Motor Vehicle Collisions Report

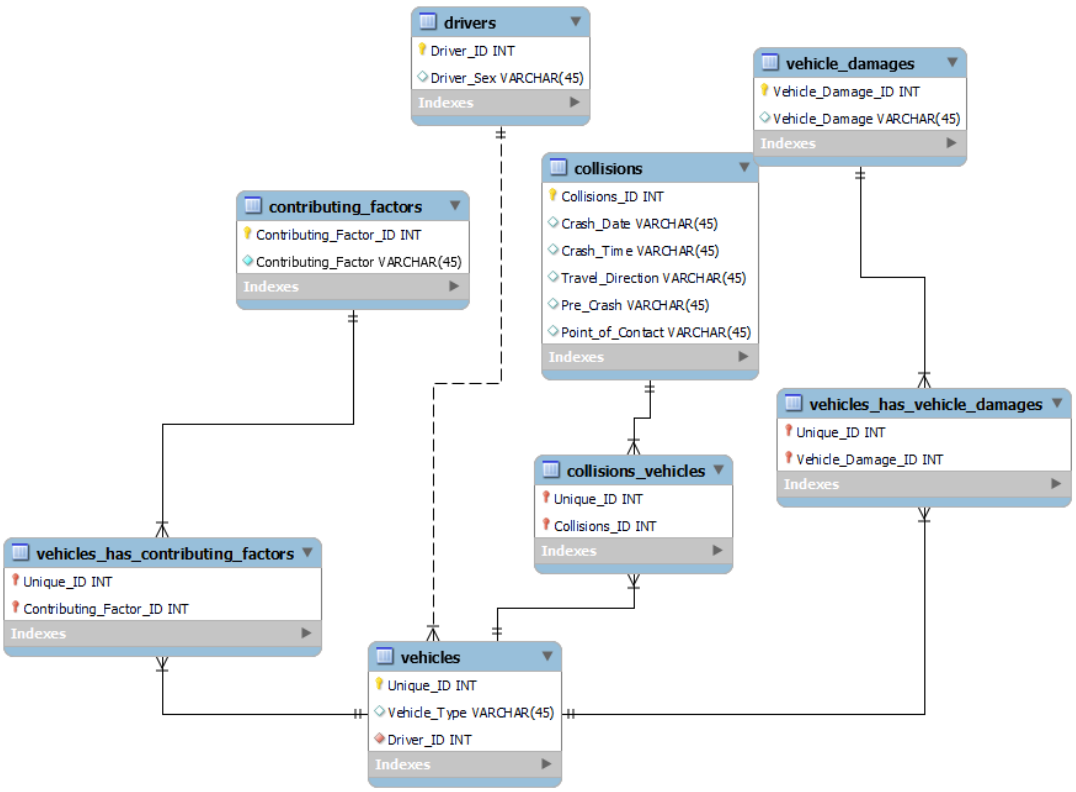
By Helga Lau, Bao Nguyen, Ronan Tocci, Tejasa Yaddula, and Omar Raza

Introduction:

The New York City Passenger Cars Collision is a comprehensive information repository designed to access reliable and detailed data concerning passenger vehicle collisions within New York. Generally, car collisions are a critical issue that everyone worries about. Vehicle collisions can severely impact human lives, properties, public safety, and transportation infrastructures. There are a lot of people who lost their lives because of accidents. Understanding that, we made a database focusing mainly on passenger vehicle collisions in New York City. Our accurate data on passenger car collisions will be a solid and reliable source to understand the trend. Also, our data can help to identify the risk factors. Moreover, this data can be an essential resource to permit law enforcement to have a better viewpoint; then, they can perform better plans to prevent or decrease the rate of accidents.

Database Description

1. ERD:



2. **Physical Database:**
    - **Entities and Relationships:** We have potential entities such as drivers, vehicle_damage, collisions, collision_vehicle, contributing_factors, etc.
    - **Design tables:** We create tables with primary and foreign keys such as unique_id, collision_id, etc.
    - **Making Relationships:** We linked all tables by using primary keys. For example, the driver's table and vehicles table are linked by unique_id.
    - **Implement Database:** For example, when we create the table collisions, we use this syntax:

```
CREATE TABLE `COLLISIONS` (
 `Unique_ID` int NOT NULL,
 `Crash_Date` DATE DEFAULT NULL,
 `Crash_Time` TIME DEFAULT NULL,
 `Travel_Direction` varchar(45) DEFAULT NULL,
 `Pre_Crash` varchar(45) DEFAULT NULL,
 `Point_of_Contact` varchar(45) DEFAULT NULL,
 PRIMARY KEY (`Unique_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
/*!40101 SET character_set_client = @saved_cs_client */;
```

    - **Database Testing:** At this step, we inserted some data into tables to test out if it works properly.

3. **Sample Data:**

We have created eight different data sheets for our sample data. The tables contain the following data: collision ID, crash date, crash time, travel direction, pre-crash, point of

impact, Unique ID, contributing factor ID, driver ID, vehicle ID, vehicle damage, driver sex, and contributing factor.

4. **Views/ Queries:**

1.)
```
CREATE VIEW Vehicles_with_Factors AS
SELECT V.Vehicle_Type, CF.Contributing_Factor
FROM VEHICLES V
JOIN VEHICLES_has_CONTRIBUTING_FACTORS VCF ON V.Vehicle_ID =
VCF.VEHICLES_Vehicle_ID
JOIN CONTRIBUTING_FACTORS CF ON
VCF.CONTRIBUTING_FACTORS_Contributing_Factor_ID =
CF.Contributing_Factor_ID
WHERE V.Vehicle_ID IN (SELECT Vehicle_ID FROM DRIVERS WHERE Driver_Sex
= 'Male');
```

2.)
```
CREATE VIEW Modified_Morning_Collisions AS
SELECT Unique_ID, Crash_Date, Crash_Time
FROM COLLISIONS
WHERE Crash_Time BETWEEN '06:00' AND '12:00';
```

3.)
```
CREATE VIEW Car_Collisions_With_Driver_Info  AS
SELECT C.Unique_ID, V.Vehicle_Type, D.Driver_Sex
FROM COLLISIONS C
JOIN COLLISIONS_VEHICLES CV ON C.Unique_ID = CV.COLLISIONS_Unique_ID
JOIN VEHICLES V ON CV.VEHICLES_Vehicle_ID = V.Vehicle_ID
JOIN DRIVERS D ON V.Vehicle_ID = D.Vehicle_ID
WHERE V.Vehicle_Type = 'Car';
```

4.)
```
SELECT Vehicle_Type AS 'Vehicle Type', COUNT(*) AS 'Number of Accidents' FROM
vehicles
GROUP BY Vehicle_Type
ORDER BY COUNT(*) DESC;
```

```
SELECT Vehicle_Type AS 'Vehicle Type', COUNT(*) AS 'Front Bumper Damage'
FROM vehicles
LEFT JOIN vehicles_has_vehicle_damages ON vehicles.Vehicle_ID =
vehicles_has_vehicle_damages.VEHICLES_Vehicle_ID
```

```
INNER JOIN vehicle_damages ON vehicle_damages.Vehicle_Damage_ID =
vehicles_has_vehicle_damages.VEHICLE_DAMAGES_Vehicle_Damage_ID
WHERE vehicle_damages.Vehicle_damage = ('Left Front Bumper' OR 'Right Front
Bumper')
GROUP BY Vehicle_Type
ORDER BY COUNT(*) DESC;
```

5.)

```
SELECT Vehicle_Type, Crash_Date, Crash_Time FROM collisions
LEFT JOIN collisions_vehicles ON collisions.Unique_ID =
collisions_vehicles.COLLISIONS_Unique_ID
INNER JOIN vehicles ON vehicles.Vehicle_ID =
collisions_vehicles.VEHICLES_Vehicle_ID
ORDER BY Crash_Time;
```

6.)

```
CREATE VIEW Inattentive_Driver_Collisions AS
SELECT V.Vehicle_Type, C.Crash_Date, C.Crash_Time
FROM VEHICLES V
JOIN COLLISIONS_VEHICLES CV ON V.Vehicle_ID = CV.VEHICLES_Vehicle_ID
JOIN COLLISIONS C ON CV.COLLISIONS_Unique_ID = C.Unique_ID
JOIN VEHICLES_has_CONTRIBUTING_FACTORS VCF ON V.Vehicle_ID =
VCF.VEHICLES_Vehicle_ID
JOIN CONTRIBUTING_FACTORS CF ON
VCF.CONTRIBUTING_FACTORS_Contributing_Factor_ID =
CF.Contributing_Factor_ID
WHERE CF.Contributing_Factor = 'Driver Inattention/Distraction';
```

	Req. A	Req. B	Req. C	Req. D	Req. E
Vehicles_with_Factors		X		X	X
Morning_Collisions		X			

Car_Collisions_ With_Driver_Info	X	X		X
4	X		X	
5	X			
6	X		X	

Changes From the Original Design:

Initially, our team decided to sort the vehicles by Passenger Vehicles to make the dataset smaller and more workable. However, sorting by Passenger Vehicles wasn't the best idea. When we created our data sheets, we noticed that some columns were null and had no data. This forced us to transform our database entirely, and we have now sorted by the 36 collisions that occurred on October 24, 2019. This yielded enough data, and therefore, we wrote better queries. Important note: Two data sheets we've created don't meet the 15-entry minimum requirements. This is the Vehicle Damages and Contributing Factors tables. There were not enough different vehicle damage types and contributing factors to get us to 15 entries.

Database Ethics Consideration:

When designing, we have some considerations about the ethics of the database. We have nine considerations on our list:

1. **Data Privacy**: We ensure that all the data related to all individuals from different parties are protected and safe.

2. **Data Fairness:** Our data is firmly fair for everyone, every organization, when it comes to collecting, storing, transferring, processing, and analyzing data. Also, the data should not disproportionately impact any particular group or community. We tried to identify and resolve any discrepancies in the data.
3. **Data Accuracy**: Our data is collected carefully to ensure accuracy and detail. We can have incorrect conclusions with incorrect data and do not want biased data.
4. **Transparency**: We can provide how we collect and analyze data. We have to make sure our data is used the right way.
5. **Data Security:** Implement all the robust methods to secure the data from misuse and unauthorized access.
6. **Lifespan:** Planning the lifespan of data, we can set it within a time to view and access. Exceeding the date of the period time, data will be removed.
7. **Limited Access:** Our data is not public. It can only be accessed by authorized organizations or personnel.
8. **Feedback:** We are considering adding a tool that lets users leave feedback about unfairness and biased data if it ever has one.
9. **Ethical Data Use:** We want to ensure our data is used properly and responsibly. Preventing using the data for personal purposes, misusing, or making any non-sense decision leads to negative unforeseen consequences.

Lessons Learned:

Our main issue was communication during our project. Our team needed help tracking which members did which steps for our database. Our group met more on video calls than via messages to ensure we communicated better. This helped as we could delegate project segments and help each other when needed. Creating and importing data was an issue as our team had initially made the wrong data, and we were importing it incorrectly. By communicating with the professor during office hours, we resolved the issue. We learned that real-time communication and asking for help is the best method for resolving problems; we will be sure to implement this in the future. In determining the issue of importing data, we had to remember to pause every

section of the dialogue box in the SQL server. Hence, we learned to double-check every element of the workspace, and we will carry this on in the future when working with new applications.

Potential Future Work:

If we were to work more on our database in the future, we would look for patterns in the data and add data related to categories of accidents. We can add data on what can be changed for each casualty category to prevent that accident. This future work will be of use to traffic regulators and officials, and state legislators in avoiding future accidents.