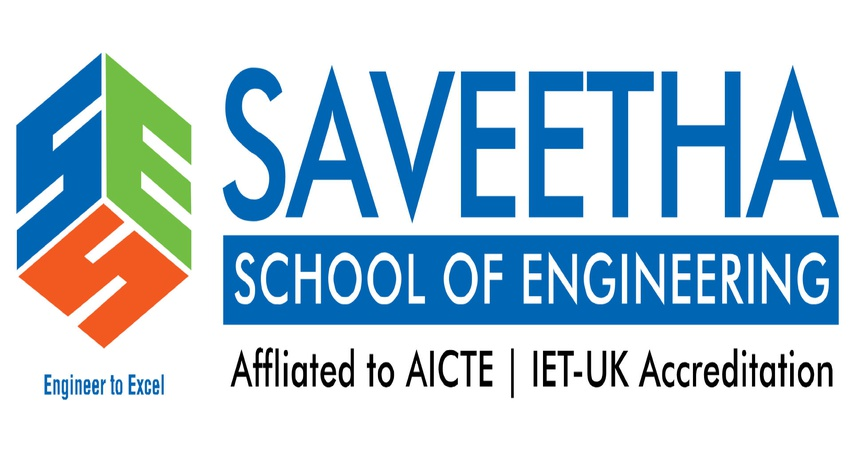
****

**Assignment - 1**

**SAVEETHA SCHOOL OF ENGINEERING**



Submitted by

**PHILIP GRACIAN SANOVA S (192324023)**

Submitted to

**Dr. Christy Melwyn**

Professor

Course Code: **CSA0556**

Course Name: **Database Management Systems for Relational Database**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Contents** | **Page no** |
| 1.1 | Introduction | 3 |
| 1.2 | Background and Objectives | 3 |
| 1.3 | System Design | 4 |
|  | -Entity-Relationship (ER) Diagram | 4 |
| -Entity Definitions | 5 |
| -Relationship Modelling | 5 |
| 1.4 | Implementation | 6 |
|  | -Database Setup | 6 |
| -Data Collection and Integration | 8 |
| 1.5 | Justification and Normalization | 8 |
|  | -Design Choices | 8 |
| -Normalization Principles | 9 |
| 1.6 | Conclusion | 11 |

**1.1 Introduction**

In urban areas, the issue of traffic congestion is acute causing huge travel time delays and resultant increase in fuel consumption as well as pollution. The city administration has opted to come up with a Traffic Flow Management System (TFMS) in order to address these challenges. Using real-time data from traffic sensors and cameras along with historical traffic patterns, the TFMS is designed to help improve transportation efficiency. This method will manage route optimization, intersection control and signal choreography in real time to enhance the mobility of transportation.

The TFMS is designed as an intelligent traffic management solution to alleviate congestion and enhance urban commuting experience for city residents. When real-time traffic data is factored in with historical reports the system will be able to recommend the most efficient routes, alter signal timings and predict appropriate infrastructure improvement. This is a useful inclusive step towards traffic management that adapts to the increasing needs of urban mobility in order for roads within and out side city to run smoothly.

Overview of TFMS design, development and implementation This starts by diving into the system design, which includes defining an entity-relationship (ER) diagram to define relationships between elements in roads,, intersections, traffic signals and traffic data. Next up is the implementation section, including creating databases and how to integrate data. Secondly, it explains the motivation behind the design choices done in this implementation like scale out architecture with real-time data processing and maintaining 3NF normalization properties. Lastly, the report provides a wrap up of basic points and recommendations for further improvements on this system.

**1.2 Background and Objectives**

**Background:**

The city is on the brink of severe traffic congestion, adding large number of vehicles to existing fleet just adds more time spent in travel as well as increase fuel wastage and departments environment pollution levels higher. Existing traffic monitoring and control infrastructure does not adjust well to changing conditions in real time, resulting in inefficiencies and delays. The proliferation of the problem requires a smart, dynamic approach towards better traffic management targeted at diffusing congestion on roads and enhancing urban mobility.

**Objectives:**

The primary objectives of implementing the Traffic Flow Management System (TFMS) are threefold:

1. **Optimizing Traffic Routes**: Utilizing real-time and historical data to suggest the most efficient travel paths.
2. **Managing Intersections**: Enhancing the control and monitoring of key junctions to ensure smoother traffic flow.
3. **Controlling Traffic Signals**: Implementing adaptive signal control algorithms to adjust signal timings dynamically based on current traffic conditions. These measures aim to reduce congestion, minimize delays, and enhance overall transportation efficiency within the city.

**1.3 System Design**

**Entity-Relationship (ER) Diagram**

The ER diagram for the Traffic Flow Management System (TFMS) illustrates the entities involved and their relationships. Here is a textual representation of the ER diagram:

#### 

#### Entity Definitions

**1. Roads**

* **RoadID (PK)**: Unique identifier for each road.
* **RoadName**: Name of the road.
* **Length**: Length of the road in meters.
* **SpeedLimit**: Speed limit on the road in km/h.

**2. Intersections**

* **IntersectionID (PK)**: Unique identifier for each intersection.
* **IntersectionName**: Name of the intersection.
* **Latitude**: Geographic latitude of the intersection.
* **Longitude**: Geographic longitude of the intersection.

**3. Traffic Signals**

* **SignalID (PK)**: Unique identifier for each traffic signal.
* **IntersectionID (FK)**: Identifier for the intersection where the signal is located.
* **SignalStatus**: Current status of the signal (Green, Yellow, Red).
* **Timer**: Countdown timer to the next signal change.

**4. Traffic Data**

* **TrafficDataID (PK)**: Unique identifier for each traffic data entry.
* **RoadID (FK)**: Identifier for the road where the data is collected.
* **Timestamp**: Date and time when the data is recorded.
* **Speed**: Average speed of vehicles on the road.
* **CongestionLevel**: Degree of traffic congestion.

#### Relationship Modelling

**1. Roads to Intersections**

* **Relationship**: One-to-Many
* **Description**: A road can connect to multiple intersections, and an intersection can be connected by multiple roads.
* **Cardinality**: One road can connect to many intersections, and each intersection can be connected by many roads.

**2. Intersections to Traffic Signals**

* **Relationship**: One-to-Many
* **Description**: An intersection can host multiple traffic signals, but each traffic signal is installed at a specific intersection.
* **Cardinality**: One intersection can have many traffic signals, but each traffic signal is at one intersection.

**3. Roads to Traffic Data**

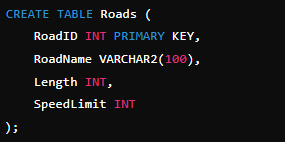
* **Relationship**: One-to-Many
* **Description**: A road can have multiple traffic data records collected over time.
* **Cardinality**: One road can have many traffic data records.

**1.4 Implementation**

**Database Setup**

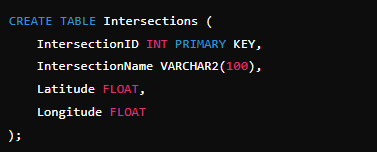
Using Oracle Apex to create setup Database

**Roads Table:**



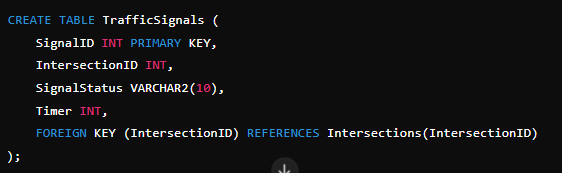


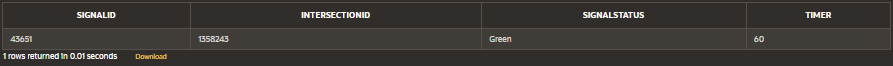
**Intersection Table:**

****

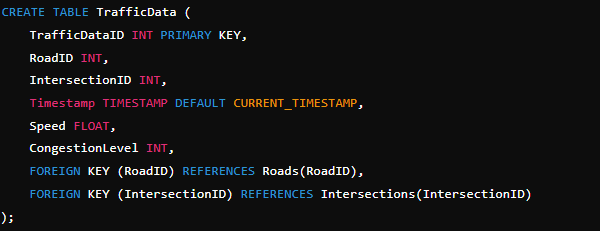
****

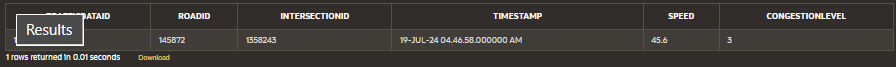
**TrafficSignals Table:**

****

****

**TrafficData Table:**

****

****

#### Data Collection

1. **Install Traffic Sensors and Cameras:**
   * **Traffic Sensors**: Install traffic sensors (e.g., inductive loop sensors, radar sensors) on roads and intersections to capture real-time data such as vehicle count, speed, and congestion levels.
   * **Cameras**: Deploy traffic cameras at strategic locations to monitor traffic flow, detect incidents, and collect additional data such as vehicle types and license plate numbers.
2. **Data Acquisition Systems:**
   * **Sensor Networks**: Use wireless or wired networks to connect sensors and cameras to central data processing units.
   * **Edge Computing**: Implement edge devices to preprocess data near the source, reducing latency and the amount of data transmitted to the central server.
3. **Real-time Data Transmission:**
   * **Communication Protocols**: Utilize communication protocols such as MQTT, HTTP, or WebSockets for real-time data transmission from sensors and cameras to the central server.
   * **Data Aggregators**: Use data aggregator devices or software to collect data from multiple sensors and cameras and send it to the central server.
4. **Data Storage:**
   * **Database Management System (DBMS)**: Store collected data in a central database (Oracle APEX) for further processing and analysis.

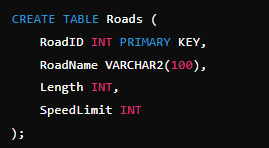
### 1.5 Justification and Normalization

#### Design Choices

1. **Entity Definitions:**
   * **Roads**: This entity captures all the critical attributes of roads such as RoadID, RoadName, Length, and SpeedLimit. It uniquely identifies each road and provides essential information for route optimization and traffic management.
   * **Intersections**: This entity is vital for managing points where roads meet. Attributes like IntersectionID, IntersectionName, Latitude, and Longitude help in accurately identifying and managing intersections.
   * **Traffic Signals**: This entity manages the state of traffic signals at intersections. Attributes like SignalID, IntersectionID, SignalStatus, and Timer are essential for adaptive signal control based on real-time traffic conditions.
   * **Traffic Data**: This entity captures real-time data collected from sensors and cameras, including TrafficDataID, RoadID, IntersectionID, Timestamp, Speed, and CongestionLevel. This data is critical for both real-time traffic management and historical analysis.
2. **Relationships:**
   * **Roads to Intersections**: Each road can connect to multiple intersections, enabling the modeling of complex road networks.
   * **Intersections to Traffic Signals**: Each intersection can have multiple traffic signals, allowing for detailed control and management of traffic flows.
   * **Traffic Data Integration**: Linking real-time traffic data to both roads and intersections ensures that the system can utilize data effectively for route optimization and signal control.
3. **Primary and Foreign Keys:**
   * **Primary Keys**: Each table has a primary key (e.g., RoadID, IntersectionID, SignalID, TrafficDataID) to uniquely identify records.
   * **Foreign Keys**: Relationships between tables are enforced through foreign keys (e.g., IntersectionID in TrafficSignals and TrafficData, RoadID in TrafficData) to maintain data integrity and enable efficient data retrieval.
4. **Scalability and Real-time Data Processing:**
   * The design supports the scalable collection of real-time traffic data from multiple sources, enabling efficient traffic management.
   * Real-time data integration through APIs and automated jobs ensures the system is up-to-date, facilitating dynamic traffic control and optimization.

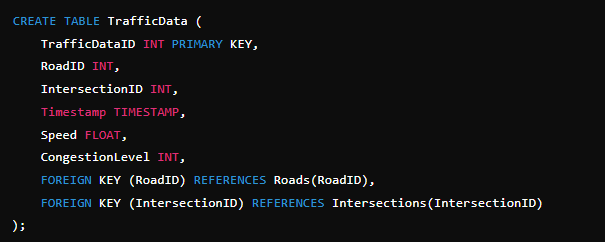
#### Normalization Principles

1. **First Normal Form (1NF):**
   * Ensures that all tables have a primary key and that each column contains atomic (indivisible) values.
   * Each table's columns store only a single value (e.g., RoadName, Latitude).



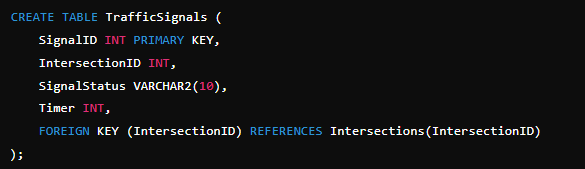
1. Second Normal Form (2NF):

* Ensures that all non-primary key attributes are fully functionally dependent on the primary key.
* Eliminates partial dependency by ensuring that attributes depend on the entire primary key



1. **Third Normal Form (3NF)**:

* Ensures that all attributes are not only fully functionally dependent on the primary key but also independent of each other.
* Removes transitive dependency by ensuring that non-primary key attributes do not depend on other non-primary key attributes.



### Justification for Adhering to Normalization Principles

1. **Minimized Data Redundancy**: By normalizing the database, redundancy is minimized, ensuring that the same piece of data is not stored in multiple places, which reduces storage costs and potential inconsistencies.
2. **Improved Data Integrity**: Foreign keys and normalization ensure that relationships between entities are maintained correctly, enhancing data integrity and consistency across the database.
3. **Enhanced Query Performance**: Normalized databases often have smaller, more focused tables, which can lead to improved query performance as data retrieval is more efficient.
4. **Ease of Maintenance**: A normalized database structure is easier to maintain and update, as changes to data structures are localized to specific tables, reducing the risk of unintended side effects.
5. **Scalability**: A normalized design supports scalability by allowing the database to handle increasing amounts of data without significant redesign, facilitating growth and expansion of the system.

**1.6 Conclusion**

The Traffic Flow Management System (TFMS) has been designed to address the growing traffic congestion issues by optimizing traffic routes, managing intersections, and controlling traffic signals through real-time data integration. The ERD developed outlines a robust schema that ensures effective management of traffic data, integrating tables for roads, intersections, traffic signals, and real-time traffic data. The system design leverages triggers and default values to automate the management of timestamps, ensuring accurate and up-to-date records. By implementing these design choices and normalization principles, the TFMS aims to enhance transportation efficiency and contribute to better urban traffic management.