---

# 🌍 Air Pollution & Health Analytics Using SQLite and Python

**Google Colab Link:** Open Project Notebook

---

## 🔎 Project Overview

This project explores the environmental and public health impact of air pollution across urban regions by integrating structured data processing with insightful visualizations. Using SQLite and Python in a Google Colab environment, the project transforms raw CSV data into a normalized relational database, performs in-depth analysis through SQL, and visualizes the most polluted cities and pollution trends over time.

---

## 🎯 Objectives

- Build a clean and scalable **SQL database** from air quality datasets.
- Conduct **data-driven analysis** on air pollution trends and city-wise pollution levels.
- Visualize **top polluted cities** and year-wise patterns using `Matplotlib`.
- Demonstrate data pipeline and analytics skills in a **real-world environmental context**.

---

## 🛠️ Tech Stack

- **Languages & Tools**: Python, SQL (SQLite), Google Colab
- **Libraries Used**: Pandas, Matplotlib, PrettyTable, ipython-sql
- **Database**: SQLite (via `ipython-sql` in Colab)

---

## 🧱 Database Schema Design

- **`cities`**: Stores unique city names with auto-incremented IDs.
- **`air_quality`**: Contains daily pollution data (CO, $NO_2$, PM10) along with temperature and humidity, linked to `cities` by `city_id`.
- **`health_indicators`** *(optional/extendable)*: Includes year-wise health metrics such as mortality rate and life expectancy.

```
CREATE TABLE IF NOT EXISTS cities (
    city_id INTEGER PRIMARY KEY AUTOINCREMENT,
    city_name TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS air_quality (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    city_id INTEGER,
    date TEXT,
    co REAL,
    no2 REAL,
    pm10 REAL,
    temperature REAL,
    humidity REAL,
    FOREIGN KEY(city_id) REFERENCES cities(city_id)
);

CREATE TABLE IF NOT EXISTS health_indicators (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    city_id INTEGER,
    year INTEGER,
    mortality_rate REAL,
    life_expectancy REAL,
    FOREIGN KEY(city_id) REFERENCES cities(city_id)
);
```

---

## 🔄 ETL Workflow (Extract, Transform, Load)

1. **Data Upload & Initial Inspection**

- Imported a CSV file (`AirQuality 2.csv`) with pollutant and weather data.
- Explored structure and identified column inconsistencies and encoding issues.

2. **Data Cleaning & Transformation**
   - Standardized column names: `CO(GT)` → `co`, `PM10 (GT)` → `pm10`, etc.
   - Added a consistent `city_name` column for relational mapping.
   - Handled delimiter issues (`;`), null values, and unnamed columns.
3. **Database Population**
   - Created SQLite database (`air_quality_health.db`) in Colab.
   - Inserted unique cities and linked pollution data using `city_id`.
   - Ensured schema integrity with foreign keys and primary key constraints.

---

# 📈 Data Analysis Using SQL

## 🔝 Top 10 Most Polluted Cities by PM10

```
SELECT c.city_name, AVG(a.pm10) AS avg_pm10
FROM air_quality a
JOIN cities c ON a.city_id = c.city_id
GROUP BY c.city_name
ORDER BY avg_pm10 DESC
LIMIT 10;
```

- **Insight**: Highlights the cities with the highest concentration of PM10 particles on average.

## 📆 Yearly Trend in PM10 for Delhi

```
SELECT SUBSTR(date, 1, 4) AS year, AVG(pm10) AS avg_pm10
FROM air_quality a
JOIN cities c ON a.city_id = c.city_id
WHERE c.city_name = 'Delhi'
GROUP BY year
```

```
ORDER BY year;
```

- **Insight**: Observes PM10 fluctuations over time for a major metropolitan area.

### 🌡️ Overall Environmental Summary

```
SELECT ROUND(AVG(temperature), 2) AS avg_temp,
       ROUND(AVG(humidity), 2) AS avg_humidity,
       ROUND(AVG(pm10), 2) AS avg_pm10
FROM air_quality;
```

- **Insight**: Provides a snapshot of environmental averages across all records.

---

## 📊 Data Visualization

- **Top Polluted Cities (Bar Chart):**
    - Created using `Matplotlib`, this chart visually represents cities ranked by average PM10 concentration.
    - Enhanced with axis labels, rotated city names, and responsive layout.

```
plt.bar(top_cities_df['city_name'], top_cities_df['avg_pm10'], color='tomato')
plt.title('Top 10 Most Polluted Cities by PM10')
```

---

## 📌 Key Skills Demonstrated

| Area | Skills |
|---|---|
| **Data Handling** | Data cleaning, transformation, Pandas |
| **SQL Mastery** | Joins, aggregation, filtering, subqueries |
| **Database Design** | Schema normalization, primary/foreign keys |
| **Visualization** | Matplotlib plots, trend analysis |
| **Integration** | SQL and Python in a unified Colab pipeline |
| **Real-World Relevance** | Pollution monitoring and public health |

---

## 📁 Dataset Details

- **File Used**: `AirQuality 2.csv` (user-uploaded)
- **Size**: < 3MB (optimized for sharing with recruiters)
- **Attributes**: Date, CO, $NO_2$, PM10, Temperature, Humidity, City

---

## 🚀 Recruiter Value Proposition

This project reflects my ability to:

- Design and manage relational databases from scratch
- Tackle real-world data challenges with SQL and Python
- Deliver actionable insights with visual storytelling
- Work in collaborative, cloud-native environments like Google Colab

---