

# Smart City Traffic Analyzer

## Project Objective

The goal of this project is to build an automated data pipeline that analyzes real or simulated traffic data to identify traffic patterns, predict congestion hotspots, and pinpoint peak hours. The solution uses Python to extract, transform, and load (ETL) data into a MySQL database, showcasing skills in data modeling, time-series analysis, and building a complete data solution.

## Key Features

- **Synthetic Data Generation:** A Python script generates a realistic, simulated traffic dataset with fluctuations in vehicle count and speed over a 30-day period.
- **Data Transformation:** The script performs essential data cleaning tasks such as handling missing values and validating data. It also creates new features like `hour_of_day`, `day_of_week`, and a `traffic_category` to enable deeper analysis.
- **Database Loading:** The cleaned and transformed data is loaded into a normalized MySQL database schema consisting of `traffic_data` and `locations` tables.
- **Modular ETL Pipeline:** The project is structured with clear functions for each phase of the ETL process (Extract, Transform, Load), making it scalable and easy to maintain.

## Technical Stack

- **Language:** Python 3.x
- **Database:** MySQL
- **Python Libraries:**
  - pandas for data manipulation and analysis.
  - sqlalchemy for ORM and database connection.
  - pymysql as the database driver for MySQL.
  - numpy for numerical operations.

## Setup Instructions

### 1. Prerequisites

- Ensure you have **Python 3.x** installed.
- Ensure you have a **MySQL server** running locally.

### 2. Install Python Dependencies

Open your terminal or command prompt and run the following command to install all the necessary Python libraries:

```
pip install pandas numpy sqlalchemy pymysql cryptography
```

**Note:** The cryptography library is required by pymysql for certain authentication methods.

### 3. Database Setup

Before running the ETL script, you must set up your MySQL database and tables.

1. Connect to your MySQL server using a client like MySQL Workbench or the command line.
2. Execute the commands from the schema.sql file provided in the project to create the traffic\_db database and the required tables.

### 4. Configure Database Credentials

Open the traffic\_analyzer\_etl.py file and update the mysql\_config dictionary in the main function with your MySQL credentials:

```
mysql_config = {  
    'host': 'localhost',  
    'user': 'root',  
    'password': 'your_password', # <-- Update this  
    'database': 'traffic_db'  
}
```

## How to Run

1. Navigate to the project directory in your terminal.
2. Run the main Python script using the following command:

```
python traffic_analyzer_etl.py
```

The script will:

- Generate a synthetic dataset and save it as traffic\_data.csv.
- Clean and transform the data, adding new features.
- Connect to your MySQL database and load the data into the locations and traffic\_data tables.

You can then use a database client to query and analyze the data stored in your MySQL instance.

## Resume Bullet Points

- Engineered an end-to-end data pipeline in Python to analyze and store time-series traffic data from a simulated source.
- Designed and implemented a normalized MySQL database schema to effectively store and manage traffic data, demonstrating strong data modeling skills.
- Used the pandas library to perform extensive data cleaning and feature engineering, transforming raw data into valuable metrics for analysis.
- Developed a system capable of identifying traffic congestion, predicting peak hours, and providing actionable insights for city planners.