CMPG321: Integrated Traffic Management Database System 2023

Zubair Ameer

37200690

Tanyaradzwa Mandizvidza

34486844

Kondwani Chindongo

34822259

Bachelor of Science in Information Technology

North West University Vaal

Deliverable 1: User Requirements Analysis, Conceptual Database Design Model

Due Date: 25 August 2023

Table of contents

Business Rules

Zubair Ameer

Title	Business Rule	Justification
Real-time traffic data collection	Throughout the campuses, the system must continuously gather and update real-time traffic data from user input and cameras.	For accurate traffic monitoring, problem detection, and giving users the most recent information, real-time data is crucial.
Incident priority and response	Hazardous material spills and accidents involving injuries should be dealt with first in the incident response queue because they pose the highest safety hazards.	Prioritizing occurrences according to safety concerns provides quick response to urgent circumstances, protecting lives and reducing possible risks.
Historical Data	To facilitate long-term trend analysis and future planning, historical traffic data should be recorded.	For identifying traffic patterns, assessing the success of interventions, and making wise decisions for campus growth, long-term historical data analysis is essential.
User Privacy	Access should be limited to authorized workers users, such as license plate numbers and personal information, should be securely encrypted and preserved.	It is both ethical and legal to protect user privacy. Sensitive personal data is protected against unauthorized use via encryption and restricted access.
User Notifications	Any planned road closures or repair tasks that can affect	Users can plan alternate routes and reduce inconvenience by

	traffic should be made users aware of in advance.	receiving timely warnings about road closures or maintenance activity.
Data sharing	The NWU Smart Campus program can use the traffic data gathered by the system for integrated analysis and spatial visualization.	Data synergy is improved by sharing traffic data with the larger Smart Campus effort. This enables thorough analysis, which can result in more effective urban planning and transportation plans. Upkeep of the system on a regular basis.
System Maintenance	To guarantee optimum performance, accurate data, and security, the system should undergo routine maintenance and updates.	Users receive correct and up-to-date information as a result of routine maintenance and upgrades that keep the system secure, dependable, and efficient.

Tanyaradzwa Mandizvidza

Source	Business Rule	Justification
Background description	Each incident needs to be categorized according to its category (such as an accident, traffic jam, roadblock, or weather-related incident).	For efficient traffic management, resource allocation, predictive analytics, user communication, and well-informed decision-making, incident categorization is crucial.
Road safety and eco-friendly commuting data.	Routes for pedestrians must be directly related to security features like crosswalks and traffic lights.	In order to inforce pedestrian safety. Pedestrian pathways must include protective measures. Integrating these measures with traffic lights and crosswalks improves safety and promotes environmentally responsible transportation.
Data related to project's objective to optimize parking and support eco-friendly	Users must be given access to real-time information about parking availability.	Real-time information on parking availability enables users to make wise decisions, decreasing traffic caused by parking, and improving environmentally friendly travel. The project's objective to maximize parking and promote environmentally

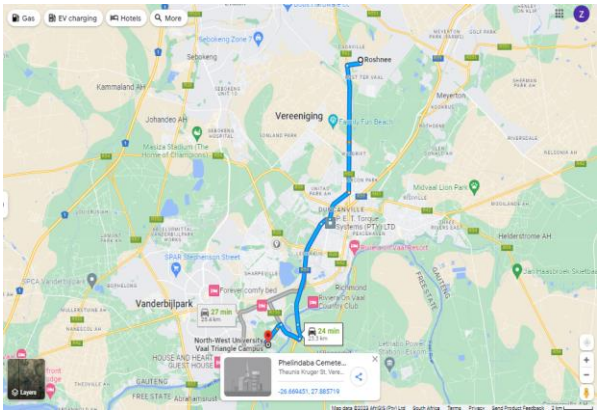
		friendly transportation methods is supported by this rule.
Traffic Studies and Reports:	The database must incorporate information from a variety of sources, such as traffic cameras, sensors, and user reviews.	A thorough comprehension of traffic patterns can be obtained by integrating data from numerous traffic data sources, such as sensors, cameras, and user comments. Real-time data analysis enables proactive traffic management and enables informed decision-making.
Data collected from environmental Factors	The database should record occurrences that happen during special events and examine how they affect traffic.	The proactive event management and traffic flow optimization are supported by observing incidents during special events and the analysis of their effects.
Data collected from Smart City Initiatives	In order to study how load shedding affects traffic, the database should link incidents to those events.	It is easier to figure out how load shedding affects traffic patterns when occurrences are linked to those events. Traffic management authorities can put necessary measures in place to lessen disruptions after analysing these effects.
Data collected from NWU students	Users must be able to comment on traffic-related concerns on the system and generate reports on traffic flow, congestion, and safety occurrences.	Users' contributions improve traffic management decisions, and the reports that are produced provide insightful analytics for future enhancements.
Requirement to implementing route planning algorithms.	Based on user preferences and real-time traffic information, the system can recommend the best routes.	It improves user experience and eases congestion to suggest the best routes based on real-time data.
Requirement of ensuring data privacy.	To secure user data and critical traffic data, data privacy and security measures must be put in place.	Data privacy and security measures are put in place to protect user information and critical traffic data.

Kondwani Chindongo

Type	Business Rule	Justification
University Affiliate	A university affiliate may be a Student, Academic Staff, General Worker or Visitor	Users who travel to and from the campus need to be identified to ensure that efficient routes are used and accurate information is acquired.
Vehicle	A university affiliate uses public or private transportation to get to campus. Public transport includes shuttles and taxis,private transport includes cars,bicycles,ubers or walking	The type of transport and vehicle used needs to be identified to plan for the routes that may be used by them as well as planning for how many users travel by that means of transportation
Distance	A university affiliate may live far from the campus or close to the campus,impacting the type of transportation they choose to use to get to campus.	The distance a user lives from campus impacts the likelihood of them using a certain type of transportation and thus needs to be identified.
Traffic Route	Depending on the type of transportation used, the traffic route for that particular type of transport may be different.	Different types of transportation occurs on different routes and thus needs to be logged to track the activities along routes used.
Parking	Transportation needs to park at the correct parking to avoid congestation.	To ensure that vehicles park in their correct parking spots for reasons of order and efficiency parking is monitored.
Incident report	At times their may be accidents at the intersections or traffic lights may stop working and incident reports need to be logged.	It is extremely important to know what incidents occur and when in order to rectify them as soon as possible and ensure that safety for vehicles traveling along those routes is ensured and to notify users to avoid routes where incidents have occurred.
Traffic lights	Transport travels through different traffic lights on their routes on the way to campus.	Knowing whether traffic lights are working or not is important to minimize the occurrence of incidents
Intersection	Transport also travels through intersections along the routes used.	Intersections are the most likely place for incidents to occur and thus they need to be monitored.

Map Routes

Zubair Ameer



Business Rules:

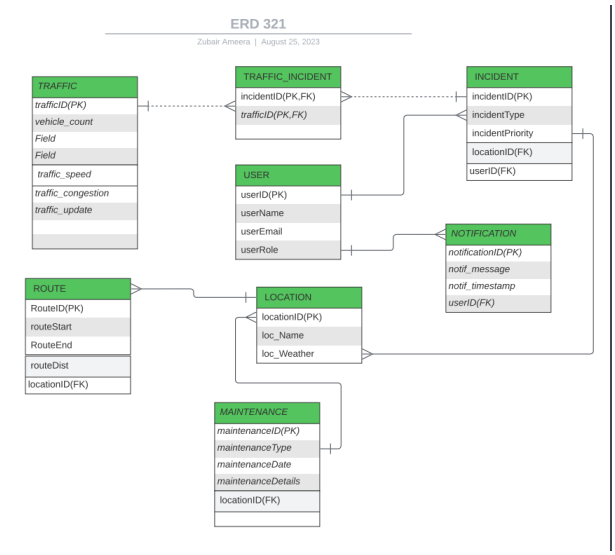
One user can report many incidents (1 to Many).

One user can receive many notifications (1 to Many)

An incident can be related to many traffic signals (Many to Many)

A specific location can have many routes

A maintenance can be done from any specific location



Tanyaradzwa Mandizvidza

Business Rules:

Each route can have multiple vehicles using it.

Vehicles travel on specific routes.

Pedestrians use sidewalks on specific routes.

Routes have designated paths for pedestrians and cyclists.

Traffic signals control the flow of vehicles on certain routes.

Each route can have multiple traffic signals.

Sensors are placed along specific routes.

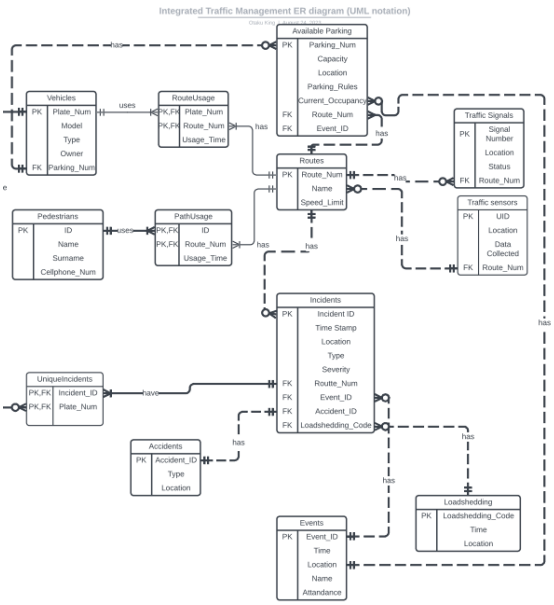
Sensors collect data about traffic conditions on those routes.

Each entry in the RouteUsage entity corresponds to a specific route.

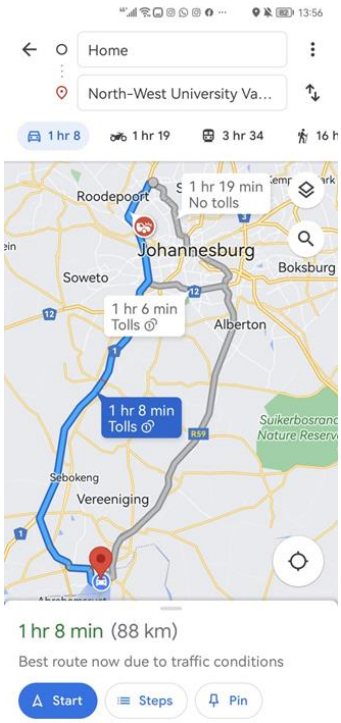
The PathUsage entity's Route name references the Route entity.

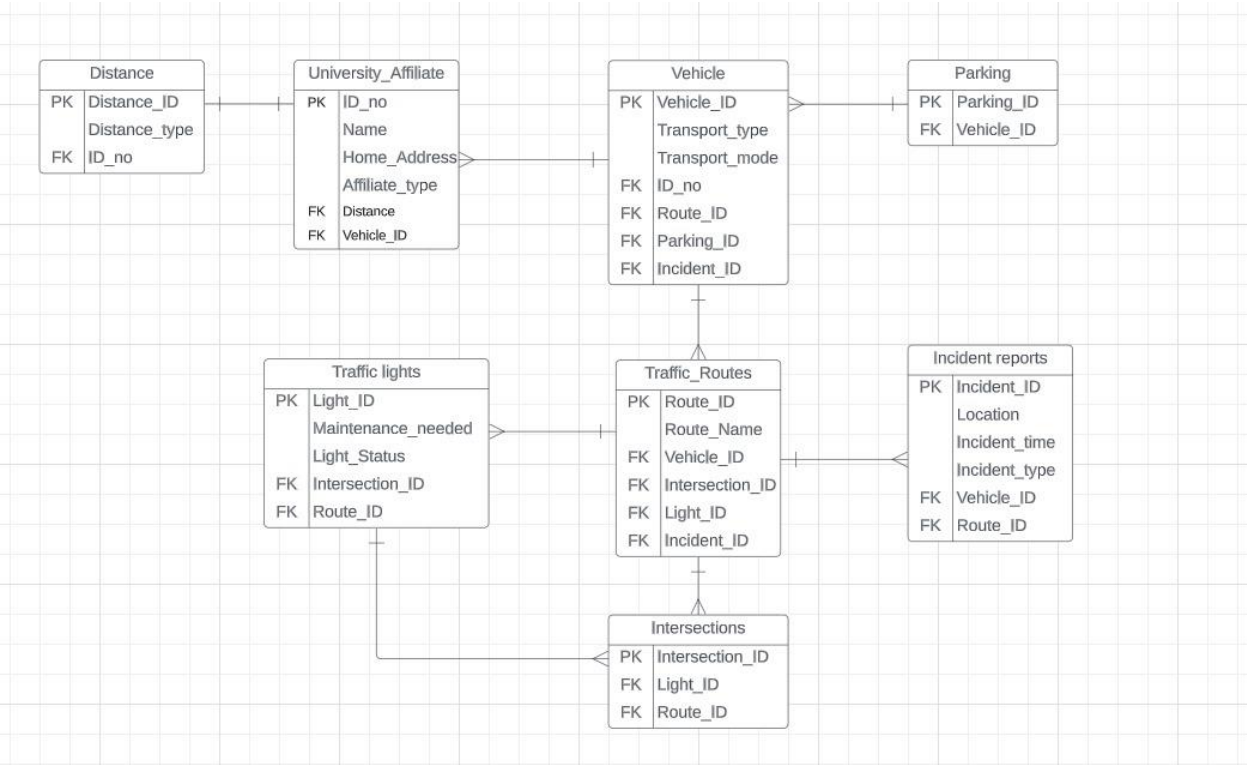
Each Pedestrian can have multiple entries in the PathUsage entity, representing their interactions with different routes.

The PathUsage entity's PedestrianID references the Pedestrian entity.

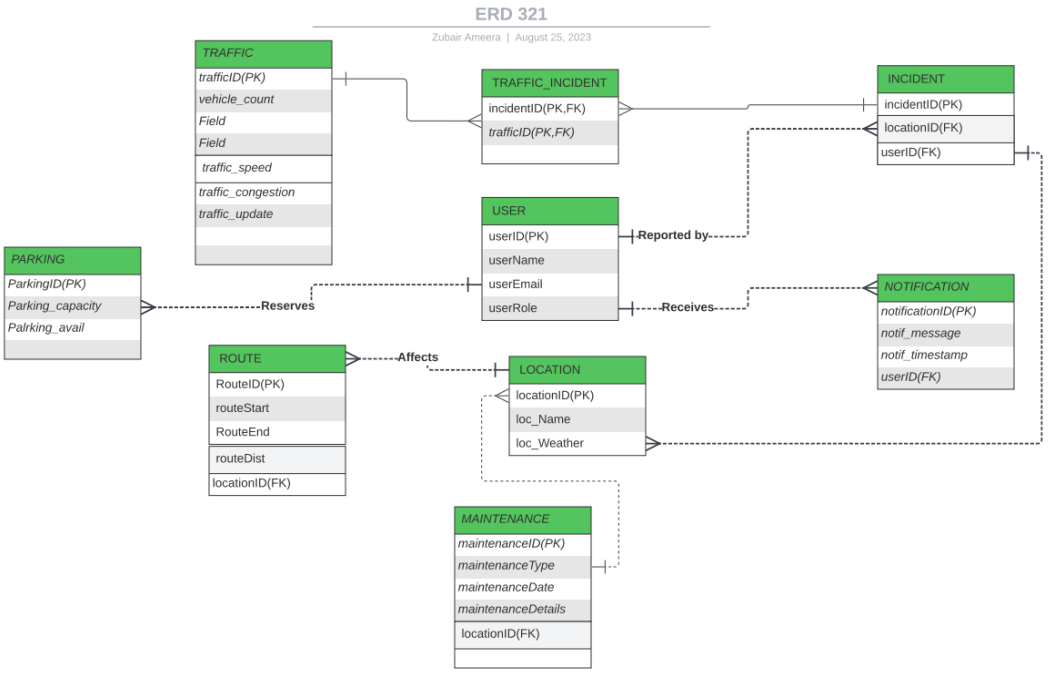


Kondwani Chindongo




Distance
PK Distance_ID
Distance_type
FK ID_no
University_Affiliate
PK ID_no
Name
Home_Address
Affiliate_type
FK Distance
FK Vehicle_ID
Vehicle
PK Vehicle_ID
Transport_type
Transport_mode
FK ID_no
FK Route_ID
FK Parking_ID
FK Incident_ID
Parking
PK Parking_ID
FK Vehicle_ID
Traffic lights
PK Light_ID
Maintenance_needed
Light_Status
FK Intersection_ID
FK Route_ID
Traffic_Routes
PK Route_ID
Route_Name
FK Vehicle_ID
FK Intersection_ID
FK Light_ID
FK Incident_ID
Incident reports
PK Incident_ID
Location
Incident_time
Incident_type
FK Vehicle_ID
FK Route_ID
Intersections
PK Intersection_ID
FK Light_ID
FK Route_ID

THE GROUP ERD (CHOSEN ERD)



NO ASSUMPTIONS MADE.

Phase 2 - Logical Schema , Additional requirements and SQL plan.

Phase 2 – Logical Schema, Additional requirements, and SQL plan

Mapping Strong Entities

1. **VEHICLE** (**VehicleID**, VehicleType)
    - PRIMARY KEY: VehicleID
2. **PARKING** (**ParkingID**, Location, Capacity, Availability)
    - PRIMARY KEY: ParkingID
3. **TRAFFIC** (**TrafficID**, TrafficDensity, SpeedLimit, CongestionLevel, TrafficStatus, Timestamp)
    - PRIMARY KEY: TrafficID
4. **ROUTES** (**RouteID**, StartLocation, EndLocation, RouteDescription, RouteOption)
    - PRIMARY KEY: RouteID
5. **IMAGES (ImageID,** ImageData, ImageDescription, TimeStamp)
    - PRIMARY KEY: ImageID

Mapping supertype/subtype relationships and weak entities

1. **VEHICLE_PARKING** (**VehicleID, ParkingID**, EntryTimestamp, ExitTimestamp)
    - Primary Key: (VehicleID, ParkingID)
    - Foreign Key: (VehicleID) REFERENCES VEHICLE
    - Foreign Key: (ParkingID) REFERENCES PARKING
2. **INCIDENT_TRAFFIC_DATA** (**IncidentID, TrafficID**, Location)
    - Primary Key: (IncidentID, TrafficID)
    - Foreign Key: (IncidentID) REFERENCES INCIDENTS
    - Foreign Key: (TrafficID) REFERENCES TRAFFIC
3. **VEHICLE_ROUTES** (**VehicleID, RouteID**)
    - Primary Key: (VehicleID, RouteID)
    - Foreign Key: (VehicleID) REFERENCES VEHICLE
    - Foreign Key: (RouteID) REFERENCES ROUTES

4. **ROUTE_SAT_MAPPING(RouteID, SatelliteViewID)**
    - Primary Key: (RouteID, SatelliteViewID)
    - Foreign Key: (RouteID) REFERENCES ROUTES
    - Foreign Key: (SatelliteViewID) REFERENCES SATELLITE_VIEW

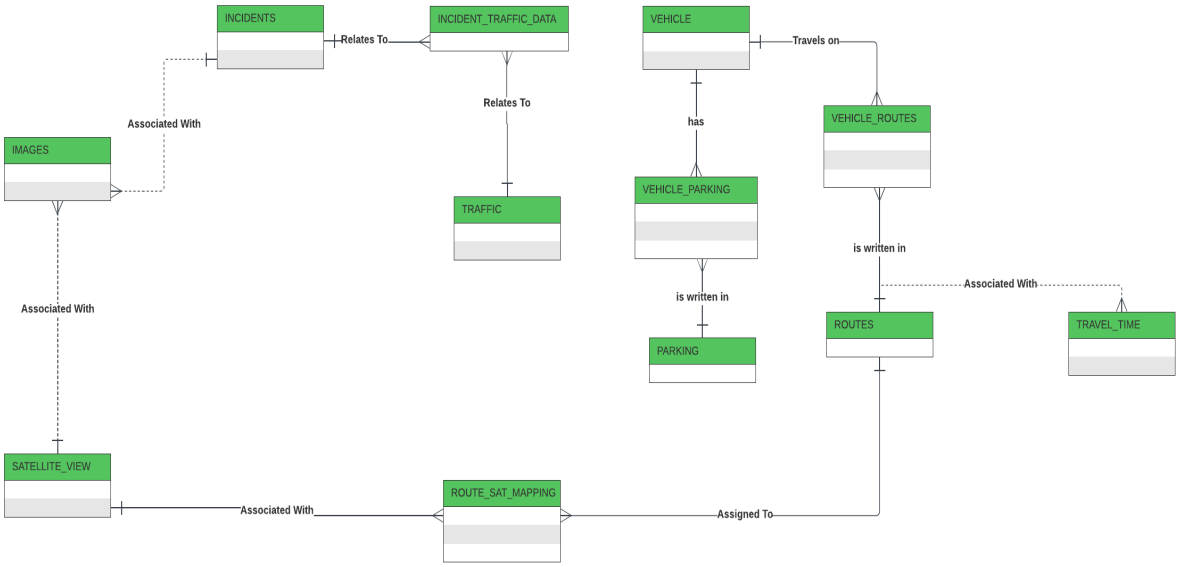
Mapping the binary relationships

1. **TRAVEL_TIME** (**EstimateID,** EstimateTime, TimeStamp, RouteID)
   PRIMARY KEY: EstimateID
   FOREIGN KEY: RouteID REFERENCES ROUTES

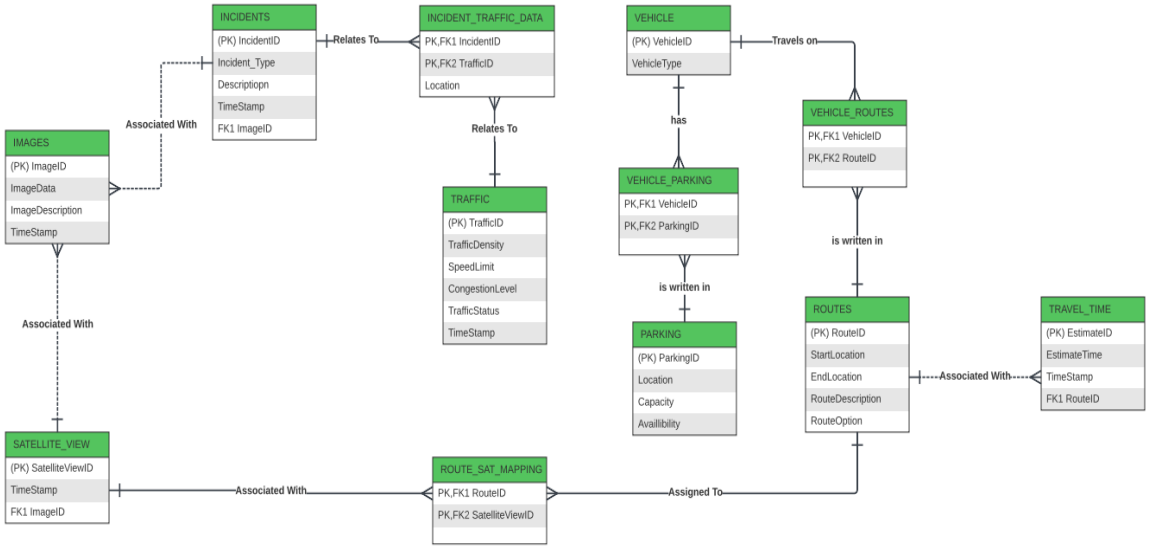
2. **INCIDENTS** (**IncidentID,** Incident_Type, Description, TimeStamp, ImageID)
   PRIMARY KEY: IncidentID
   FOREIGN KEY: ImageID REFERENCES IMAGES

3. **SATELLITE_VIEW** (**SatelliteViewID,** TimeStamp, ImageID)
   PRIMARY KEY : SatelliteViewID
   FOREIGN KEY: ImageID REFERENCES IMAGES

Initial Diagram



Logical Diagram



Normalization

Table name: Image

Image(ImageID,ImageData,ImageDescription,TimeStamp)

Table name: Incidents

Incidents(IncidentID,Incident_Type,Description,TimeStamp,ImageID)

Table name: Incident_Traffic_Data

Incident_Traffic_Data(IncidentID,TrafficID,Location)

Table name: Traffic

Traffic(TrafficID,TrafficDensity,SpeedLimit,CongestionLevel,TrafficStatus,TimeStamp)

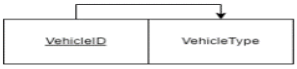


Table name: Vehicle

Vehicle(VehicleID,VehicleType)

Table name: Vehicle_Parking

Vehicle_Parking(VehicleID,ParkingID)

Table name: Parking

Parking(ParkingID,Location,Capacity,Availability)

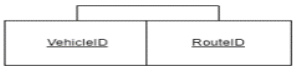


Table name: Vehicle_Routes

Vehicle_Routes(VehicleID,RouteID)

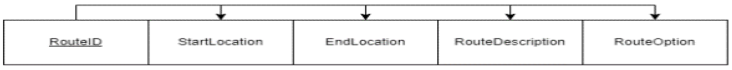


Table name: Routes

Routes(RouteID,StartLocation,EndLocation,RouteDescription,RouteOption)

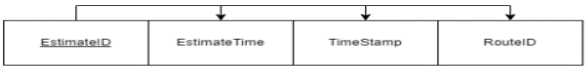


Table name: Travel_Time

Travel_Time(EstimateID,EstimateTime,TimeStamp,RouteID)

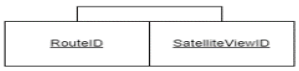


Table name: Route_Sat_Mapping

Route_Sat_Mapping(RouteID,SatelliteViewID)

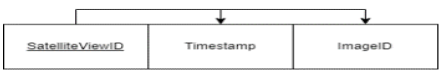
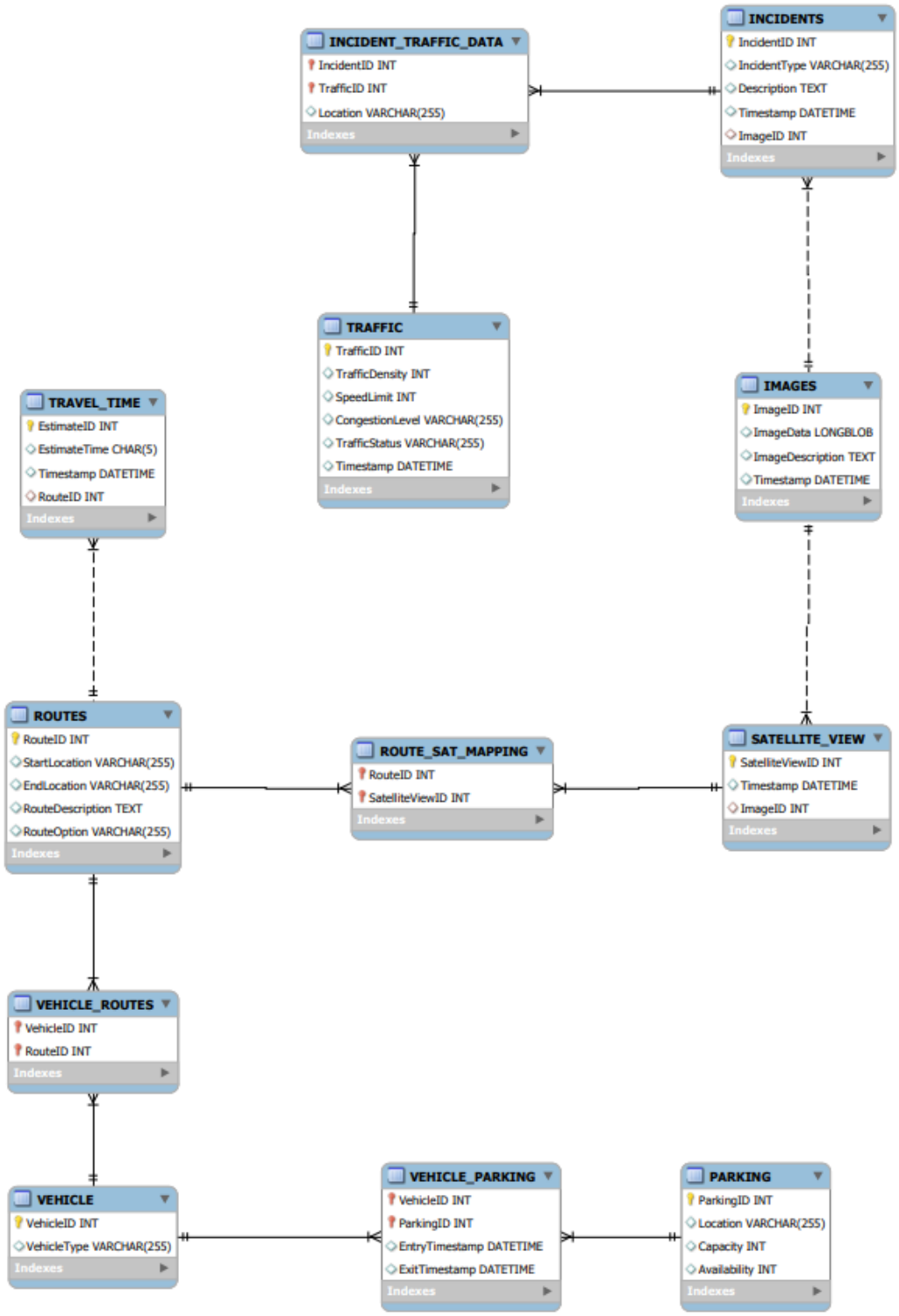


Table name: SatelliteView

SatelliteView(SatelliteViewID,Timestamp,ImageID)



The Above is the Crows Foot Notation ERD made in MS SQL Server

Documentation

Entity Name	Description	Role and Purpose	Relationships
Parking	Represents parking locations within the system.	Used to store information about different parking facilities, including their unique IDs, locations, capacity, and availability.	ONE TO MANY relationship with composite entity VEHICLE_PARKING
Incidents	Records various types of incidents on roadways.	This entity is essential for logging incidents such as accidents, construction, or road closures, and includes attributes like IncidentID, IncidentType, Description, and TimeStamp.	ONE TO MANY relationship with entity IMAGES and composite entity INCIDENT_TRAFFIC_DATA
Traffic	Stores data about traffic conditions.	Contains information about traffic density, speed limits, congestion levels, and overall traffic status. Helps monitor	ONE TO MANY relationship with composite entity INCIDENT_TRAFFIC_DATA

		and manage traffic flow.	
Vehicle	Contains information about vehicles.	Stores data about vehicles, including their unique IDs (VehicleID) and vehicle types. Used for traffic management and tracking.	ONE TO MANY relationship with 2 composite entities VEHICLE_ROUTES and VEHICLE_PARKING
SatelliteView	Stores satellite view data.	This entity holds satellite view data and its timestamps. It's crucial for visualizing traffic conditions and events on maps.	ONE TO MANY relationship with entity IMAGES and composite entity ROUTE_SAT_MAPPING
Routes	Contains data about predefined routes.	Helps define specific routes with details such as StartLocation, EndLocation, RouteDescription, and RouteOption. Useful for navigation and route planning.	ONE TO MANY relationship with 2 composite entities VEHICLE_ROUTES and ROUTE_SAT_MAPPING

TravelTime	Records estimated travel times.	Contains data about travel time estimates with attributes like EstimateID, EstimateTime, and TimeStamp. Useful for providing travel time information to users.	MANY TO ONE relationship with ROUTES entity
Image	Stores image data and descriptions.	Used for storing images related to traffic, incidents, or other relevant data. Includes attributes like ImageID, ImageData, ImageDescription, and TimeStamp.	MANT TO ONE relationship with entities SATELLITE_VIEW and INCIDENTS
VEHICLE_ROUTES	Composite entity that connects vehicles to routes.	Serves as a link between vehicles and predefined routes, allowing tracking of vehicle movements and usage of specific routes.	MANY TO ONE relationship between entities ROUTES AND VEHICLE
VEHICLE_PARKING	Composite entity linking vehicles to	Records information about which vehicles are	MANY TO ONE relationship between

	parking locations.	parked at various parking facilities, aiding in monitoring parking usage.	entities PARKING AND VEHICLE.
ROUTE_SAT_MAPPING	Composite entity connecting routes to satellite views.	Defines associations between predefined routes and corresponding satellite views, enabling visual route planning.	MANY TO ONE relationship between entities ROUTES AND SATELLITE_VIEW
INCIDENT_TRAFFIC_DATA	Composite entity linking incidents to traffic data.	Provides a connection between recorded incidents and the resulting traffic data, allowing the assessment of traffic effects due to incidents.	MANY TO ONE relationship between entities INCIDENTS AND TRAFFIC

Validating the logical model integrity constraints

VEHICLE Entity

1. **VehicleID:**
    - **Domain**: Numeric
    - **Description**: This attribute represents a unique identifier for each vehicle in the system.
    - **Constraints**:
        - Type: Numeric
        - Range: Low value = 1000, High value = 9999

- Display format: 9999
- Length: 4

2. **Vehicle Type:**
    - **Domain**: Character
    - **Description**: This attribute specifies the type or category of the vehicle (e.g., car, truck, motorcycle).
    - **Constraints**:
        - Type: Character
        - Length: Variable (based on the vehicle type)

PARKING Entity

1. **ParkingID:**
    - **Domain**: Numeric
    - **Description**: This attribute represents a unique identifier for each parking location in the system.
    - **Constraints**:
        - Type: Numeric
        - Range: Low value = 1000, High value = 9999
        - Display format: 9999
        - Length: 4
2. **Location:**
    - **Domain**: Character
    - **Description**: This attribute specifies the physical location or address of the parking facility.
    - **Constraints**:
        - Type: Character
        - Length: Variable (based on the location's address)
3. **Capacity:**
    - **Domain**: Numeric
    - **Description**: This attribute indicates the total parking capacity of the location, i.e., the maximum number of vehicles it can accommodate.
    - **Constraints**:
        - Type: Numeric
        - Range: Positive integers (e.g., 1, 2, 100, etc.)
        - Length: Variable (based on the specific capacity)

4. **Availability:**
    - **Domain**: Numeric
    - **Description**: This attribute represents the current availability or the number of parking spaces that are vacant at the location.
    - **Constraints**:
        - Type: Numeric
        - Range: Non-negative integers (e.g., 0, 1, 2, etc.)

- Length: Variable (based on the specific availability)

ROUTES Entity

1. **RouteID:**
    - **Domain**: Numeric
    - **Description**: This attribute represents a unique identifier for each route in the system.
    - **Constraints**:
        - Type: Numeric
        - Range: Low value = 1000, High value = 9999
        - Display format: 9999
        - Length: 4
2. **StartLocation:**
    - **Domain**: Character
    - **Description**: This attribute specifies the starting point or location of the route.
    - **Constraints**:
        - Type: Character
        - Length: Variable (based on the specific location)
3. **EndLocation:**
    - **Domain**: Character
    - **Description**: This attribute specifies the ending point or location of the route.
    - **Constraints**:
        - Type: Character
        - Length: Variable (based on the specific location)
4. **RouteDescription:**
    - **Domain**: Character
    - **Description**: This attribute provides a textual description or additional information about the route.
    - **Constraints**:
        - Type: Character
        - Length: Variable (based on the description)
5. **RouteOption:**
    - **Domain**: Character
    - **Description**: This attribute indicates a route option or type (e.g., scenic, fastest, shortest, etc.).
    - **Constraints**:
        - Type: Character
        - Length: Variable (based on the specific route option)

TRAVEL_TIME Entity

1. **EstimateID:**
    - **Domain**: Numeric

- **Description**: This attribute represents a unique identifier for each travel time estimate in the system.
- **Constraints**:
    - Type: Numeric
    - Range: Low value = 1000, High value = 9999
    - Display format: 9999
    - Length: 4

2. **EstimateTime:**
    - **Domain**: Character
    - **Description**: This attribute specifies the estimated travel time for a route, typically in a time format.
    - **Constraints**:
        - Type: Character
        - Display format: 99:99 (24-hour clock)
        - Display range: 00:00 to 23:59
        - Length: 5
3. **TimeStamp:**
    - **Domain**: Date and Time
    - **Description**: This attribute represents the timestamp or date and time when the travel time estimate was recorded.
    - **Constraints**:
        - Type: Date and Time
        - Format: YYYY-MM-DD HH:MM:SS (or an appropriate date and time format)
        - Range: Based on system requirements

TRAFFIC Entity

1. **TrafficID:**
    - **Domain**: Numeric
    - **Description**: This attribute represents a unique identifier for each traffic data entry in the system.
    - **Constraints**:
        - Type: Numeric
        - Range: Low value = 1000, High value = 9999
        - Display format: 9999
        - Length: 4
2. **TrafficDensity:**
    - **Domain**: Numeric
    - **Description**: This attribute indicates the traffic density or the number of vehicles on a specific road segment.
    - **Constraints**:
        - Type: Numeric
        - Range: Positive integers (e.g., 1, 2, 100, etc.)

- Length: Variable based on specific data requirements

3. **SpeedLimit:**
    - **Domain**: Numeric
    - **Description**: This attribute specifies the speed limit for the road segment in the traffic data.
    - **Constraints**:
        - Type: Numeric
        - Range: Positive integers (e.g., 10, 20, 60, etc.)
        - Length: Variable based on specific speed limits
4. **CongestionLevel:**
    - **Domain**: Character
    - **Description**: This attribute represents the congestion level on the road segment (e.g., low, moderate, high).
    - **Constraints**:
        - Type: Character
        - Length: Variable based on specific congestion levels
5. **TrafficStatus:**
    - **Domain**: Character
    - **Description**: This attribute indicates the overall traffic status (e.g., normal, heavy, accident, etc.).
    - **Constraints**:
        - Type: Character
        - Length: Variable based on specific traffic status descriptions
6. **TimeStamp:**
    - **Domain**: Date and Time
    - **Description**: This attribute represents the timestamp or date and time when the traffic data was recorded.
    - **Constraints**:
        - Type: Date and Time
        - Format: YYYY-MM-DD HH:MM:SS (or an appropriate date and time format)
        - Range: Based on system requirements

INCIDENTS Entity

1. **IncidentID:**
    - **Domain**: Numeric
    - **Description**: This attribute represents a unique identifier for each incident record in the system.
    - **Constraints**:
        - Type: Numeric
        - Range: Low value = 1000, High value = 9999
        - Display format: 9999
        - Length: 4

2. **IncidentType:**
    - **Domain**: Character
    - **Description**: This attribute specifies the type or category of the incident (e.g., accident, construction, road closure).
    - **Constraints**:
        - Type: Character
        - Length: Variable based on specific incident types
3. **Description:**
    - **Domain**: Character
    - **Description**: This attribute provides a textual description or additional information about the incident.
    - **Constraints**:
        - Type: Character
        - Length: Variable based on the specific description
4. **TimeStamp:**
    - **Domain**: Date and Time
    - **Description**: This attribute represents the timestamp or date and time when the incident data was recorded.
    - **Constraints**:
        - Type: Date and Time
        - Format: YYYY-MM-DD HH:MM:SS (or an appropriate date and time format)
        - Range: Based on system requirements

IMAGES Entity

1. **ImageID:**
    - **Domain**: Numeric
    - **Description**: This attribute represents a unique identifier for each image in the system.
    - **Constraints**:
        - Type: Numeric
        - Range: Low value = 1000, High value = 9999
        - Display format: 9999
        - Length: 4
2. **ImageData:**
    - **Domain**: Binary or Character
    - **Description**: This attribute contains the actual image data, which can be stored as binary data or as a character (text) representation of the image file location or reference.
    - **Constraints**:
        - Type: Binary (if storing the image data as binary) or Character (if storing as a file path or URL)
        - Length: Variable based on the specific storage method
3. **ImageDescription:**
    - **Domain**: Character

- **Description**: This attribute provides a textual description or metadata related to the image.
- **Constraints**:
    - Type: Character
    - Length: Variable based on the specific image descriptions

4. **TimeStamp:**
    - **Domain**: Date and Time
    - **Description**: This attribute represents the timestamp or date and time when the image data was recorded or added to the system.
    - **Constraints**:
        - Type: Date and Time
        - Format: YYYY-MM-DD HH:MM:SS (or an appropriate date and time format)
        - Range: Based on system requirements

SATELLITE_VIEW Entity

1. **SatelliteViewID:**
    - **Domain**: Numeric
    - **Description**: This attribute represents a unique identifier for each satellite view in the system.
    - **Constraints**:
        - Type: Numeric
        - Range: Low value = 1000, High value = 9999
        - Display format: 9999
        - Length: 4
2. **TimeStamp:**
    - **Domain**: Date and Time
    - **Description**: This attribute represents the timestamp or date and time when the satellite view data was recorded or added to the system.
    - **Constraints**:
        - Type: Date and Time
        - Format: YYYY-MM-DD HH:MM:SS (or an appropriate date and time format)
        - Range: Based on system requirement

SQL Statements for logical plan

```
-- Create the VEHICLE table (Represents information about vehicles)
CREATE TABLE VEHICLE (
   VehicleID INT UNSIGNED NOT NULL,    -- Unique vehicle identifier
   VehicleType VARCHAR(255),         -- Type or category of the vehicle (e.g., car, truck,
motorcycle)
   PRIMARY KEY (VehicleID)
);
```

```
-- Create the PARKING table (Represents information about parking locations)
CREATE TABLE PARKING (
   ParkingID INT UNSIGNED NOT NULL,   -- Unique parking location identifier
   Location VARCHAR(255),          -- Physical location or address of the parking facility
   Capacity INT UNSIGNED,           -- Total parking capacity of the location
   Availability INT UNSIGNED,       -- Current availability or number of vacant parking spaces
   PRIMARY KEY (ParkingID)
);

-- Create the TRAFFIC table (Stores traffic data)
CREATE TABLE TRAFFIC (
   TrafficID INT UNSIGNED NOT NULL,   -- Unique traffic data entry identifier
   TrafficDensity INT UNSIGNED,      -- Traffic density or the number of vehicles on a road segment
   SpeedLimit INT UNSIGNED,         -- Speed limit for the road segment
   CongestionLevel VARCHAR(255),     -- Congestion level on the road segment
   TrafficStatus VARCHAR(255),       -- Overall traffic status (e.g., normal, heavy, accident)
   Timestamp DATETIME,             -- Timestamp of when the traffic data was recorded
   PRIMARY KEY (TrafficID)
);

-- Create the ROUTES table (Represents route information)
CREATE TABLE ROUTES (
   RouteID INT UNSIGNED NOT NULL,    -- Unique route identifier
   StartLocation VARCHAR(255),       -- Starting point or location of the route
   EndLocation VARCHAR(255),         -- Ending point or location of the route
   RouteDescription TEXT,           -- Additional textual description of the route
   RouteOption VARCHAR(255),         -- Route option or type (e.g., scenic, fastest, shortest)
   PRIMARY KEY (RouteID)
);

-- Create the IMAGES table (Stores image data)
CREATE TABLE IMAGES (
   ImageID INT UNSIGNED NOT NULL,    -- Unique image identifier
   ImageData LONGBLOB,              -- Binary image data
   ImageDescription TEXT,           -- Description or metadata related to the image
   Timestamp DATETIME,              -- Timestamp of when the image data was recorded
   PRIMARY KEY (ImageID)
);

-- Create the INCIDENTS table (Stores incident information)
CREATE TABLE INCIDENTS (
   IncidentID INT UNSIGNED NOT NULL, -- Unique incident record identifier
   IncidentType VARCHAR(255),       -- Type or category of the incident (e.g., accident,
construction)
```

```
    Description TEXT,              -- Description or additional information about the incident
    Timestamp DATETIME,            -- Timestamp of when the incident data was recorded
    ImageID INT UNSIGNED,          -- Foreign key referencing the IMAGES table for incident images
    PRIMARY KEY (IncidentID),
    FOREIGN KEY (ImageID) REFERENCES IMAGES(ImageID)
);

-- Create the SATELLITE_VIEW table (Stores satellite view data)
CREATE TABLE SATELLITE_VIEW (
    SatelliteViewID INT UNSIGNED NOT NULL, -- Unique satellite view identifier
    Timestamp DATETIME,            -- Timestamp of when the satellite view data was recorded
    ImageID INT UNSIGNED,          -- Foreign key referencing the IMAGES table for satellite view
images
    PRIMARY KEY (SatelliteViewID),
    FOREIGN KEY (ImageID) REFERENCES IMAGES(ImageID)
);

-- Create the VEHICLE_PARKING table (Associates vehicles with parking locations)
CREATE TABLE VEHICLE_PARKING (
    VehicleID INT UNSIGNED,        -- Foreign key referencing the VEHICLE table for the vehicle
    ParkingID INT UNSIGNED,        -- Foreign key referencing the PARKING table for the parking
location
    EntryTimestamp DATETIME,       -- Timestamp of vehicle entry into the parking location
    ExitTimestamp DATETIME,        -- Timestamp of vehicle exit from the parking location
    PRIMARY KEY (VehicleID, ParkingID),
    FOREIGN KEY (VehicleID) REFERENCES VEHICLE(VehicleID),
    FOREIGN KEY (ParkingID) REFERENCES PARKING(ParkingID)
);

-- Create the INCIDENT_TRAFFIC_DATA table (Associates incidents with traffic data)
CREATE TABLE INCIDENT_TRAFFIC_DATA (
    IncidentID INT UNSIGNED,       -- Foreign key referencing the INCIDENTS table for the incident
    TrafficID INT UNSIGNED,        -- Foreign key referencing the TRAFFIC table for the traffic data
    Location VARCHAR(255),         -- Location information related to the incident
    PRIMARY KEY (IncidentID, TrafficID),
    FOREIGN KEY (IncidentID) REFERENCES INCIDENTS(IncidentID),
    FOREIGN KEY (TrafficID) REFERENCES TRAFFIC(TrafficID)
);

-- Create the VEHICLE_ROUTES table (Associates vehicles with routes)
CREATE TABLE VEHICLE_ROUTES (
    VehicleID INT UNSIGNED,        -- Foreign key referencing the VEHICLE table for the vehicle
    RouteID INT UNSIGNED,          -- Foreign key referencing the ROUTES table for the route
    PRIMARY KEY (VehicleID, RouteID),
```

```
    FOREIGN KEY (VehicleID) REFERENCES VEHICLE(VehicleID),
    FOREIGN KEY (RouteID) REFERENCES ROUTES(RouteID)
);

-- Create the ROUTE_SAT_MAPPING table (Associates routes with satellite views)
CREATE TABLE ROUTE_SAT_MAPPING (
    RouteID INT UNSIGNED,          -- Foreign key referencing the ROUTES table for the route
    SatelliteViewID INT UNSIGNED,   -- Foreign key referencing the SATELLITE_VIEW table for the
satellite view
    PRIMARY KEY (RouteID, SatelliteViewID),
    FOREIGN KEY (RouteID) REFERENCES ROUTES(RouteID),
    FOREIGN KEY (SatelliteViewID) REFERENCES SATELLITE_VIEW(SatelliteViewID)
);

-- Create the TRAVEL_TIME table (Stores travel time estimates)
CREATE TABLE TRAVEL_TIME (
    EstimateID INT UNSIGNED NOT NULL, -- Unique travel time estimate identifier
    EstimateTime CHAR(5),           -- Estimated travel time in HH:MM format
    Timestamp DATETIME,             -- Timestamp of when the estimate was recorded
    RouteID INT UNSIGNED,           -- Foreign key referencing the ROUTES table for the route
    PRIMARY KEY (EstimateID),
    FOREIGN KEY (RouteID) REFERENCES ROUTES(RouteID)
);
```