رواد مصر الرقمية

Real-Time Traffic Analytics Using Azure Stream Analytics

Project Information

Project Title: Real-Time Traffic Analytics Using Azure Stream Analytics Course: AI & Data Science Microsoft Data Engineer – DEBI
Round: 3

Group Members & Roles

- **Abdalla Nasser Bekhit Abdalla Ewida:** Team Leader , Data Simulation & Python Integration
- **Nour Hassan Hamdy Gamil:** System Testing & Quality Assurance
- **Menna Muhammad Ali Eissa:** Visualization Lead - Power BI
- **Zahran Alaa Sayed Mohammed:** Data Engineering & SQL Queries
- **Rawan Khaled Mohamed Mustafa:** Data Engineering & SQL Queries
- **Sohaila Ahmed Ali Elgmal:** Azure Cloud & Pipeline Architect

Project Description

This project aims to build an end-to-end system for the real-time analysis of traffic in Greater Cairo. The system will be based on a realistic data source generated by a custom Python script, which creates data that simulates traffic flow, including real street names, various vehicle types, and realistic speed patterns.

This streaming data will be sent to Microsoft Azure for processing using Azure Stream Analytics. The resulting insights and processed data will then be stored in a cloud database. Finally, key performance indicators (such as congestion zones, average speeds, and alerts) will be displayed on an interactive dashboard using Power BI.

Objectives

- **Activate and utilize** the script to generate realistic, streaming traffic data.
- **Build a data pipeline** to receive data from the script and ingest it into Azure Event Hubs.
- **Develop queries** in Azure Stream Analytics to analyze the data in real-time and detect important patterns (e.g., congestion, accidents).
- **Store** the processed data and results in an Azure SQL Database.
- **Design and develop** an interactive dashboard in Power BI to display live metrics and alerts.

Tools & Technologies

- **Data Source:** Python Script.
- **Data Ingestion:** Azure Event Hubs.
- **Real-Time Processing:** Azure Stream Analytics.
- **Storage:** Azure SQL Database.
- **Visualization:** Power BI.

Milestones & Deadlines

Milestone	Key Tasks	Expected Deliverables	Proposed Deadline
Milestone 1: Data Ingestion Setup	- Ensure the simulation script is fully operational. - Create an Azure Event Hub. - Modify the script to send data directly to the Event Hub.	- A modified Python script. - An Event Hub successfully receiving data.	October 13, 2025
Milestone 2: Real-Time Processing	- Connect Stream Analytics to the Event Hub. - Write queries to analyze data (e.g., average speed, traffic jams). - Connect Stream Analytics output to an Azure SQL database.	- An active Stream Analytics Job. - Tables in Azure SQL populated with processed data.	November 20, 2025
Milestone 3: Visualization & Final Report	- Connect Power BI to the Azure SQL database. - Design and develop the dashboard. - Prepare the final project report and presentation.	- An interactive dashboard. - The final report in PDF format. -	December 27, 2025

Stakeholder Analysis

This project involves several stakeholders who play important roles in its development and potential use.

The project team members are responsible for designing, developing, and testing the full data pipeline — from data simulation using Python to real-time processing on Azure Stream Analytics and visualization in Power BI.

The mentors and supervisors provide technical guidance and ensure the project meets academic and quality standards.

The Ministry of Communications and Information Technology (MCIT), through the Digital Egypt Pioneers Initiative (DEPI), acts as the project sponsor and platform provider, offering the environment and tools for implementation and evaluation.

The traffic authorities represent potential end users who could benefit from the final analytical dashboard to monitor traffic conditions and identify congestion trends in real time.

Finally, the general public benefits indirectly from improved traffic management and decision-making supported by the project's analytical insights.

Database Design

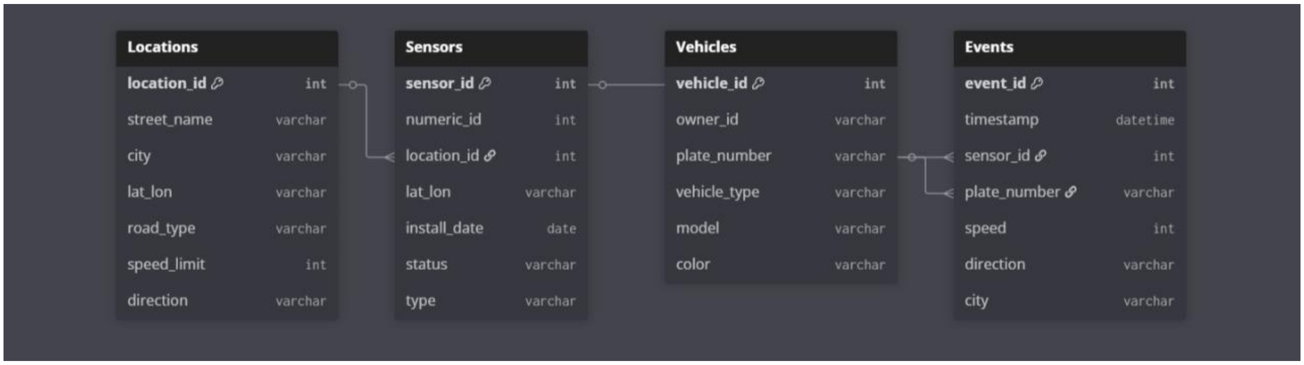
The database consists of four main tables: Locations, Sensors, Vehicles, and Events. These tables are designed to store and manage traffic monitoring data efficiently.

- **Locations Table:**
  Stores details about different road segments, including street name, city, geographic coordinates, road type, speed limit, and direction.
- **Sensors Table:**
  Contains information about sensors installed at various locations. Each sensor is linked to a specific location through the location_id field and includes details such as installation date, type, and operational status.
- **Vehicles Table:**
  Holds records of registered vehicles, including their unique plate numbers, owner IDs, vehicle types, models, and colors.
- **Events Table:**
  Records traffic events detected by the sensors. Each event includes a timestamp, the related sensor ID, the vehicle's plate number, the recorded speed, direction, and city of occurrence.

Relationships:

- Each location can have multiple sensors (one-to-many).
- Each sensor can generate multiple events (one-to-many).
- Each vehicle can be involved in multiple events (one-to-many).

This relational structure allows efficient tracking of traffic activity, sensor performance, and vehicle movement across various locations.



Locations	
location_id 🔑	int
street_name	varchar
city	varchar
lat_lon	varchar
road_type	varchar
speed_limit	int
direction	varchar

Sensors	
sensor_id 🔑	int
numeric_id	int
location_id 🔗	int
lat_lon	varchar
install_date	date
status	varchar
type	varchar

Vehicles	
vehicle_id 🔑	int
owner_id	varchar
plate_number	varchar
vehicle_type	varchar
model	varchar
color	varchar

Events	
event_id 🔑	int
timestamp	datetime
sensor_id 🔗	int
plate_number 🔗	varchar
speed	int
direction	varchar
city	varchar

UI/UX Design

The project's user interface is designed using Power BI to display real-time traffic data processed through Azure Stream Analytics.
The dashboard focuses on showing key metrics and charts that help understand traffic patterns clearly.

It includes:

- A panel showing total events, average speed, and traffic density.
- Charts for speed distribution and vehicle types.
- A time-based graph showing how traffic changes over time.
- A live event table listing the latest traffic data received.

The design is simple, data-focused, and updates automatically as new events are streamed from Azure.

KPIs (Key Performance Indicators)

“Note: KPIs will be measured after the completion of Azure integration and Power BI visualization in later project phases.”

1. Data Ingestion & Pipeline

KPI	Description	Measurement Plan
Successful ingestion of generated events into Azure Event Hubs	Measure the percentage of events successfully transmitted from the Python script to Azure Event Hub.	After Event Hub integration – compare sent vs. received event count.
End-to-end data latency (script → Azure SQL)	Time difference between event generation and storage in Azure SQL.	To be measured after pipeline deployment using Event Hub + Stream Analytics latency metrics.

2. SQL Integration & Queries

KPI	Description	Measurement Plan
Query accuracy for traffic jam detection	Evaluate accuracy of SQL logic that identifies congestion based on event data.	Validate with test cases after data is stored in SQL.
Query performance	Average execution time of analytical queries on Azure SQL.	Measure using SQL Profiler or Query Insights after schema creation.

3. Visualization (Dashboard)

KPI	Description	Measurement Plan
Dashboard data refresh time	Time taken for dashboard to refresh with latest event data.	Measure once Power BI or Synapse dashboard is connected.
Dashboard load time	Time for dashboard to fully load visual elements.	Evaluate after deployment of the visualization layer.

4. Presentation (Report / Slide Deck)

KPI	Description	Measurement Plan
Report completeness	Evaluate if final report covers all project phases and deliverables.	Review during final submission.
Stakeholder clarity/feedback score	Assess clarity and quality of final presentation.	Feedback collected from mentors or evaluators.