Pitch deck link:

https://gamma.app/docs/Improving-Urban-Traffic-Management-and-Air-Quality-aa6o0wkgqaqbud6

Problem Definition

Problem: Inefficient Traffic Management Leading to Increased Air Pollution in Urban Areas

Description:

Many urban areas experience severe traffic congestion due to inefficient traffic management systems. This congestion not only leads to increased travel times and reduced quality of life for residents but also contributes significantly to air pollution. Traffic jams cause vehicles to idle for longer periods, resulting in higher emissions of pollutants such as nitrogen oxides (NOx) and particulate matter (PM). These pollutants contribute to respiratory problems and other health issues for city residents.

Data-Driven Approach to Address the Problem:

To address this problem, we can use a combination of data sources and advanced analytics to improve traffic management and reduce air pollution. Here's a detailed approach:

1. **Data Collection:**
    - Traffic Flow Data: Collect real-time traffic data from sensors, cameras, and GPS devices installed on vehicles.
    - Air Quality Data: Use air quality monitoring stations to gather data on pollutant levels across different parts of the city.
    - Weather Data: Include weather conditions such as temperature, humidity, and wind speed, as these can affect both traffic flow and pollution dispersion.
    - Historical Traffic Data: Analyze past traffic patterns and congestion hotspots.
2. **Data Analysis:**
    - Traffic Pattern Analysis: Use data analytics to identify patterns and peak congestion times. This can help in understanding which areas are most affected and when.
    - Pollution Correlation: Analyze the relationship between traffic congestion and air pollution levels to identify how congestion contributes to increased pollution.
    - Predictive Modeling: Develop predictive models to forecast traffic congestion and pollution levels based on real-time data and historical trends.
3. **Solution Implementation:**
    - Optimized Traffic Signals: Use data to optimize traffic signal timings and improve traffic flow. Adaptive traffic signal systems can adjust in real-time based on current traffic conditions.
    - Traffic Management Strategies: Implement congestion pricing or alternative routing to reduce traffic in high-density areas.

- Public Transportation Integration: Enhance public transportation options based on traffic and pollution data to encourage fewer personal vehicle trips.
- Real-Time Alerts and Apps: Develop apps that provide real-time traffic and pollution updates to help drivers make informed decisions and reduce idling.

4. **Monitoring and Evaluation:**

    - Continuous Monitoring: Keep track of traffic flow and air quality data to assess the impact of implemented solutions.
    - Feedback Loop: Use data to continuously refine and improve traffic management strategies and pollution reduction efforts.

By leveraging data in this way, urban areas can make more informed decisions to manage traffic efficiently, which can lead to reduced air pollution and improved quality of life for residents.

ERD (Entity-Relationship Diagram)

Simplified ERD for the traffic management and air pollution project:

Entities

1. TrafficSensor
   - `SensorID` (Primary Key)
   - `Location` (Geographical coordinates or address)
   - `Type` (e.g., Camera, GPS, etc.)

2. TrafficData
   - `DataID` (Primary Key)
   - `SensorID` (Foreign Key from TrafficSensor)
   - `Timestamp` (Date and time of data collection)
   - `VehicleCount` (Number of vehicles detected)
   - `AverageSpeed` (Average speed of vehicles)
   - `TrafficFlow` (Flow rate of traffic)

3. AirQualityStation
   - `StationID` (Primary Key)
   - `Location` (Geographical coordinates or address)
   - `Type` (e.g., Fixed station, Mobile unit)

4. AirQualityData
   - `DataID` (Primary Key)
   - `StationID` (Foreign Key from AirQualityStation)
   - `Timestamp` (Date and time of data collection)
   - `NOxLevel` (Nitrogen oxides level)
   - `PMLevel` (Particulate matter level)

- `O3Level` (Ozone level)

5. WeatherData
   - `WeatherID` (Primary Key)
   - `Timestamp` (Date and time of data collection)
   - `Temperature` (Temperature in degrees Celsius)
   - `Humidity` (Relative humidity percentage)
   - `WindSpeed` (Wind speed in meters per second)

6. TrafficManagementStrategy
   - `StrategyID` (Primary Key)
   - `Name` (Name of the strategy)
   - `Description` (Description of the strategy)

7. Implementation
   - `ImplementationID` (Primary Key)
   - `StrategyID` (Foreign Key from TrafficManagementStrategy)
   - `Timestamp` (Date and time of implementation)
- `Location` (Geographical location or area affected)

SQL Statements to Create the Database Schema

```sql
-- Creating TrafficSensor Table
CREATE TABLE TrafficSensor (
    SensorID INT PRIMARY KEY,
    Location VARCHAR(255),
    Type VARCHAR(50)
);

-- Creating TrafficData Table
CREATE TABLE TrafficData (
    DataID INT PRIMARY KEY,
    SensorID INT,
    Timestamp DATETIME,
    VehicleCount INT,
    AverageSpeed DECIMAL(5, 2),
    TrafficFlow DECIMAL(5, 2),
    FOREIGN KEY (SensorID) REFERENCES TrafficSensor(SensorID)
);

-- Creating AirQualityStation Table
CREATE TABLE AirQualityStation (
    StationID INT PRIMARY KEY,
    Location VARCHAR(255),
    Type VARCHAR(50)
);
```

```
-- Creating AirQualityData Table
CREATE TABLE AirQualityData (
    DataID INT PRIMARY KEY,
    StationID INT,
    Timestamp DATETIME,
    NOxLevel DECIMAL(5, 2),
    PMLevel DECIMAL(5, 2),
    O3Level DECIMAL(5, 2),
    FOREIGN KEY (StationID) REFERENCES AirQualityStation(StationID)
);

-- Creating WeatherData Table
CREATE TABLE WeatherData (
    WeatherID INT PRIMARY KEY,
    Timestamp DATETIME,
    Temperature DECIMAL(5, 2),
    Humidity DECIMAL(5, 2),
    WindSpeed DECIMAL(5, 2)
);

-- Creating TrafficManagementStrategy Table
CREATE TABLE TrafficManagementStrategy (
    StrategyID INT PRIMARY KEY,
    Name VARCHAR(255),
    Description TEXT
);

-- Creating Implementation Table
CREATE TABLE Implementation (
    ImplementationID INT PRIMARY KEY,
    StrategyID INT,
    Timestamp DATETIME,
    Location VARCHAR(255),
    FOREIGN KEY (StrategyID) REFERENCES TrafficManagementStrategy(StrategyID)
);
```

Sample Data

```sql
-- Inserting sample data into TrafficSensor
INSERT INTO TrafficSensor (SensorID, Location, Type) VALUES
(1, 'Downtown Intersection', 'Camera'),
(2, 'Main Street', 'GPS');

-- Inserting sample data into TrafficData
INSERT INTO TrafficData (DataID, SensorID, Timestamp, VehicleCount, AverageSpeed,
TrafficFlow) VALUES
(1, 1, '2024-08-13 08:00:00', 150, 25.5, 5.0),
(2, 2, '2024-08-13 08:00:00', 120, 22.0, 4.5);

-- Inserting sample data into AirQualityStation
INSERT INTO AirQualityStation (StationID, Location, Type) VALUES
(1, 'Central Park', 'Fixed station'),
(2, 'City Center', 'Mobile unit');

-- Inserting sample data into AirQualityData
INSERT INTO AirQualityData (DataID, StationID, Timestamp, NOxLevel, PMLevel, O3Level)
VALUES
(1, 1, '2024-08-13 08:00:00', 50.5, 30.0, 70.0),
(2, 2, '2024-08-13 08:00:00', 60.0, 40.0, 65.0);

-- Inserting sample data into WeatherData
INSERT INTO WeatherData (WeatherID, Timestamp, Temperature, Humidity, WindSpeed)
VALUES
(1, '2024-08-13 08:00:00', 22.5, 65.0, 3.0);

-- Inserting sample data into TrafficManagementStrategy
INSERT INTO TrafficManagementStrategy (StrategyID, Name, Description) VALUES
(1, 'Adaptive Signal Control', 'Adjusts traffic signal timings based on real-time traffic
conditions.');

-- Inserting sample data into Implementation
INSERT INTO Implementation (ImplementationID, StrategyID, Timestamp, Location) VALUES
(1, 1, '2024-08-13 09:00:00', 'Downtown Intersection');
```

Data Retrieval Queries

1. Retrieve Current Traffic Data for a Specific Sensor

```sql
SELECT *
FROM TrafficData
WHERE SensorID = 1
ORDER BY Timestamp DESC
```

```
LIMIT 1;
```

2. **Retrieve Air Quality Data for a Specific Station**

```sql
SELECT *
FROM AirQualityData
WHERE StationID = 2
ORDER BY Timestamp DESC
LIMIT 1;
```

3. **Retrieve Weather Data for a Specific Date and Time**

```sql
SELECT *
FROM WeatherData
WHERE Timestamp = '2024-08-13 08:00:00';
```

4. **Retrieve All Implementations of a Specific Traffic Management Strategy*

```sql
SELECT *
FROM Implementation
WHERE StrategyID = 1;
```

Data Analysis Queries

1. Calculate Average Vehicle Count and Average Speed by Location

```sql
SELECT T.Location, AVG(TD.VehicleCount) AS AvgVehicleCount, AVG(TD.AverageSpeed) AS AvgSpeed
FROM TrafficData TD
JOIN TrafficSensor T ON TD.SensorID = T.SensorID
GROUP BY T.Location;
```

2. Find the Peak NOx Levels and PM Levels Across All Stations

```sql
SELECT MAX(NOxlLevel) AS MaxNOxLevel, MAX(PMLevel) AS MaxPMLevel
FROM AirQualityData;
```

3. Analyze Correlation Between Traffic Flow and Pollution Levels

```sql
SELECT TD.Timestamp, TD.TrafficFlow, AQD.NOxLevel, AQD.PMLevel
FROM TrafficData TD
JOIN AirQualityData AQD ON TD.Timestamp = AQD.Timestamp
WHERE TD.SensorID = 1 AND AQD.StationID = 1
ORDER BY TD.Timestamp;
```

4. Identify the Effectiveness of Traffic Management Strategies

```sql
SELECT I.Location, TS.Name AS StrategyName, AVG(TD.VehicleCount) AS
AvgVehicleCountBefore, AVG(TD2.VehicleCount) AS AvgVehicleCountAfter
FROM Implementation I
JOIN TrafficManagementStrategy TS ON I.StrategyID = TS.StrategyID
JOIN TrafficData TD ON I.Location = TD.SensorID
JOIN TrafficData TD2 ON TD.Timestamp < I.Timestamp AND TD2.Timestamp >= I.Timestamp
WHERE TD.SensorID = TD2.SensorID
GROUP BY I.Location, TS.Name;
```

5. Determine Pollution Levels on Days with High Traffic Congestion

```sql
SELECT TD.Timestamp, TD.VehicleCount, AQD.NOxLevel, AQD.PMLevel
FROM TrafficData TD
JOIN AirQualityData AQD ON TD.Timestamp = AQD.Timestamp
WHERE TD.VehicleCount > (SELECT AVG(VehicleCount) FROM TrafficData)
ORDER BY TD.VehicleCount DESC;
```