DTSA 5733 Relational Database Design
Final Project

Step 1: Business Story

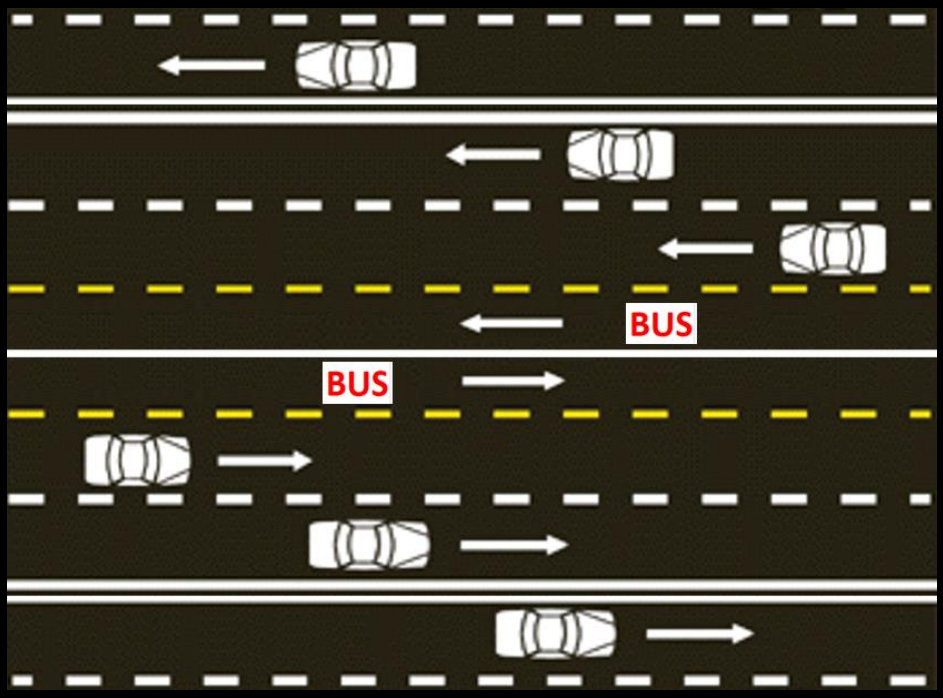
The first step is the describe the background story of our case here, such that I chose a project that is related to my BSc background which is Civil Engineering. I hope this idea reflects effectively my learning outcomes to the Peer Evaluation , and I hope also this final project be useful to my career as a Data Scientist with a Civil Engineering background.

In this final project I am going to design a Relational Database for a Traffic Management system that is connecting to IoT sensors that collect information about the current status of the highway, such that I am going to assume some entities and their attributes from my mind , then I am going to apply the next five steps to reach to my goal mentioned earlier in this paragraph.

Let’s briefly imagine the case we have by assuming the following points :

1- We have two main sides in opposite directions for all vehicles.
2- We have two special sides in opposite directions for Amman Bus Rapid Transit (Amman is the capital city of the Kingdom of Jordan).
3- Refer to below image that describes directions of the vehicles.

Finally, please note that I used Amman Bus Rapid Transit project just as title to deliver this business case of my interest, other than the project name all information provided are only based on my own assumptions, please refer to the link provided in the reference section for more reading about this real project.



Step 2: Building the Entity Relationship Model (ER Model)

Below are the entities and their attributes after several modifications to the original stated ones. Please refer to the first point in References section.

- Highway: HighwayID, HighwayName, HighwayLocation, HighwayLength, LanesNumbers, ConstructionDate, MaintenaceSchedule, AvgDailyTraffic, MaxAllowableSpeed. HighwayID is the identifier.

- Segment: SegmentID, StartLocation, EndLocation, Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions. SegmentID is the identifier.

- IoT_Device: DeviceID, Location, SensorType, InstallationDate, MaintenanceSchedule, Status, CommunicationProtocol. DeviceID is the identifier.

- Traffic_Camera: CameraID, Location, InstalledDirection, FieldView, Resolution, CameraStatus. CameraID is the identifier.

- Traffic_Flow_Sensor: SensorID, Location, SensorType, InstalledDirection, Accuracy, DataCollectionFrequency, SensorStatus. SensorID is the identifier.

- Weather_Sensor: SensorID, Location, SensorType, Accuracy, DataCollectionFrequency, SensorStatus. SensorID is the identifier.

- Maintenance: MaintenanceID, Location, StartTime, MaintenanceType, Description, ImpactOnTraffic. MaintenanceID is the identifier.

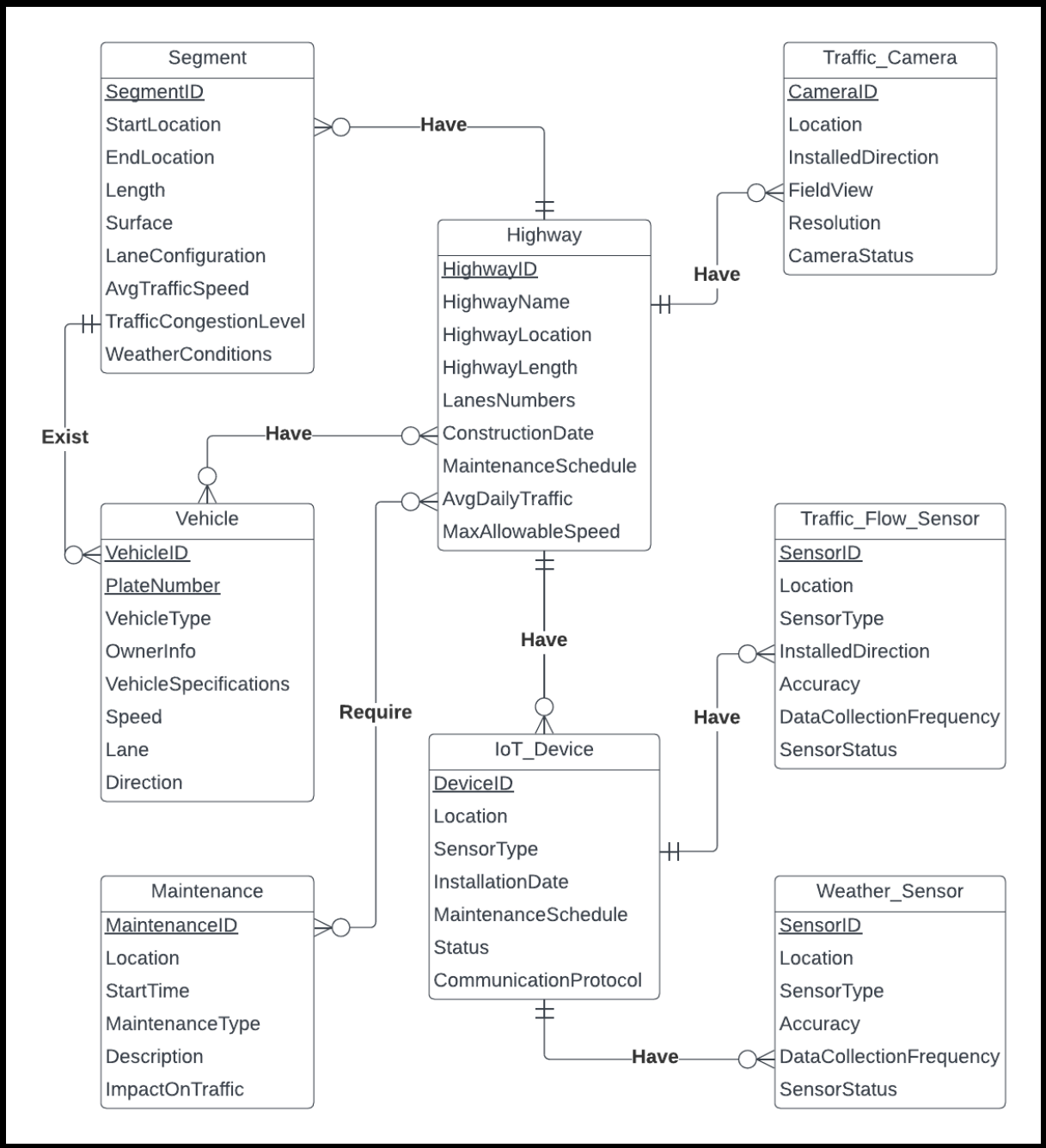
- Vehicle: VehicleID, PlateNumber, VehicleType, OwnerInfo, VehicleSpecifications, Speed, Lane, Direction. VehicleID, and PlateNumber are both identifiers

The relationships are as follows :

- A highway may have one or more segments. A segment must belong to one and only one highway.
- An IoT Device may have one or more sensors. A sensor must belong to one and only one IoT Device.
- A highway may have one or more IoT Devices. An IoT Device must belong to one and only one highway.
- A highway ma have one or more cameras. A camera must belong to one and only one highway.
- A highway may require one or more maintenance services. A maintenance may be implemented to one or more highways.

- A highway may have one or more (zero or more) vehicles. A vehicle may use one or more highways.
- A vehicle must exist in one and only one segment. A segment may contain one or more vehicles.

Step 3: Drawing the Entity Relationship Diagram (ERD)



Step 4: Converting the ERD into a Relational Model

- Highway (HighwayID, HighwayName, HighwayLocation, HighwayLength, LanesNumbers, ConstructionDate, MaintenaceSchedule, AvgDailyTraffic, MaxAllowableSpeed).

- Segment (SegmentID, StartLocation, EndLocation, Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions).

- IoT_Device (DeviceID, Location, SensorType, InstallationDate, MaintenanceSchedule, Status, CommunicationProtocol).

- Traffic_Camera (CameraID, Location, InstalledDirection, FieldView, Resolution, CameraStatus).

- Traffic_Flow_Sensor (SensorID, Location, SensorType, InstalledDirection, Accuracy, DataCollectionFrequency, SensorStatus).

- Weather_Sensor (SensorID(fk), Location, SensorType, Accuracy, DataCollectionFrequency, SensorStatus).

- Maintenance (MaintenanceID, Location, StartTime, MaintenanceType, Description, ImpactOnTraffic).

- Vehicle (VehicleID, PlateNumber, VehicleType, OwnerInfo, VehicleSpecifications, Speed, Lane, Direction).

Step 5: Normalization the Relational Model to the Third Normal Form (3NF)

Step 5 is implemented in two steps as follows, the first is by understanding and collecting Functional Dependencies based on the business understanding according to stakeholders of the project.

The second step is by eliminating partial dependencies ending up with 2NF, then eliminating transitive dependencies reaching to 3NF as required.

The normalization process is implemented as follows:

- Highway (HighwayID, HighwayName, HighwayLocation, HighwayLength, LanesNumbers, ConstructionDate, MaintenaceSchedule, AvgDailyTraffic, MaxAllowableSpeed).

  FD1: HighwayID → HighwayName, HighwayLocation, HighwayLength, LanesNumbers, ConstructionDate, MaintenaceSchedule, AvgDailyTraffic, MaxAllowableSpeed

- Segment (SegmentID, StartLocation, EndLocation, Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions).

  FD1: SegmentID → StartLocation, EndLocation, Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions

  FD2: Length → StartLocation, EndLocation

- IoT_Device (DeviceID, Location, SensorType, InstallationDate, MaintenanceSchedule, Status, CommunicationProtocol).

  FD1: DeviceID → Location, SensorType, InstallationDate, MaintenanceSchedule, Status, CommunicationProtocol

  FD2: Status → MaintenanceSchedule

- Traffic_Camera (CameraID, Location, InstalledDirection, FieldView, Resolution, CameraStatus).

  FD1: CameraID → Location, InstalledDirection, FieldView, Resolution, CameraStatus

- Traffic_Flow_Sensor (SensorID, Location, SensorType, InstalledDirection, Accuracy, DataCollectionFrequency, SensorStatus).

  FD1: SensorID → Location, SensorType, InstalledDirection, Accuracy, DataCollectionFrequency, SensorStatus

- Weather_Sensor (SensorID(fk), Location, SensorType, Accuracy, DataCollectionFrequency, SensorStatus).

  FD1: SensorID → Location, SensorType, Accuracy, DataCollectionFrequency, SensorStatus

- Maintenance (MaintenanceID, Location, StartTime, MaintenanceType, Description, ImpactOnTraffic).

  FD1: MaintenanceID → Location, StartTime, MaintenanceType, Description, ImpactOnTraffic

- Vehicle (VehicleID, PlateNumber, VehicleType, OwnerInfo, VehicleSpecifications, Speed, Lane, Direction).

  FD1: VehicleID, PlateNumber → VehicleType, OwnerInfo, VehicleSpecifications, Speed, Lane, Direction

The entities Highway, Traffic_Camera, Traffic_Flow_Sensor, Weather_Sensor, Maintenance, and Vehicle are all in 3NF, because they are in 1NF; they have no partial functional dependencies so they are in 2NF; and they have no transitive functional dependencies so they are in 3NF.

The entities Segment, and IoT_Device are in 2NF, because they are in 1NF, they have no partial functional dependencies so they are in 2NF, but they are not in 3NF, because FD2 in both entities is a transitive functional dependency, such that we need to normalize these two entities, so let's start with normalization steps for Segment entity as follows :

- Segment (SegmentID, StartLocation, EndLocation, Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions).

  FD1: SegmentID → StartLocation, EndLocation, Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions

  FD2: Length → StartLocation, EndLocation

  FD2 is a transitive functional dependency the we need to eliminate, so we need to create new relations to achieve that.

- Segment (SegmentID, Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions)

  FD1: SegmentID → Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions

- Segment_Length (Length, StartLocation, EndLocation)

  FD1: Length → StartLocation, EndLocation

  Both new relations are now normalized to 3NF, now we will implement the same normalization steps for the IoT_Device entity as follows:

- IoT_Device (DeviceID, Location, SensorType, InstallationDate, MaintenanceSchedule, Status, CommunicationProtocol).

  FD1: DeviceID → Location, SensorType, InstallationDate, MaintenanceSchedule, Status, CommunicationProtocol

  FD2: Status → MaintenanceSchedule

- IoT_Device (DeviceID, Location, SensorType, InstallationDate, Status, CommunicationProtocol)

  FD1: DeviceID → Location, SensorType, InstallationDate, Status, CommunicationProtocol

- IoT_Device_Status (Status, MaintenanceSchedule)

  FD1: Status → MaintenanceSchedule

Both new relations are now normalized to 3NF.

Step 6: Final Output of the Relational Model in 3NF

- Highway (HighwayID, HighwayName, HighwayLocation, HighwayLength, LanesNumbers, ConstructionDate, MaintenaceSchedule, AvgDailyTraffic, MaxAllowableSpeed).

  FD1: HighwayID → HighwayName, HighwayLocation, HighwayLength, LanesNumbers, ConstructionDate, MaintenaceSchedule, AvgDailyTraffic, MaxAllowableSpeed

- Segment (SegmentID, Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions)

  FD1: SeqgmentID → Length, Surafce, LaneConfiguration, AvgTrafficSpeed, TrafficCongestionLevel, WeatherConditions

- Segment_Length (Length, StartLocation, EndLocation)

  FD1: Length → StartLocation, EndLocation

- IoT_Device (DeviceID, Location, SensorType, InstallationDate, Status, CommunicationProtocol)

  FD1: DeviceID → Location, SensorType, InstallationDate, Status, CommunicationProtocol

- IoT_Device_Status (Status, MaintenanceSchedule)

  FD1: Status → MaintenanceSchedule

- Traffic_Camera (CameraID, Location, InstalledDirection, FieldView, Resolution, CameraStatus).

  FD1: CameraID → Location, InstalledDirection, FieldView, Resolution, CameraStatus

- Traffic_Flow_Sensor (SensorID, Location, SensorType, InstalledDirection, Accuracy, DataCollectionFrequency, SensorStatus).

  FD1: SensorID → Location, SensorType, InstalledDirection, Accuracy, DataCollectionFrequency, SensorStatus

- Weather_Sensor (SensorID(fk), Location, SensorType, Accuracy, DataCollectionFrequency, SensorStatus).

  FD1: SensorID → Location, SensorType, Accuracy, DataCollectionFrequency, SensorStatus

- Maintenance (MaintenanceID, Location, StartTime, MaintenanceType, Description, ImpactOnTraffic).

  FD1: MaintenanceID → Location, StartTime, MaintenanceType, Description, ImpactOnTraffic

- Vehicle (VehicleID, PlateNumber, VehicleType, OwnerInfo, VehicleSpecifications, Speed, Lane, Direction).

  FD1: VehicleID, PlateNumber → VehicleType, OwnerInfo, VehicleSpecifications, Speed, Lane, Direction

Reference:

1- The following link shows how initiated the assumed entities and attributes using ChatGPT before I make my modifications https://poe.com/s/nADI9BNc83KUOs5heIPQ

2- Amman Bus Rapid Transit – Wikipedia https://en.wikipedia.org/wiki/Amman_Bus_Rapid_Transit

3- Tool used for drawing ERD is Lucid Chart www.lucidchart.com