

Experiment 10 Output

Real-time data warehousing with **streaming integration**. Continuously ingest and store live data for dynamic analysis. Implement **efficient data handling** and **visualization**.

Creating Database and Data Table

```
# Create Database
cursor.execute("CREATE DATABASE IF NOT EXISTS streaming_warehouse")
cursor.execute("USE streaming_warehouse")

# Create Table for Real-time Data
cursor.execute("""
    CREATE TABLE IF NOT EXISTS real_time_data (
        id INT AUTO_INCREMENT PRIMARY KEY,
        timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        sensor_value FLOAT,
        device_name VARCHAR(50),
        status VARCHAR(20)
    )
""")
```

Fig 1: Database and Table Creation

Inserting Data Via SQL and Python

```
# Insert Query
insert_query = """
    INSERT INTO real_time_data (timestamp, sensor_value, device_name, status)
    VALUES (%s, %s, %s, %s)
"""

devices = ["Sensor_A", "Sensor_B", "Sensor_C", "Sensor_D"]
statuses = ["Normal", "Warning", "Critical"]
status_colors = {"Normal": "green", "Warning": "orange", "Critical": "red"}

plot_count = 0 # Counter for plots
insert_interval = (2, 6) # Mention interval in seconds (min, max)

# Function to fetch and plot data
def fetch_and_plot():
    cursor.execute("SELECT timestamp, sensor_value, status FROM real_time_data")
    data = cursor.fetchall()

    if not data:
        print("⚠ No data available!")
    return
```

Fig 2: Data Insertion

Plots Showing Streaming Data Status

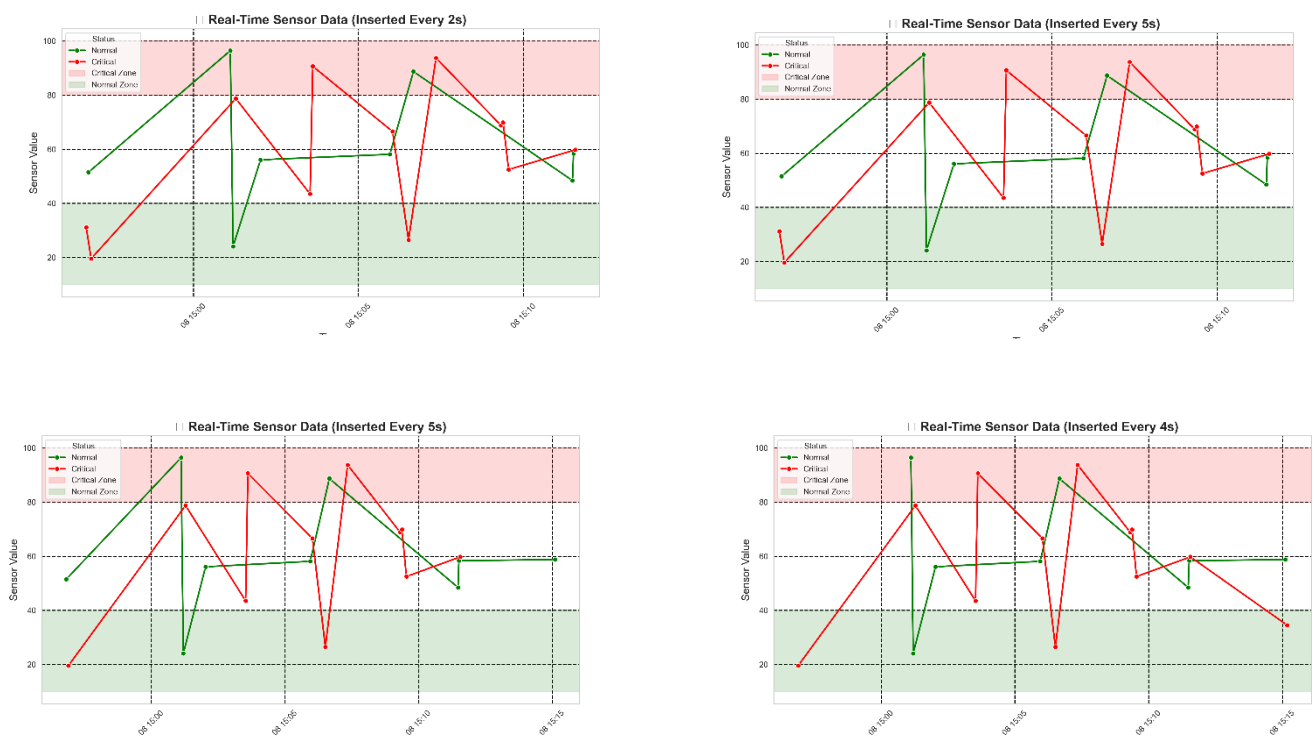


Fig 3: Sensor Distribution Graph