How to use dummy data?

1. Open pgAdmin.
2. Create or select the target database where you want to insert the dummy data.
3. Right-click the database and choose Restore.
4. In the Restore window:
   - Change Format to Plain.
   - Click the file icon and select your `dummy.sql` file.
5. Click the Restore button to execute the script and insert the dummy data.

FFmpeg Installation

1. Unzip 'ffmpeg' folder and place in C: Drive
2. Add FFmpeg to the system path
   - Open "Environment Variables.."
   - Under System variables, select "Path" variable, then click "Edit.."
   - Click "New"
   - Add the full path to the 'bin' directory of your FFmpeg (C:\ffmpeg\bin)
3. Open a new Command Prompt
4. Type `'ffmpeg -version'` and press Enter to ensure the ffmpeg is installed correctly

PIP Installation

1. Install the requirements.txt file
   - In the terminal, enter `'pip install -r READ_ME/requirements.txt'`

Use case Execution

1. Use case 13
   - Before running use case 13, run the following command from the project's root folder:
     - `'python notify.py'`
   - This will trigger `transcribe.py` and `patient_report.py`

Modify .env file

DB_HOST=localhost
DB_PORT=5432
DB_NAME=postgres
DB_USER=postgres
DB_PASSWORD=your_db_password
DB_SCHEMA=SIGMAmed

KINETICA_HOST=https://<your_cluster>/gpudb-0
KINETICA_PORT=443
KINETICA_USERNAME=your_user
KINETICA_PASSWORD=your_password
KINETICA_SCHEMA=
# leave blank to use your default namespace, or set if you have schema rights
KINETICA_USE_TLS=true

Run Sliding Window

1. Run the following command from the project's root folder:
   - `py medication_compliance.py`

Run Kinetica

1. Run the following command from the project's root folder:
   - `py kinetica_analysis_medication.py --limit 500   # omit --limit for full load`

In the Kinetica Dashboard (one workbook per query)

Analysis 1
Patients Reporting the Most Side Effects:

```
SELECT patient_name, patient_username, report_count,
unique_side_effects, medications
FROM side_effects_by_patient
ORDER BY report_count DESC
LIMIT 20;
```

X-axis: patient_username
Y-axis: report_count

Analysis 2
Top Side Effects by Medication:

```
SELECT side_effect_name, medication_name, severity, report_count
FROM side_effects_by_medication
ORDER BY report_count DESC
LIMIT 20;
```

X-axis: medication_name
Y-axis: report_count

Analysis 3
Top Side Effects (overall):

```
SELECT side_effect_name, total_reports, top_medication,
top_medication_report_count
FROM top_side_effects
ORDER BY total_reports DESC
LIMIT 20;
```

X-axis: side_effect_name
Y-axis: total_reports

Analysis 4
Medication Compliance Summary

```
SELECT
    patient_id,
    medication_name,
    alert_type,
    severity,
    violation_count,
    window_start,
    window_end,
    last_violation_date
FROM patient_compliance_alerts
ORDER BY last_violation_date DESC
LIMIT 100
```

X-axis: medication_name
Y-axis: violation_count

Analysis 5
Violation Trend (Week)

```
SELECT
    DATE_TRUNC('week', last_violation_date) as week,
    alert_type,
    severity,
    COUNT(*) as alert_count
FROM patient_compliance_alerts
GROUP BY week, alert_type, severity
ORDER BY week DESC
```

X-axis: week
Y-axis: alert_count
Series column: alert_type

Analysis 6
High-Risk Patient Analysis

```
SELECT
    patient_id,
    COUNT(*) as total_alerts,
    MAX(violation_count) as max_violations
FROM patient_compliance_alerts
GROUP BY patient_id
ORDER BY total_alerts DESC
LIMIT 50
```

X-axis: patient_id
Y-axis: total_alerts