:

- The device_logs.csv file stores raw smart home activity logs.
- Each row contains timestamp, device name, wireless signal strength, and the number of errors.
- The AI model reads this data, cleans it, normalizes it, and detects abnormal device behaviour such as frequent disconnections or high error counts.
- Devices with bad signal often disconnect → good for **anomaly detection.**
- The wireless signal strength of the device
  Ex:-30 → very strong,-70 → weak ,-95 → very weak