Road Safety Analysis System

Deliverable 3

Group 25

Group members:

Kavya Gopal (6581-2209)

Kasiviswanathan Srikant Iyer (5222-2519)

Rema Veeranna Gowda (9462-8005)

Swaathi Reena Velavan (1230-8520)

Contents

Accident Analysis System

Introduction

We create a table Accident which includes the place of accident, time and date of it. Along with it, factors which caused accidents and losses incurred are recorded. We use the count of the accidents in specific areas, time and different attributes to plot the various graphs. It forms the main table connecting to every other table. How many years ago is a derived attribute which is got from the system time and the date of accident.

Insurance holds the policies which the drivers who were a part of the accident have taken. While comparing the different vehicles to buy, the users make use of it to compare different sets of parameters and make an informed decision.

Accidents lead to legal cases and they are used by the law enforcement agencies to make an estimation of where the accidents happen and anything they could come up with as a solution.

Vehicle type and different factors which might have caused are analysed and plotted for the users to make a decision of buying based on the information acquired.

Transformation of ER diagram into a collection of Relational Schemas

Diagram

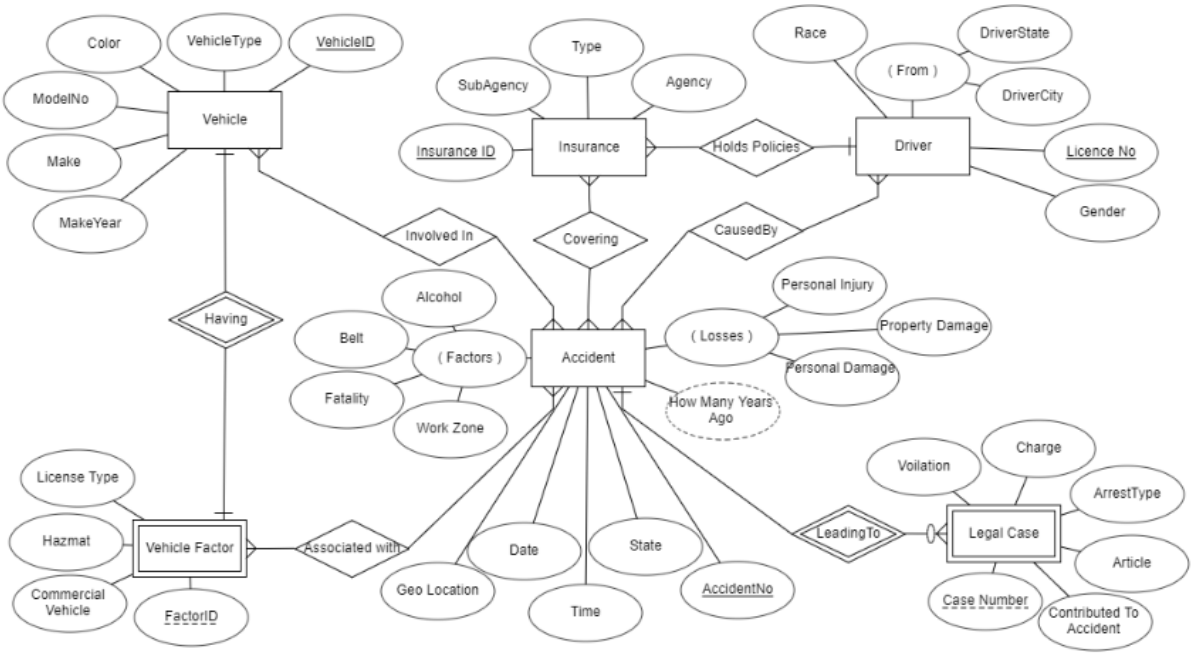

Figure 1: ER diagram

Relationship Schema

Vehicle (Vehicle_ID : varchar, vehicle_type : varchar, color: varchar, model_no: varchar, make: varchar, MakeYear: int)

The vehicle table has the following attributes:
Vehicle_ID is the primary key used to uniquely identify a tuple in this table.
Vehicle_type is an attribute that describes what type of vehicle we are concerned with.
Color is an attribute that describes the color of the concerned vehicle.

Model_no is an attribute that describes the model number of the concerned vehicle.
Make is an attribute that describes the make of the concerned vehicle.
MakeYear is an attribute that describes the make-year of the concerned vehicle.

Vehicle_Factor (FactorID: int, commercial vehicle:varchar, Hazmat: varchar, LicenseType: varchar, Vehicle_ID : varchar)

Since Vehicle_Factor is a **weak entity** type, a relation (table) is created that includes all simple attributes of the weak entity(Vehicle_Factor). The primary key of the new relation should be the partial key of the weak entity plus the primary key of its owner (Vehicle table).

The **Vehicle_Factor** table has the following attributes -
The ***Factor_ID*** along with Vehicle_ID together form the primary key used to uniquely identify a tuple in this table.
commercial_vehicle is an attribute that describes what kind of vehicle we are concerned with.
Hazmat is an attribute that specifies if hazmat was present or not in the concerned vehicle.
LicenseType is an attribute that describes the license type of the concerned vehicle.
Vehicle_ID is primary key of the identifying entity Vehicle. Thus, the primary key of the weak entity vehicle_Factor is Vehicle_ID along with Factor_ID.

Accident *(Accident_no:varchar, Accident_Time: DATETIME,Work_Zone: varchar, Alcohol:varchar, Belt:varchar, Fatality:varchar, Geo Location:varchar, State:varchar, Personal _Injury:varchar, Personal_Damage:varchar, Property_Damage:varchar, Date: DATETIME)*

For **composite attributes** such as Losses and factors, we just store sub attributes.

The Accident table has the following attributes -
The ***Accident_no*** is the primary key used to uniquely identify a tuple in this table.
Accident_Time is an attribute that describes the time of the accident.
Work_Zone is an attribute that indicates whether the accident has taken place in a work zone.
Alcohol is an attribute that indicates the influence of alcohol in the accident.
Belt is an attribute that indicates whether a seat belt was worn during the accident.
Fatality is an attribute that indicates the fatality associated with the accident.
Geolocation is an attribute that specifies the longitude and latitude of the accident.
State is an attribute that specifies the state in which the accident has taken place.
Personal_Injury is an attribute that indicates the injury associated with the accident.
Personal_Damage is an attribute that indicates the damage associated with the accident.
Property_Damage is an attribute that indicates the property damage associated with the accident.
Date is an attribute that indicates the date of the accident. The data type is DATE.

Legal_Case *(CaseNumber: varchar, Violation: varchar, charge: varchar, arrestType:varchar, article:varchar, contributedToAccident:varchar, Accident_no:varchar)*

Since Accident and Legal_Case tables have a 1:n relationship, the primary key of the 1's is added as a foreign key on the n side relation. Thus, **Legal_Case** has **Accident_no**(primary key of accident table) as a **foreign key**.

The Legal_Case table has the following attributes -
The ***CaseNumber*** is the primary key used to uniquely identify a tuple in this table.
Violation is an attribute that describes whether a violation has taken place during the accident.
Charge is an attribute that indicates whether any charges have been filed.
arrestType is an attribute that indicates the type of arrest during the accident.
article is an attribute that indicates the article under which the case was booked.
contributedToAccident is an attribute that indicates the contribution associated with the accident.
Accident_No is the foreign key attribute. It identifies the accident number the legal case is associated with.

Insurance *(Insurance_ID: varchar, Type:varchar, Agency:varchar, SubAgency:varchar, License_No: varchar)*

Since Driver and Insurance tables have a 1:n relationship, the primary key of the 1's is added as a foreign key on the n side relation. Thus, Insurance has License_no(primary key of accident table) as a foreign key.

Insurance_ID: Is the primary key that is used to uniquely identify insurance policies.
Type: the type of insurance the policy belongs to
Agency: Agency the insurance policy belongs to
SubAgency: subagence the insurance policy belongs to
License_NO***:*** The unique identifier for the driver who has the insurance. This acts as the foreign key which is used to reference the table Driver.

Driver *(LicenseNo:varchar, Gender:varchar, Race:varchar, DriverState:varchar, DriverCity:varchar)*

License_No is the primary key that is used to uniquely identify a Driver who is involved in the accident in the relation Driver.
Gender represents the gender of the Driver
Race represents the race of the Race
DriverState The state the driver belongs to.
DriverCity City the Driver belongs to.

Involved_In *(VehicleID:varchar, Accident_no:varchar)*

Since Vehicle and Accident have a m:n relationship cardinality, when mapping to relational schema a new table is to be created whose primary key is a combination of both entities' primary keys. (The primary key of accident is Accident_No and the primary key of Vehicle is VehicleID)

VehicleID: The unique identifier of the vehicle that was involved in the accident.
Accident_No: The accident number given to uniquely identify the accident.

Covering *(Insurance_ID:varchar,Accident_no:varchar)*

Since Insurance and Accident have a **m:n** relationship cardinality, when mapping to relational schema a new table is to be created whose primary key is a combination of both entities' primary keys. (The primary key of accident is Accident_No and the primary key of Insurance is Insurance_ID)

Insurance_ID: The Insurance_ID from table Insurance is used as a partial primary key and a foreign key connecting to the table Insurance.
Accident_No: The Accident_No from Accident table is used as a foreign key and a partial primary key in the Covering table to connect it with Insurance

CausedBy *(License_No:varchar,Accident_no:varchar)*

Since Driver and Accident have a **m:n** relationship cardinality, when mapping to relational schema a new table is to be created whose primary key is a combination of both entities' primary keys. (The primary key of accident is Accident_No and the primary key of Driver is Licence_No)

License_No: The License_no which is the primary key of table Driver which is connected in a many to many relationship with the table Accident.
Accident_No: The Accident_No from table Accident is the foreign key and partial primary key in Caused_by table.

Associated_With *(FactorID: varchar,Accident_no:varchar)*

Since Vehicle_Factor and Accident have a m:n relationship cardinality, when mapping to relational schema a new table is to be created whose primary key is a combination of both entities' primary keys. (The primary key of accident is Accident_No and the primary key of Vehicle_Factor is Factor_ID)
Factor_ID: The Factor_ID from Vehicle_Factor table is used as a foreign key and a partial primary key due to the many to many relationship between Vehicle_Factor and Accident Tables
Accident_No: The Accident_No from Accident table is used as a foreign key and a partial primary key in the Associated_With table to connect it with Vehicle_Factor

Constraints:

1. *MakeYear:* Constraints for years are set as a range of numbers not being less than 1900 and 2021. As no accidents records exist from before 1900 and as current date cannot be used with check command, these values are hardcoded into the table to deter from receiving garbage values.

2. *Sate:* The state is set as an option between 50 states. All the options of states are provided separated with “,”. This ensures that only valid state codes are given.

3. *Gender:* The gender is given constraints between “Male”, “Female” and “Other”.

4. *Boolean Constraint:* As there is no boolean constraint in Oracle sql command, we check if the value is either of “Yes” or “No” values. This constraint is used on multiple attributes like VehicleType.Hazmat, Accident.WorkZone, Accident.Alcohol, Accident.Belt, Accident.Fatality, Accident.Personal_Injury, Accident.Personal_Damage, Accident.Property_Damage, LegalCase.Contributed_To_Accident etc.

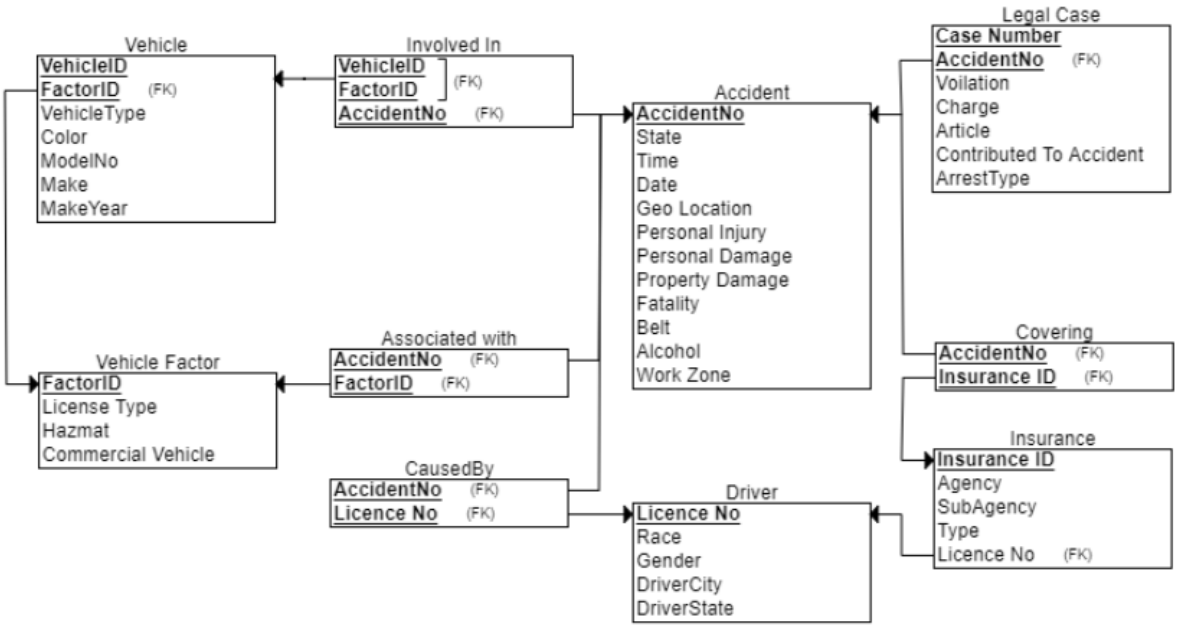

Figure 2: Relationship Diagram

Transformation of Relational Schemas to SQL table schemas

TABLE:

1. *Vehicle Table:*

Vehicle(Vehicle_ID : varchar, vehicle_type : varchar, color: varchar, model_no: varchar, make: varchar, MakeYear: int)

```
Create Table Vehicle
(  VehicleID varchar(50),
   Vehicle_Type varchar(50),
   colour varchar(50),
   ModelNo varchar(50),
   Make varchar(50),
   MakeYear INT CONSTRAINT MakeYearConstraint CHECK (MakeYear>='1900' and
MakeYear<='2021') NOT NULL,
   PRIMARY KEY (VehicleID)
);
```

```
Create Table Vehicle
(   VehicleID varchar(50),
    Vehicle_Type varchar(50),
    colour varchar(50),
    ModelNo varchar(50),
    Make varchar(50),
    MakeYear INT
        CONSTRAINT MakeYearConstraint
            CHECK (MakeYear>='1900' and MakeYear<='2021') NOT NULL,
    PRIMARY KEY (VehicleID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	VEHICLEID	VARCHAR2(50 BYTE)	No	(null)
2	VEHICLE_TYPE	VARCHAR2(50 BYTE)	Yes	(null)
3	COLOUR	VARCHAR2(50 BYTE)	Yes	(null)
4	MODELNO	VARCHAR2(50 BYTE)	Yes	(null)
5	MAKE	VARCHAR2(50 BYTE)	Yes	(null)
6	MAKEYEAR	NUMBER(38,0)	No	(null)

2. *Vehicle Factor Table*

Vehicle_Factor(FactorID: int, commercial vehicle:varchar, Hazmat: varchar, LicenseType: varchar, Vehicle_ID : varchar)

```
Create Table Vehicle_Factor (
   VehicleId varchar(50),
         Factor_ID varchar(50),
         Commercial varchar(50),
         Hazmat varchar(50) DEFAULT 'NO',
        CONSTRAINT HazmatConstraint CHECK (Hazmat IN ('YES','NO')),
         LicenseType varchar(50),
   PRIMARY KEY (Factor_ID, VehicleId),
         foreign key (VehicleId) REFERENCES Vehicle (VehicleID)
);
```

```
Create Table Vehicle_Factor (
    VehicleId varchar(50),
    Factor_ID varchar(50),
    Commercial varchar(50),
    Hazmat varchar(50) DEFAULT 'NO',
            CONSTRAINT HazmatConstraint CHECK (Hazmat IN ('YES','NO')),
    LicenseType varchar(50),
    PRIMARY KEY (Factor_ID, VehicleId),
    foreign key (VehicleId) REFERENCES Vehicle (VehicleID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	VEHICLEID	VARCHAR2(50 BYTE)	No	(null)
2	FACTOR_ID	VARCHAR2(50 BYTE)	No	(null)
3	COMMERCIAL	VARCHAR2(50 BYTE)	Yes	(null)
4	HAZMAT	VARCHAR2(50 BYTE)	Yes	'NO'
5	LICENSETYPE	VARCHAR2(50 BYTE)	Yes	(null)

3. *Accident Table*

Accident(Accident_no: varchar, Time: DATETIME,Work Zone: varchar, Alcohol:varchar, Belt:varchar, Fatality:varchar, Geo Location:varchar, State:varchar, Personal Injury:varchar, Personal Damage:varchar, Property Damage:varchar, Date: DATETIME)

```
Create Table "Accident" (    Accident_no varchar(50),
   Accident_Time DATE,
   Work_Zone varchar(50) DEFAULT 'NO',
      CONSTRAINT WorkConstraint CHECK (Work_Zone IN ('YES','NO')),
   Alcohol varchar(3) DEFAULT 'NO',
      CONSTRAINT AlcoholConstraint CHECK (Alcohol IN ('YES','NO')),
   Belt varchar(50) DEFAULT 'NO',
      CONSTRAINT BeltConstraint CHECK (Belt IN ('YES','NO')),
   Fatality varchar(50) DEFAULT 'NO',
      CONSTRAINT FatalityConstraint CHECK (Fatality IN ('YES','NO')),
   Geo_Location varchar(50),
   State varchar(50),
      CONSTRAINT StateConstraint CHECK (State IN ('AL','AK','AZ','AR','CA',
      'CO','CT','DE','FL','GA','HI','ID','IL','IN','IA','KS','KY','LA','ME',
      'MA','MI','MN','MS','MO','MT','NE','NV','NH','NJ','NM','NY','NC','ND',
      'OH','OK','OR','PA','RI','SC','SD','TN','TX','UT','VT','VA','WA','WV',
      'WI','MD','WY')),
   Personal_Injury varchar(50) DEFAULT 'NO',
      CONSTRAINT InjuryConstraint CHECK (Personal_Injury IN ('YES','NO')),
   Personal_Damage varchar(50) DEFAULT 'NO',
            CONSTRAINT PersonalDamageConstraint CHECK (Personal_Damage IN ('YES','NO')),
   Property_Damage varchar(50) DEFAULT 'NO',
            CONSTRAINT Property_DamageConstraint CHECK (Property_Damage IN ('YES','NO')),
   AccidentDate DATE,
   primary key (Accident_no)
);
```

```
Create Table "Accident" (    Accident_no varchar(50),
    Accident_Time DATE,
    Work_Zone varchar(50) DEFAULT 'NO',
        CONSTRAINT WorkConstraint CHECK (Work_Zone IN ('YES','NO')),
    Alcohol varchar(3) DEFAULT 'NO',
        CONSTRAINT AlcoholConstraint CHECK (Alcohol IN ('YES','NO')),
    Belt varchar(50) DEFAULT 'NO',
        CONSTRAINT BeltConstraint CHECK (Belt IN ('YES','NO')),
    Fatality varchar(50) DEFAULT 'NO',
        CONSTRAINT FatalityConstraint CHECK (Fatality IN ('YES','NO')),
    Geo_Location varchar(50),
    State varchar(50),
        CONSTRAINT StateConstraint CHECK (State IN ('AL','AK','AZ','AR','CA',
        'CO','CT','DE','FL','GA','HI','ID','IL','IN','IA','KS','KY','LA','ME',
        'MA','MI','MN','MS','MO','MT','NE','NV','NH','NJ','NM','NY','NC','ND',
        'OH','OK','OR','PA','RI','SC','SD','TN','TX','UT','VT','VA','WA','WV',
        'WI','MD','WY')),
    Personal_Injury varchar(50) DEFAULT 'NO',
        CONSTRAINT InjuryConstraint CHECK (Personal_Injury IN ('YES','NO')),
    Personal_Damage varchar(50) DEFAULT 'NO',
        CONSTRAINT PersonalDamageConstraint CHECK (Personal_Damage IN ('YES','NO')),
    Property_Damage varchar(50) DEFAULT 'NO',
        CONSTRAINT Property_DamageConstraint CHECK (Property_Damage IN ('YES','NO')),
    AccidentDate DATE,
    primary key (Accident_no));
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	ACCIDENT_NO	VARCHAR2(50 BYTE)	No	(null)
2	ACCIDENT_TIME	DATE	Yes	(null)
3	WORK_ZONE	VARCHAR2(50 BYTE)	Yes	'NO'
4	ALCOHOL	VARCHAR2(3 BYTE)	Yes	'NO'
5	BELT	VARCHAR2(50 BYTE)	Yes	'NO'
6	FATALITY	VARCHAR2(50 BYTE)	Yes	'NO'
7	GEO_LOCATION	VARCHAR2(50 BYTE)	Yes	(null)
8	STATE	VARCHAR2(50 BYTE)	Yes	(null)
9	PERSONAL_INJURY	VARCHAR2(50 BYTE)	Yes	'NO'
10	PERSONAL_DAMAGE	VARCHAR2(50 BYTE)	Yes	'NO'
11	PROPERTY_DAMAGE	VARCHAR2(50 BYTE)	Yes	'NO'
12	ACCIDENTDATE	DATE	Yes	(null)

4. *Legal Case*

Legal_*Case*(CaseNumber: varchar, Violation: varchar, charge: varchar, arrestType:varchar, article:varchar, contributedToAccident:varchar, Accident_no:varchar)

```
create table Legal_Case
(       Case_Number varchar(50),
        Violation varchar(50),
        Charge varchar(50),
```

```
        Arrest_Type varchar(50),
        Article varchar(50),
  Contributed_To_Accident varchar(50) DEFAULT 'NO',
    CONSTRAINT ContributedConstraint
      CHECK (Contributed_To_Accident IN ('YES','NO')),
  Accident_no varchar(50),
  foreign key (accident_no) REFERENCES Accident(Accident_no),
  primary key(Case_Number,Accident_no)
);
```

```
create table Legal_Case
(   Case_Number varchar(50),
    Violation varchar(50),
    Charge varchar(50),
    Arrest_Type varchar(50),
    Article varchar(50),
    Contributed_To_Accident varchar(50) DEFAULT 'NO',
        CONSTRAINT ContributedConstraint
            CHECK (Contributed_To_Accident IN ('YES','NO')),
    Accident_no varchar(50),
    foreign key (accident_no) REFERENCES Accident(Accident_no),
    primary key(Case_Number,Accident_no)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	CASE_NUMBER	VARCHAR2(50 BYTE)	No	(null)
2	VIOLATION	VARCHAR2(50 BYTE)	Yes	(null)
3	CHARGE	VARCHAR2(50 BYTE)	Yes	(null)
4	ARREST_TYPE	VARCHAR2(50 BYTE)	Yes	(null)
5	ARTICLE	VARCHAR2(50 BYTE)	Yes	(null)
6	CONTRIBUTED_TO_ACCIDENT	VARCHAR2(50 BYTE)	Yes	'NO'
7	ACCIDENT_NO	VARCHAR2(50 BYTE)	Yes	(null)

5. *Driver Table*

Driver(LicenseNo:varchar, Gender:varchar, Race:varchar,DriverState:varchar,DriverCity:varchar)

```
create table Driver(
        License_No varchar(50),
        Gender varchar(20),
        CONSTRAINT GenderConstraint CHECK (Gender IN ('Female','Male','Others')),
        Race varchar(50),
        DriverState varchar(50),
        DriverCity varchar(50),
        primary key (License_No)
);
```

```
create table Driver(
    License_No varchar(50),
    Gender varchar(20),
     CONSTRAINT GenderConstraint CHECK (Gender IN ('Female','Male','Others')),
    Race varchar(50),
    DriverState varchar(50),
        DriverCity varchar(50),
        primary key (License_No)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	LICENSE_NO	VARCHAR2(50 BYTE)	No	(null)
2	GENDER	VARCHAR2(20 BYTE)	Yes	(null)
3	RACE	VARCHAR2(50 BYTE)	Yes	(null)
4	DRIVERSTATE	VARCHAR2(50 BYTE)	Yes	(null)
5	DRIVERCITY	VARCHAR2(50 BYTE)	Yes	(null)

6. *Insurance Table*

Insurance(Insurance_ID:varchar, Type:varchar, Agency:varchar, SubAgency:varchar,License_No:varchar)

```
create table Insurance (
   Insurance_ID  varchar(50),
   Type varchar(50),
   Agency varchar(50),
   License_No varchar(50),
   Primary key (Insurance_ID),
   foreign key (License_No) REFERENCES Driver (License_No)
);
```

```
create table Insurance (
    Insurance_ID  varchar(50),
    Type varchar(50),
    Agency varchar(50),
    License_No varchar(50),
    Primary key (Insurance_ID),
    foreign key (License_No) REFERENCES Driver (License_No)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	INSURANCE_ID	VARCHAR2(50 BYTE)	No	(null)
2	TYPE	VARCHAR2(50 BYTE)	Yes	(null)
3	AGENCY	VARCHAR2(50 BYTE)	Yes	(null)
4	LICENSE_NO	VARCHAR2(50 BYTE)	Yes	(null)

7. *Involved in Table*

Involved_In(VehicleID:varchar,Accident_no:varchar)

```
create table Involved_In (
            VehicleID varchar(50),
            Accident_no varchar(50),
            PRIMARY KEY (VehicleID, Accident_no),
            foreign key (VehicleID) REFERENCES Vehicle (VehicleID),
            foreign key (Accident_no) REFERENCES Accident (Accident_no)
);
```

```
create table Involved_In (
    VehicleID varchar(50),
    Accident_no varchar(50),
    PRIMARY KEY (VehicleID, Accident_no),
    foreign key (VehicleID) REFERENCES Vehicle (VehicleID),
    foreign key (Accident_no) REFERENCES Accident (Accident_no)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	VEHICLEID	VARCHAR2(50 BYTE)	No
2	ACCIDENT_NO	VARCHAR2(50 BYTE)	No

8. *Covering Table*

Covering(Insurance_ID:varchar,Accident_no:varchar)

```
create table Covering (
   Insurance_ID  varchar(50),
         Accident_no varchar(50),
         PRIMARY KEY (Insurance_ID, Accident_no),
   foreign key (Insurance_ID) REFERENCES Insurance (Insurance_ID),
   foreign key (Accident_no) REFERENCES Accident (Accident_no)
);
```

```
create table Covering (
    Insurance_ID  varchar(50),
    Accident_no varchar(50),
    PRIMARY KEY (Insurance_ID, Accident_no),
    foreign key (Insurance_ID) REFERENCES Insurance (Insurance_ID),
    foreign key (Accident_no) REFERENCES Accident (Accident_no)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	INSURANCE_ID	VARCHAR2(50 BYTE)	No
2	ACCIDENT_NO	VARCHAR2(50 BYTE)	No

9. *Caused By Table*

CausedBy(LicenseNo:varchar,Accident_no:varchar)

```
create table CausedBy (
        License_NO  varchar(50),
        Accident_no varchar(50),
        PRIMARY KEY (License_NO, Accident_no),
        foreign key (License_NO) REFERENCES Driver (License_NO),
        foreign key (Accident_no) REFERENCES Accident (Accident_no)
);
```

```
create table CausedBy (
    License_NO  varchar(50),
    Accident_no varchar(50),
        PRIMARY KEY (License_NO, Accident_no),
    foreign key (License_NO) REFERENCES Driver (License_NO),
        foreign key (Accident_no) REFERENCES Accident (Accident_no)
);
```

1	LICENSE_NO	VARCHAR2(50 BYTE)	No
2	ACCIDENT_NO	VARCHAR2(50 BYTE)	No

10. *Associated with Table*

Associated_With (Factor_ID: varchar,Accident_no:varchar)

```
create table Associated_With (
        Factor_ID  varchar(50),
        Accident_no varchar(50),
        PRIMARY KEY (Factor_ID, Accident_no),
        foreign key (Factor_ID) REFERENCES Vehicle_Factor (Factor_ID),
        foreign key (Accident_no) REFERENCES Accident (Accident_no)
);
```

```
create table Associated_With (
    Factor_ID  varchar(50),
    Accident_no varchar(50),
PRIMARY KEY (Factor_ID, Accident_no),
foreign key (Factor_ID) REFERENCES Vehicle_Factor (Factor_ID),
foreign key (Accident_no) REFERENCES Accident (Accident_no)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	FACTOR_ID	VARCHAR2(50 BYTE)	No
2	ACCIDENT_NO	VARCHAR2(50 BYTE)	No

Work Division

Below is the work division on each phase of the product.

Database: Kasiviswanathan Srikant Iyer, Kavya Gopal
PHP (backend): Rema Veeranna Gowda, Kasiviswanathan Srikant Iyer
Website frontend: Swaathi Reena Velavan, Rema Veeranna Gowda
Queries: Kavya Gopal, Swaathi Reena Velevan

For every consecutive phase the responsibilities will be passed on in a cyclic manner so that everyone gets to work on every part of the project and will be registered on the documentation submitted further.