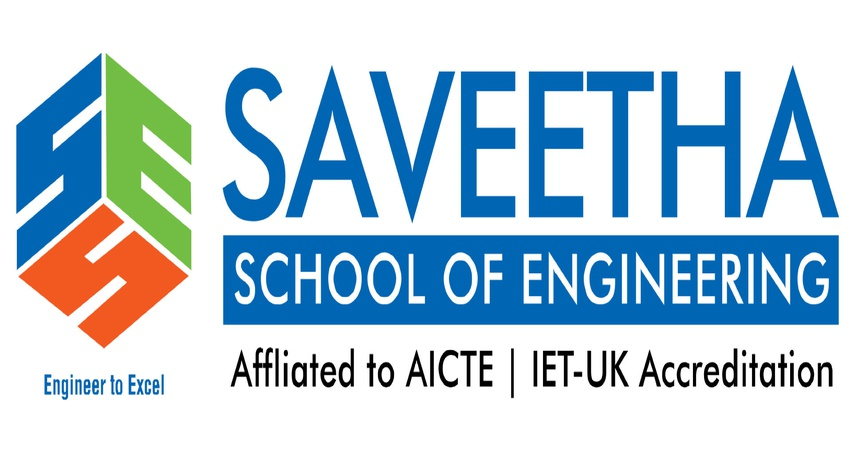
****

**Assignment - 1**

**SAVEETHA SCHOOL OF ENGINEERING**



Submitted by

**PADMANABHAN S (192324076)**

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**

The traffic congestion is a severe issue, causing significant travel delays and leading to increased fuel consumption and pollution. To address these challenges, the city administration has decided to implement a Traffic Flow Management System (TFMS). By utilizing real-time data from traffic sensors and cameras, along with historical traffic patterns, the TFMS aims to improve transportation efficiency. This system will optimize routes, control intersections, and choreograph signals in real time to enhance mobility.

The TFMS is an intelligent traffic management solution designed to alleviate congestion and improve the urban commuting experience for city residents. By incorporating real-time traffic data with historical reports, the system can recommend the most efficient routes, adjust signal timings, and predict necessary infrastructure improvements. This adaptive approach to traffic management addresses the growing needs of urban mobility, ensuring smooth traffic flow within and outside the city.

**Overview of TFMS Design, Development, and Implementation**

The process begins with system design, which involves creating an entity-relationship (ER) diagram to define relationships between elements such as roads, intersections, traffic signals, and traffic data. The implementation phase includes developing databases and integrating data. Additionally, it explains the rationale behind design choices, such as using a scalable architecture with real-time data processing and maintaining 3NF normalization properties. Finally, the report concludes with a summary of key points and recommendations for further improvements to the system.

**1.2 Background and Objectives**

#### Background:

The city is on the verge of severe traffic congestion, with the increasing number of vehicles exacerbating travel times, fuel wastage, and environmental pollution. The current traffic monitoring and control infrastructure does not adapt well to changing conditions in real time, leading to inefficiencies and delays. This escalating problem necessitates a smart, dynamic approach to traffic management, aimed at reducing road congestion 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:

#### 

## Task 1: Entity Identification and Attributes

### Entities and Attributes

1. **Roads**
   * RoadID (PK)
   * RoadName
   * Length (in meters)
   * SpeedLimit (in km/h)
2. **Intersections**
   * IntersectionID (PK)
   * IntersectionName
   * Latitude
   * Longitude
3. **Traffic Signals**
   * SignalID (PK)
   * IntersectionID (FK)
   * SignalStatus (Green, Yellow, Red)
   * Timer (countdown to next change)
4. **Traffic Data**
   * TrafficDataID (PK)
   * RoadID (FK)
   * Timestamp
   * Speed (average speed on the road)
   * CongestionLevel (degree of traffic congestion).

## Task 2: Relationship Modeling

### Relationships

1. **Roads to Intersections**
   * A road can connect to multiple intersections.
   * An intersection can connect to multiple roads.
   * **Cardinality:** Many-to-Many (Requires a junction table: RoadIntersections)
2. **Intersections to Traffic Signals**
   * An intersection can host multiple traffic signals.
   * A traffic signal is located at one intersection.
   * **Cardinality:** One-to-Many
3. **Roads to Traffic Data**
   * A road can have multiple traffic data entries.
   * Traffic data is related to one road.
   * **Cardinality:** One-to-Many

### Optionality Constraints

* **Roads to Intersections:** Mandatory for both entities (a road must connect to intersections, and an intersection must be part of the road network)
* **Intersections to Traffic Signals:** Mandatory for intersections (an intersection must have traffic signals), optional for traffic signals (not all intersections may have traffic signals)
* **Roads to Traffic Data:** Optional for roads (not all roads may have traffic data), mandatory for traffic data (each traffic data must be related to a road).

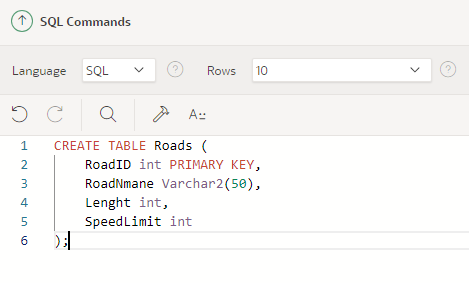
**1.4 Implementation**

**Database Setup**

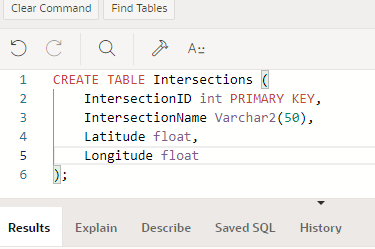
Using Oracle Apex to create Database

Using Oracle SQL DATA MODELER

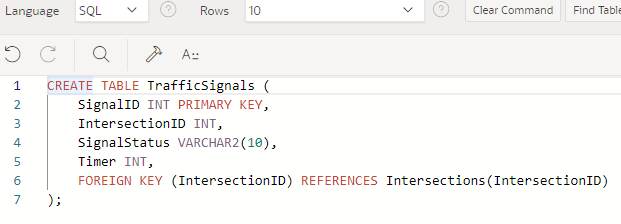
**Roads Table:**



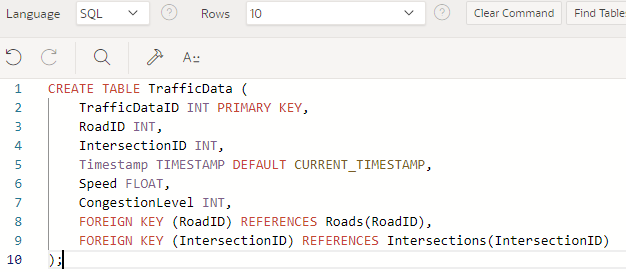
**Intersection Table:**

****

**TrafficSignals Table:**

****

**TrafficData Table:**

****

### Data Collection :

1. **Install Traffic Sensors and Cameras:**
   * **Traffic Sensors:** Utilize inductive loop sensors, radar sensors, etc., 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 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 (e.g., Oracle APEX) for further processing and analysis.

### Justification and Normalization :

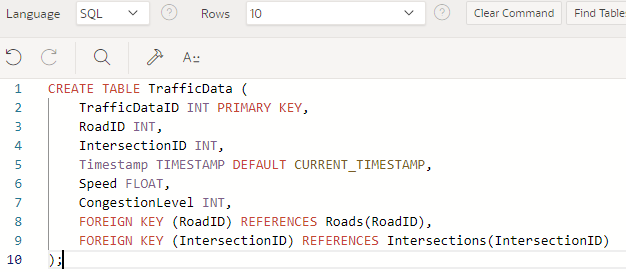
1. **Design Choices:**
   * **Entity Definitions:**
     + **Roads:** Attributes like RoadID, RoadName, Length, and SpeedLimit uniquely identify each road and provide essential information for route optimization and traffic management.
     + **Intersections:** Attributes like IntersectionID, IntersectionName, Latitude, and Longitude help in accurately identifying and managing intersections.
     + **Traffic Signals:** Attributes like SignalID, IntersectionID, SignalStatus, and Timer are essential for adaptive signal control based on real-time traffic conditions.
     + **Traffic Data:** Attributes like TrafficDataID, RoadID, IntersectionID, Timestamp, Speed, and CongestionLevel capture real-time data collected from sensors and cameras.
   * **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 effective utilization of data for route optimization and signal control.
   * **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.
2. **Scalability and Real-time Data Processing:**
   * The design supports 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.

This structured approach ensures a comprehensive system for traffic data collection, processing, and utilization, ultimately aiding in efficient traffic management and optimization.

#### Normalization Principles :

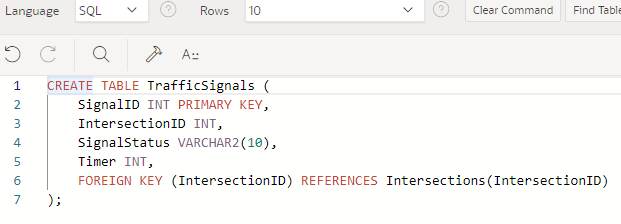
1. First Normal Form (1NF)**:**
   * Ensures that all tables have a primary key and that each column contains atomic (indivisible) values.
   * Each table's columns store only a single value (e.g., RoadName, Latitude).
2. Second Normal Form (2NF):

* 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

### Minimized Data Redundancy

1. By normalizing the database, redundancy is minimized, ensuring that the same piece of data is not stored in multiple places. This reduces storage costs and potential inconsistencies. For example, the information about a road's name and speed limit is stored only once in the Roads table, preventing duplication and inconsistencies across the database.

### Improved Data Integrity

1. Normalization, along with the use of foreign keys, ensures that relationships between entities are maintained correctly. This enhances data integrity and consistency across the database. For instance, the IntersectionID in the Traffic Signals table is a foreign key that references the Intersections table, ensuring that every traffic signal is linked to a valid intersection.

### Enhanced Query Performance

1. Normalized databases often have smaller, more focused tables, which can lead to improved query performance as data retrieval is more efficient. By splitting data into related tables and reducing redundancy, queries can be executed faster. For example, querying for all traffic data related to a specific road can be done efficiently without unnecessary data processing.

### Ease of Maintenance

1. A normalized database structure is easier to maintain and update. Changes to data structures are localized to specific tables, reducing the risk of unintended side effects. For example, updating the speed limit of a road only requires changes in the Roads table, without affecting other related tables.

### Scalability

1. A normalized design supports scalability by allowing the database to handle increasing amounts of data without significant redesign. This facilitates growth and expansion of the system. For instance, adding new roads or intersections can be easily accommodated without altering the existing database structure significantly, allowing the system to scale smoothly with the city's traffic management needs.
2. By adhering to these normalization principles, the Traffic Flow Management System (TFMS) will be robust, efficient, and capable of handling real-time data and historical analysis, thereby optimizing traffic management in the city.

## 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. The adherence to normalization principles minimizes data redundancy, improves data integrity, enhances query performance, eases maintenance, and supports scalability. Overall, the TFMS is poised to significantly improve the city's traffic flow and provide a foundation for future transportation planning and improvements.