

ACCIDENTS DATABASE

ABSTRACT

The objective of this project is to collect, clean and build Accidents database . The primary source of extraction is data repositories from web and the data is cleaned to obtain a consistent dataset. Furthermore, the dataset was cleaned and audited to make it unique and error free. Data was successfully loaded in Database.

A number of intrinsic and contextual attributes, including severity, location, time, temperature, humidity, visibility, crossing etc. are included in each accident record. This database aids in our understanding of how the aforementioned factors affect accidents.

Crashes data includes crash event level details such as the nearest intersection, with distance and direction of the crash from nearest intersection. Also included are the involved party (vehicle involved with), primary collision factor.

Vehicles data includes the vehicle level details of the crash such as vehicle types, driver's (vehicle, party) age and sex, driver conditions, severity of injury in terms of fatalities, and severe, moderate, and minor injuries per crash and violations preceding the crash.

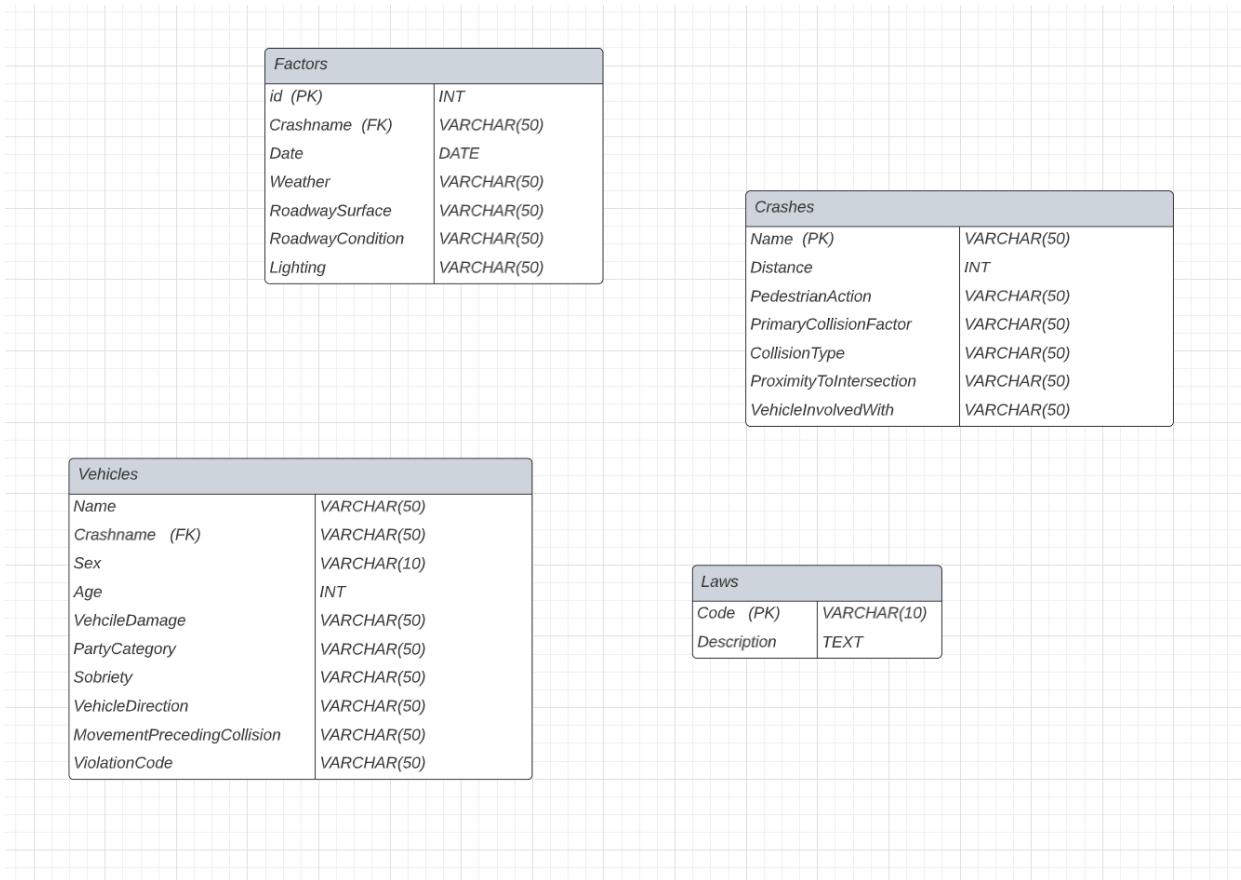
There is a one-to-many relationship built that relates the crash to the vehicles involved. The Crash name in vehicle data relates to the Name in the Crash data.

Laws data gives information related to the violation codes and their respective descriptions.

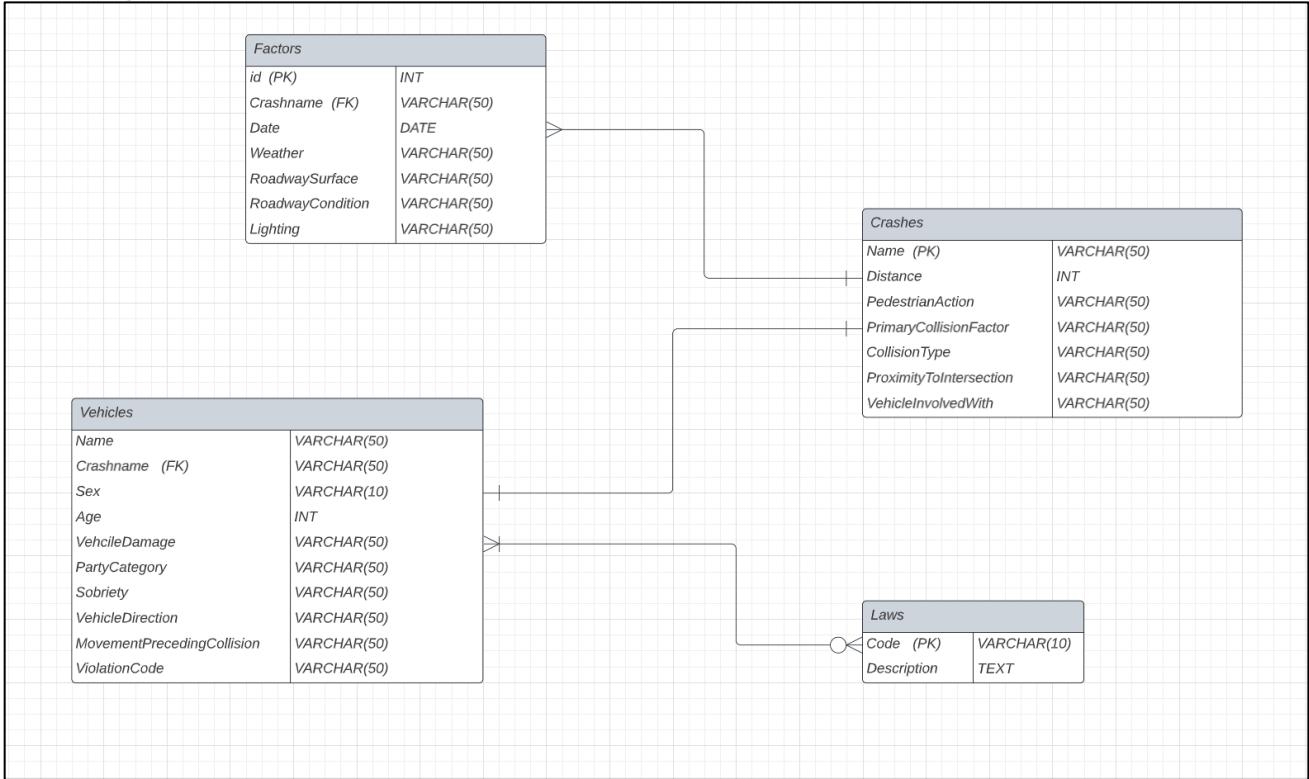
Factors data represents weather, roadway surface, different periods of day, roadway conditions, and time of day.

Initial Schema:

At inception of the project the basic DB schema was designed as depicted below.



ER Diagram



Steps followed in creating the final Database

Sources of Data:

To obtain relevant and high-quality data, choose the proper data source. For this assignment, we have collected most of the data from data repository.

- <https://data.sanjoseca.gov/dataset/crashes-data>
- <https://data.sanjoseca.gov/dataset/918fb7f0-60c0-484e-b31c-334d1ec74e92/resource/c19a01f2-33e1-4c66-9498-85d489f90da4/download/crashdata2011-2020.csv>
- <https://data.sanjoseca.gov/dataset/918fb7f0-60c0-484e-b31c-334d1ec74e92/resource/a92d2d50-f8c5-46a3-a0a8-5eaf8b3a865f/download/vehiclecrashdata2021-present.csv>

Download and reformat the data:

The considered data was raw data and we audited, cleaned, and validated with completeness. This involved downloading and reformatting the raw data.

Data Validation:

Completeness: refers to the extent to which an entity contains the information needed to describe a real-world object. The presence of null values, which are typically regarded as missing values, in tables in relational database systems can be used to determine how full a table is.

Consistency: The degree to which a set of semantic rules are violated such as a specific data type, an interval for a numerical column, or a set of values for a categorical column.

Accuracy: The correctness of the data and can be measured in two dimensions: syntactic and semantic. Semantic accuracy contrasts a value with its actual representation, while syntactic accuracy compares a value's representation with a domain of definition that corresponds

Data Cleaning Process

Row values with NULL and unknown have been removed to maintain data accuracy.

40% of empty rows in the Age column are filled with the mean value.

Irrelevant row values having Age factor > 18 are preserved, eliminating the remaining.

30% of blank rows in the Sex column are adjusted with the nearest row string value.

CSV Files

All Data which has been obtained after the data cleaning procedure has been attached in a CSV file folder to the GitHub link.

Database Loaders

Python script is used to create Database and Tables. And add the extracted Accidents data into the MySQL Database.

NORMALIZING THE DATA

To create a structured Database for the data obtained from data repositories we need to normalize it. So as to normalize our data, we have separated the raw data into four different tables

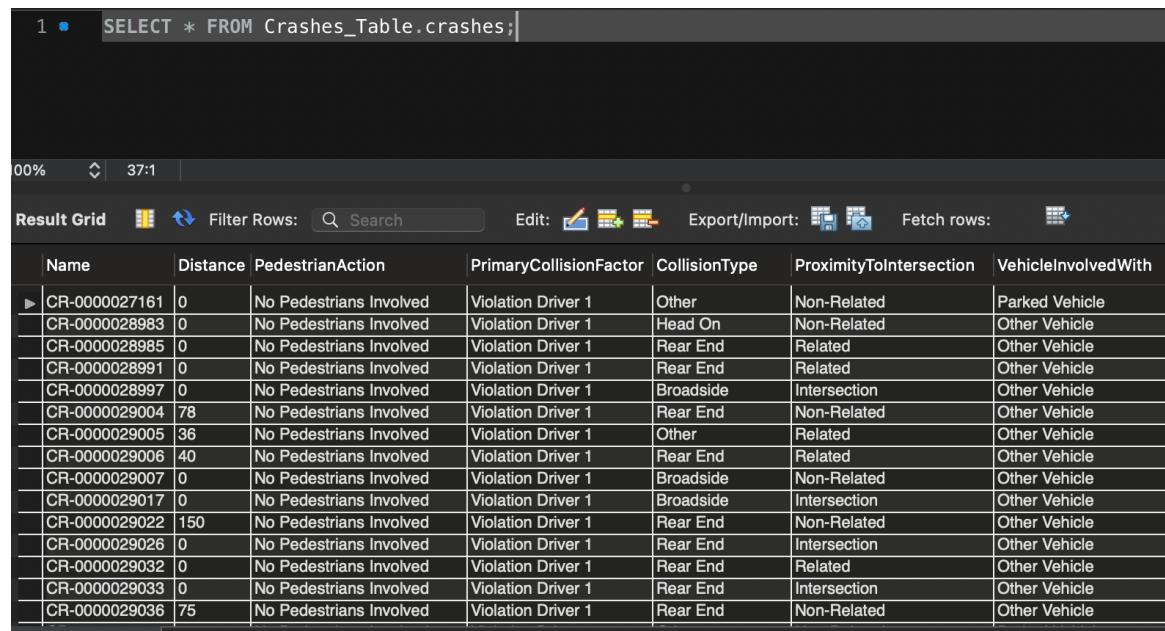
First Normal Form

First normal form (1NF) is a property of a relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain. Also, primary keys should be uniquely identified.

All the tables considered are in 1NF as all the below conditions are met:

1. There are only Single Valued Attributes. (Not a multi-valued and a composite attribute)
2. Attribute Domain does not change.
3. There is a unique name for every Attribute/Column.

Crashes Table data



The screenshot shows a database query results grid. At the top, there is a command line interface with the text "1 • SELECT * FROM Crashes_Table.crashes;". Below this is a progress bar at 00% and a timestamp of 37:1. The main area is a "Result Grid" with various controls like "Edit", "Filter Rows", "Search", "Export/Import", and "Fetch rows". The grid has columns labeled "Name", "Distance", "PedestrianAction", "PrimaryCollisionFactor", "CollisionType", "ProximityToIntersection", and "VehicleInvolvedWith". The data consists of 16 rows of crash information, such as "CR-0000027161" with "Distance" 0, "PedestrianAction" "No Pedestrians Involved", "PrimaryCollisionFactor" "Violation Driver 1", and "CollisionType" "Other".

Name	Distance	PedestrianAction	PrimaryCollisionFactor	CollisionType	ProximityToIntersection	VehicleInvolvedWith
CR-0000027161	0	No Pedestrians Involved	Violation Driver 1	Other	Non-Related	Parked Vehicle
CR-0000028983	0	No Pedestrians Involved	Violation Driver 1	Head On	Non-Related	Other Vehicle
CR-0000028985	0	No Pedestrians Involved	Violation Driver 1	Rear End	Related	Other Vehicle
CR-0000028991	0	No Pedestrians Involved	Violation Driver 1	Rear End	Related	Other Vehicle
CR-0000028997	0	No Pedestrians Involved	Violation Driver 1	Broadside	Intersection	Other Vehicle
CR-0000029004	78	No Pedestrians Involved	Violation Driver 1	Rear End	Non-Related	Other Vehicle
CR-0000029005	36	No Pedestrians Involved	Violation Driver 1	Other	Related	Other Vehicle
CR-0000029006	40	No Pedestrians Involved	Violation Driver 1	Rear End	Related	Other Vehicle
CR-0000029007	0	No Pedestrians Involved	Violation Driver 1	Broadside	Non-Related	Other Vehicle
CR-0000029017	0	No Pedestrians Involved	Violation Driver 1	Broadside	Intersection	Other Vehicle
CR-0000029022	150	No Pedestrians Involved	Violation Driver 1	Rear End	Non-Related	Other Vehicle
CR-0000029026	0	No Pedestrians Involved	Violation Driver 1	Rear End	Intersection	Other Vehicle
CR-0000029032	0	No Pedestrians Involved	Violation Driver 1	Rear End	Related	Other Vehicle
CR-0000029033	0	No Pedestrians Involved	Violation Driver 1	Rear End	Intersection	Other Vehicle
CR-0000029036	75	No Pedestrians Involved	Violation Driver 1	Rear End	Non-Related	Other Vehicle

Factors Table data

```
1 •  SELECT * FROM Crashes_Table.Factors;
```

100% 37:1

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

	id	Crashname	Date	Weather	RoadwaySurface	RoadwayCondition	Lighting
▶	1	CR-0000029061	2011-01-05	Clear	Dry	No Unusual Conditions	Daylight
	2	CR-0000029463	2011-01-21	Clear	Dry	No Unusual Conditions	Daylight
	3	CR-0000029283	2011-01-11	Rain	Wet	No Unusual Conditions	Daylight
	4	CR-0000027128	2011-02-11	Clear	Dry	No Unusual Conditions	Daylight
	5	CR-0000029284	2011-01-11	Cloudy	Dry	No Unusual Conditions	Daylight
	6	CR-0000029062	2011-01-05	Clear	Dry	No Unusual Conditions	Daylight
	7	CR-0000029285	2011-01-11	Clear	Dry	No Unusual Conditions	Dark - Street Light
	8	CR-0000029464	2011-01-21	Clear	Dry	No Unusual Conditions	Dusk - Dawn
	9	CR-0000029286	2011-01-11	Cloudy	Wet	No Unusual Conditions	Dark - Street Light
	10	CR-0000029063	2011-01-05	Clear	Dry	No Unusual Conditions	Dark - Street Light
	11	CR-0000029287	2011-01-11	Clear	Dry	No Unusual Conditions	Dark - Street Light
	12	CR-0000027161	2011-02-06	Clear	Dry	No Unusual Conditions	Dark - Street Light
	13	CR-0000029064	2011-01-05	Clear	Dry	No Unusual Conditions	Daylight

Laws Table data

```
1 •  SELECT * FROM Crashes_Table.laws;
```

100% 34:1

Result Grid Filter Rows: Search Edit: Export/Import:

	Code	Description
▶	21000	Wherever in this division department occurs, it means the Department of the California...
	21001	The provisions of this division refer exclusively to the operation of vehicles upon the hig...
	2105	Every person riding or driving an animal upon a highway has all of the rights and is subj...
	21051	The following sections apply to trolley coaches:
	21052	The provisions of this code applicable to the drivers of vehicles upon the highways appl...
	21053	This code, except Chapter 1 (commencing with Section 20000) of Division 10, Article 2 (...
	21054	The provisions of this division do not apply to the duly authorized representatives of any...
	21055	The driver of an authorized emergency vehicle is exempt from Chapter 2 (commencing...
	21056	Section 21055 does not relieve the driver of a vehicle from the duty to drive with due re...
	21057	Every police and traffic officer is hereby expressly prohibited from using a siren or drivin...
	21058	A physician traveling in response to an emergency call shall be exempt from the provisi...
	21059	Sections 21211, 21650, 21660, 22502, 22504, and subdivision (h) of Section 22500 do...
	21060	Between the hours of 1 a.m. and 5 a.m., Sections 21650, 21660, 22502, 22504, and su...
	21061	(a) In addition to any action prescribed in Division 17 (commencing with Section 40000....
	21062	The arresting officer shall, before the end of the next working day, transmit, or cause to...

Vehicles Table data

The screenshot shows a database query results grid. At the top, there is a command line interface with the text "1 • SELECT * FROM Crashes_Table.Vehicles;". Below this is a toolbar with various icons for editing, filtering, and exporting data. The main area is a "Result Grid" containing 15 rows of data. The columns are: Name, Crashname, Sex, Age, VehicleDamage, PartyCategory, Sobriety, VehicleDirecti..., MovementPrecedingCollision, and Violation. The data includes various crash details such as driver information, vehicle damage, and movement directions.

Name	Crashname	Sex	Age	VehicleDamage	PartyCategory	Sobriety	VehicleDirecti...	MovementPrecedingCollision	Violation
ACV-00000000030	CR-0000063652	M	26	Minor	Driver	Impairment Not Known	East	Proceeding Straight	1
ACV-00000000032	CR-0000064498	F	50	Minor	Driver	Had Not Been Drinking	South	Parking Maneuver	1
ACV-00000000033	CR-0000068721	M	19	Minor	Driver	Had Not Been Drinking	North	Proceeding Straight	1
ACV-00000000039	CR-0000063854	M	76	Major	Driver	Had Not Been Drinking	West	Proceeding Straight	1
ACV-00000000042	CR-0000068989	M	48	Minor	Driver	Impairment Not Known	East	Slowing/Stopping	1
ACV-00000000043	CR-0000069177	M	20	Minor	Driver	Had Not Been Drinking	West	Making Left Turn	1
ACV-00000000044	CR-0000068965	F	18	Moderate	Driver	Had Not Been Drinking	West	Proceeding Straight	1
ACV-00000000045	CR-0000069519	M	26	Major	Driver	Impairment Not Known	South	Proceeding Straight	1
ACV-00000000046	CR-0000065216	F	38	Major	Driver	Had Not Been Drinking	West	Proceeding Straight	1
ACV-00000000047	CR-0000069560	M	45	Minor	Driver	Had Not Been Drinking	East	Slowing/Stopping	1
ACV-00000000048	CR-0000063080	M	62	Moderate	Driver	Had Not Been Drinking	West	Proceeding Straight	1
ACV-00000000049	CR-0000063051	M	38	Major	Driver	Had Not Been Drinking	South	Ran Off Road	1
ACV-00000000050	CR-0000068520	F	20	Moderate	Driver	Had Not Been Drinking	North	Entering Traffic	1
ACV-00000000051	CR-0000063874	M	36	Moderate	Driver	Had Not Been Drinking	West	Ran Off Road	1
ACV-00000000052	CR-0000068895	M	26	Moderate	Driver	Impairment Not Known	North	Proceeding Straight	1

Second Normal Form

A relation is in the second normal form if it satisfies following two criteria:

1. All requirements for 1st NF must be met.
2. No partial dependencies.
3. All the columns are dependent on only one column i.e., the primary key column.

Crashes, Factors Laws, and the Vehicles table have only a single key as a primary key and are already in 1NF. We find no partial dependencies in any of the attributes.

Third Normal Form

A relation is called to be in a 3rd Normal Form if all the following criteria are satisfied:

1. All requirements for 2nd NF must be met.
2. Eliminate fields that do not directly depend on the primary key; that is no transitive dependencies.

The final database has 4 tables, which are normalized with reference to the above definition. Displaying the normalized structured data.

Crashes Table

```
1 •  SELECT * FROM Crashes_Table.crashes;
2
```

100% 1:2

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

Name	Distance	PedestrianAction	PrimaryCollisionFactor	CollisionType	ProximityToIntersection	VehicleInvolvedWith
CR-0000027161	0	No Pedestrians Involved	Violation Driver 1	Other	Non-Related	Parked Vehicle
CR-0000028983	0	No Pedestrians Involved	Violation Driver 1	Head On	Non-Related	Other Vehicle
CR-0000028985	0	No Pedestrians Involved	Violation Driver 1	Rear End	Related	Other Vehicle
CR-0000028991	0	No Pedestrians Involved	Violation Driver 1	Rear End	Related	Other Vehicle
CR-0000028997	0	No Pedestrians Involved	Violation Driver 1	Broadside	Intersection	Other Vehicle
CR-0000029004	78	No Pedestrians Involved	Violation Driver 1	Rear End	Non-Related	Other Vehicle
CR-0000029005	36	No Pedestrians Involved	Violation Driver 1	Other	Related	Other Vehicle
CR-0000029006	40	No Pedestrians Involved	Violation Driver 1	Rear End	Related	Other Vehicle
CR-0000029007	0	No Pedestrians Involved	Violation Driver 1	Broadside	Non-Related	Other Vehicle
CR-0000029017	0	No Pedestrians Involved	Violation Driver 1	Broadside	Intersection	Other Vehicle
CR-0000029022	150	No Pedestrians Involved	Violation Driver 1	Rear End	Non-Related	Other Vehicle
CR-0000029026	0	No Pedestrians Involved	Violation Driver 1	Rear End	Intersection	Other Vehicle
CR-0000029032	0	No Pedestrians Involved	Violation Driver 1	Rear End	Related	Other Vehicle

Vehicles Table

```
1 •  SELECT * FROM Crashes_Table.Vehicles;
```

100% 1:1

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

Name	Crashname	Sex	Age	VehicleDamage	PartyCategory	Sobriety	VehicleDirection	MovementPrecedingCollision	ViolationCode
ACV-0000000030	CR-0000063652	M	26	Minor	Driver	Impairment Not Known	East	Proceeding Straight	1
ACV-0000000032	CR-0000064498	F	50	Minor	Driver	Had Not Been Drinking	South	Parking Maneuver	1
ACV-0000000033	CR-0000068721	M	19	Minor	Driver	Had Not Been Drinking	North	Proceeding Straight	1
ACV-0000000039	CR-0000063854	M	76	Major	Driver	Had Not Been Drinking	West	Proceeding Straight	1
ACV-0000000042	CR-0000068989	M	48	Minor	Driver	Impairment Not Known	East	Slowing/Stopping	1
ACV-0000000043	CR-0000069177	M	20	Minor	Driver	Had Not Been Drinking	West	Making Left Turn	1
ACV-0000000044	CR-0000068965	F	18	Moderate	Driver	Had Not Been Drinking	West	Proceeding Straight	1
ACV-0000000045	CR-0000069519	M	26	Major	Driver	Impairment Not Known	South	Proceeding Straight	1
ACV-0000000046	CR-0000065216	F	38	Major	Driver	Had Not Been Drinking	West	Proceeding Straight	1
ACV-0000000047	CR-0000069560	M	45	Minor	Driver	Had Not Been Drinking	East	Slowing/Stopping	1
ACV-0000000048	CR-0000063080	M	62	Moderate	Driver	Had Not Been Drinking	West	Proceeding Straight	1
ACV-0000000049	CR-0000063051	M	38	Major	Driver	Had Not Been Drinking	South	Ran Off Road	1
ACV-0000000050	CR-0000068520	F	20	Moderate	Driver	Had Not Been Drinking	North	Entering Traffic	1

Factors Table

```
1 •   SELECT * FROM Crashes_Table.Factors;
```

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

id	Crashname	Date	Weather	RoadwaySurface	RoadwayCondition	Lighting
► 1	CR-0000029061	2011-01-05	Clear	Dry	No Unusual Conditions	Daylight
2	CR-0000029463	2011-01-21	Clear	Dry	No Unusual Conditions	Daylight
3	CR-0000029283	2011-01-11	Rain	Wet	No Unusual Conditions	Daylight
4	CR-0000027128	2011-02-11	Clear	Dry	No Unusual Conditions	Daylight
5	CR-0000029284	2011-01-11	Cloudy	Dry	No Unusual Conditions	Daylight
6	CR-0000029062	2011-01-05	Clear	Dry	No Unusual Conditions	Daylight
7	CR-0000029285	2011-01-11	Clear	Dry	No Unusual Conditions	Dark - Street Light
8	CR-0000029464	2011-01-21	Clear	Dry	No Unusual Conditions	Dusk - Dawn
9	CR-0000029286	2011-01-11	Cloudy	Wet	No Unusual Conditions	Dark - Street Light
10	CR-0000029063	2011-01-05	Clear	Dry	No Unusual Conditions	Dark - Street Light
11	CR-0000029287	2011-01-11	Clear	Dry	No Unusual Conditions	Dark - Street Light
12	CR-0000027161	2011-02-06	Clear	Dry	No Unusual Conditions	Dark - Street Light
13	CR-0000029064	2011-01-05	Clear	Dry	No Unusual Conditions	Daylight

Laws Table

```
1 •   SELECT * FROM Crashes_Table.laws;
```

Result Grid Filter Rows: Search Edit: Export/Import:

Code	Description
► 21000	Wherever in this division department occurs, it means the Department of the California...
21001	The provisions of this division refer exclusively to the operation of vehicles upon the hig...
2105	Every person riding or driving an animal upon a highway has all of the rights and is subj...
21051	The following sections apply to trolley coaches:
21052	The provisions of this code applicable to the drivers of vehicles upon the highways appl...
21053	This code, except Chapter 1 (commencing with Section 20000) of Division 10, Article 2 (...)
21054	The provisions of this division do not apply to the duly authorized representatives of any...
21055	The driver of an authorized emergency vehicle is exempt from Chapter 2 (commencing...
21056	Section 21055 does not relieve the driver of a vehicle from the duty to drive with due re...
21057	Every police and traffic officer is hereby expressly prohibited from using a siren or drivin...
21058	A physician traveling in response to an emergency call shall be exempt from the provisi...
21059	Sections 21211, 21650, 21660, 22502, 22504, and subdivision (h) of Section 22500 do...
21060	Between the hours of 1 a.m. and 5 a.m., Sections 21650, 21660, 22502, 22504, and su...
21061	(a) In addition to any action prescribed in Division 17 (commencing with Section 40000....
21062	The arresting officer shall, before the end of the next working day, transmit, or cause to...

SQL Statements for the conceptual model:

Crashes Table:

```
CREATE TABLE Crashes_Table.crashes(
Name varchar(50) primary key ,
Distance int,
PedestrianAction varchar(100),
PrimaryCollisionFactor varchar(50),
CollisionType varchar(50),
ProximityToIntersection varchar(20),
VehicleInvolvedWith varchar(50)
);
```

Vehicles Table:

```
CREATE TABLE Crashes_Table.Vehicles(
Name varchar(50) primary key,
Crashname varchar(50),
Sex varchar(10),
Age int,
VehicleDamage varchar(50),
PartyCategory varchar(50),
Sobriety varchar(50),
VehicleDirection varchar(50),
MovementPrecedingCollision varchar(50),
ViolationCode varchar(50),
FOREIGN KEY (Crashname) REFERENCES Crashes(Name)
);
```

Factors Table:

```
CREATE TABLE Crashes_Table.Factors(
id int AUTO_INCREMENT PRIMARY KEY;
Crashname varchar(50),
Date date,
Weather varchar(50),
RoadwaySurface varchar(50),
RoadwayCondition varchar(50),
Lighting varchar(50),
FOREIGN KEY (Crashname) REFERENCES Crashes(Name);
```

Laws Table:

```
CREATE TABLE Crashes_Table.laws (
Code varchar(10) primary key,
Description text
);
```

Views for all Use Cases

1. Use Case: Display the code that has been violated the most along with the description

```
CREATE VIEW Violation_Code AS
SELECT COUNT(ViolationCode) AS Num, Code, Description
FROM Crashes_Table.laws
INNER JOIN Crashes_Table.Vehicles
ON Vehicles.Violationcode = laws.Code
GROUP BY Code
ORDER BY Num DESC
LIMIT 1;
```

2. Use Case: Based on the roadway surface display distance of the crash

```
CREATE VIEW Roadway_Surface AS
SELECT DISTINCT RoadwaySurface, Distance
FROM Crashes_Table.Factors
INNER JOIN Crashes_Table.crashes
ON crashes.Name = Factors.Crashname
ORDER BY RoadwaySurface ASC, Distance;
```

3. Use Case: Display the Age groups involved in accidents between 2021 to present

```
CREATE VIEW Age_Groups AS
SELECT Age, count(Age)
FROM Crashes_Table.Vehicles
LEFT JOIN Crashes_Table.Factors
ON Vehicles.Crashname = Factors.Crashname
WHERE Date BETWEEN '2021-01-01' AND '2022-01-01'
GROUP BY Age
ORDER BY count(Age) DESC
LIMIT 10;
```

4. Use Case: Display CollisionType occurrences and the severity of the accident according to Sobriety of the Driver involved.

```
CREATE VIEW Collision_Type AS
SELECT CollisionType, count(CollisionType), VehicleDamage,
Sobriety
FROM Crashes_Table.crashes
INNER JOIN Crashes_Table.Vehicles
ON crashes.Name = Vehicles.Crashname
GROUP BY CollisionType, VehicleDamage, Sobriety
ORDER BY CollisionType, VehicleDamage ASC;
```

5. Use Case: Display the number of crashes involved considering the weather conditions impact on Roadway conditions

```
CREATE VIEW Weather_Conditions AS
SELECT b.Crashname, count(b.Crashname) as c, a.Weather,
a.RoadwayCondition
FROM Crashes_Table.Factors a
LEFT JOIN Crashes_Table.Vehicles b
ON b.Crashname = a.Crashname
GROUP BY b.Crashname, a.Weather, a.RoadwayCondition
ORDER BY a.RoadwayCondition, a.Weather ASC;
```

6. Use Case: Relationship between Collision Type and Movement PrecedingCollision

```
CREATE VIEW Movement_Preceding_Collision AS
SELECT DISTINCT CollisionType, MovementPrecedingCollision
FROM Crashes_Table.crashes
INNER JOIN Crashes_Table.Vehicles
ON Crashes.Name = Vehicles.Crashname;
```

7. Use Case: Number of crash occurrences according to the lighting conditions

```
CREATE VIEW lighting AS
SELECT f.Lighting, f.Crashname, count(f.Crashname)
FROM Crashes_Table.Factors f
JOIN Crashes_Table.Crashes
ON f.Crashname = Crashes.Name
JOIN Crashes_Table.Vehicles
ON Crashes.Name = Vehicles.Crashname
GROUP BY Lighting, Crashname;
```

8. Use Case: Party involved in the crash and the vehicle responsible for it.

```
CREATE VIEW Party AS
SELECT PartyCategory, VehicleInvolvedWith,
count(PartyCategory)
FROM Crashes_Table.crashes
JOIN Crashes_Table.Vehicles
ON Crashes.Name = Vehicles.Crashname
GROUP BY PartyCategory, VehicleInvolvedWith
ORDER BY PartyCategory ASC;
```

9. Use Case: Display the count of crashes that have involved drunk driving and sober drivers for the past 2 years

```
CREATE VIEW COUNT(CRASHES) AS
SELECT (SELECT count(*) AS sober_crashes
FROM Crashes_Table.crashes c
JOIN Crashes_Table.Vehicles v
JOIN Crashes_Table.Factors f
ON c.Name = v.Crashname
AND f.Crashname = v.Crashname
WHERE f.Date BETWEEN '2021-01-01' AND '2022-01-01'
AND v.Sobriety LIKE '%Had Not Been Drinking%'),
(SELECT COUNT(*) AS drunk_crashes
FROM Crashes_Table.crashes c
JOIN Crashes_Table.Vehicles v
JOIN Crashes_Table.Factors f
ON f.Crashname = v.Crashname
AND c.name = v.Crashname
WHERE f.Date BETWEEN '2021-01-01' AND '2022-01-01'
AND v.Sobriety LIKE '%Had Been Drinking%')
FROM dual;
```

10. Use Case: Relationship between Pedestrian activity before collision and Movement preceding collision

```
CREATE VIEW Pedestrian_activity AS
SELECT PedestrianAction, MovementPrecedingCollision
FROM Crashes_Table.crashes
JOIN Crashes_Table.Vehicles
ON crashes.Name = Vehicles.Crashname
GROUP BY PedestrianAction, MovementPrecedingCollision
ORDER BY PedestrianAction ASC, MovementPrecedingCollision ASC;
```

USE - CASES

1. **Use Case:** Display the code that has been violated the most along with the description

Description: Displays the traffic code violated the most.

Actor: Person

Steps:

Post Condition: Successfully displays the the code

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect

2. **Use Case:** Based on the roadway surface display distance of the crash

Description: Distance of the crash based on roadway surface

Actor: Person

Steps:

Post Condition: Successfully displays the crash distance based on roadway surface

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect

3. **Use Case:** Display the Age groups involved in accidents between 2021 to present

Description: Age groups involved in accidents between 2021 to present.

Actor: Person

Steps:

Post Condition: Successfully displays list of Age groups.

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect

4. **Use Case:** Display CollisionType occurrences and the severity of the accident according to Sobriety of the Driver involved.

Description: CollisionType occurrences and the severity of the accident according to Sobriety of the Driver involved.

Actor: Person

Steps:

Post Condition: Successfully displays the expected result

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect

5. **Use Case:** Display the number of crashes involved considering the weather conditions impact on Roadway conditions

Description: Crashes involved considering the weather conditions impact on Roadway conditions

Actor: Person

Steps:

Post Condition: Successfully displays the Crashes involved considering the weather conditions and the impact on Roadway conditions

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect
6. **Use Case:** Relationship between Collision Type and Movement PrecedingCollision

Description: Displays the relationship between Collision Type and Movement PrecedingCollision

Actor: Person

Steps:

Post Condition: Successfully displays the source details

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect
7. **Use Case:** Number of crash occurrences according to the lighting conditions

Description: crash occurrences according to the lighting conditions

Actor: Person

Steps:

Post Condition: Successfully displays the source details

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect
8. **Use Case:** Party involved in the crash and the vehicle responsible for it.

Description: PartyCategory involved in crash and the vehicle responsible for it.

Actor: Person

Steps:

Post Condition: Successfully displays the active source details.

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect.
9. **Use Case:** Display the count of crashes that have involved drunk driving and sober drivers for the past 2 years

Description: Count of crashes that have involved drunk driving and sober drivers.

Actor: Person

Steps:

Post Condition: Successfully displays the active source details.

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect

10. Use Case: Relationship between Pedestrian activity before collision and Movement preceding

Description: Displays the Relationship between Pedestrian activity before collision and Movement preceding

Actor: Person

Steps:

Post Condition: Successfully displays the active source details.

Alternate Path: The request is not correct, and system throws an error.

Error: Information is incorrect

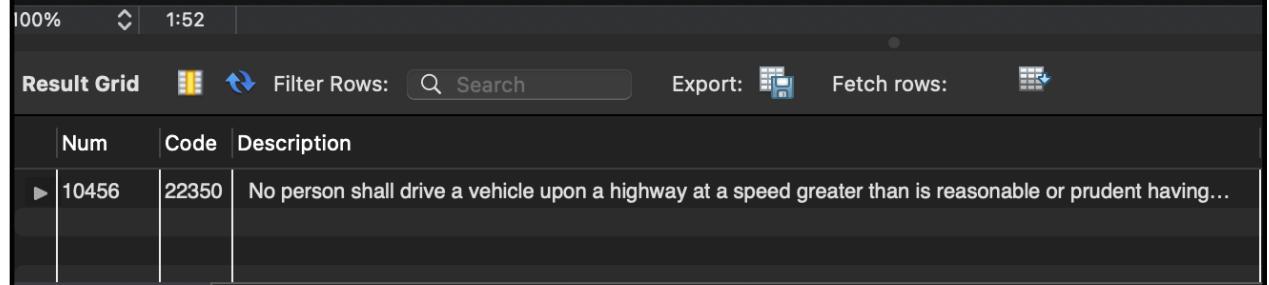
SQL STATEMENTS

1. Use Case: Display the code that has been violated the most along with the description

```
SELECT COUNT(ViolationCode) AS Num, Code, Description
FROM Crashes_Table.laws
INNER JOIN Crashes_Table.Vehicles
ON Vehicles.Violationcode = laws.Code
GROUP BY Code
ORDER BY Num DESC
LIMIT 1;
```

Query Output

```
45 •  SELECT COUNT(ViolationCode) AS Num, Code, Description
46   FROM Crashes_Table.laws
47     INNER JOIN Crashes_Table.Vehicles
48       ON Vehicles.Violationcode = laws.Code
49         GROUP BY Code
50           ORDER BY Num DESC
51             LIMIT 1;
```



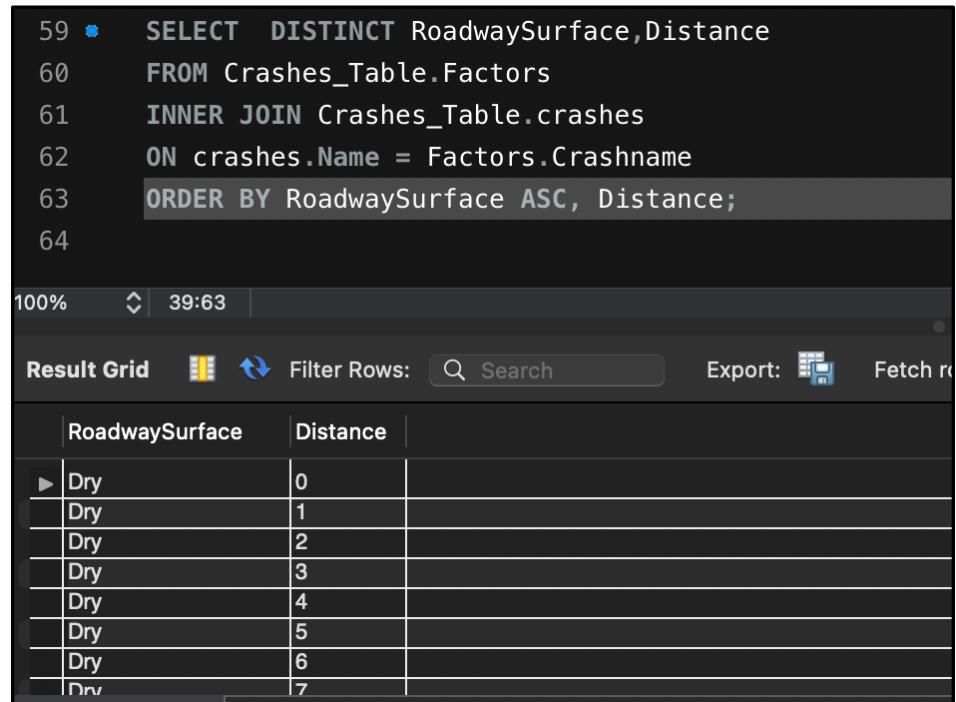
The screenshot shows a SQL query editor interface. At the top, there is a code editor window displaying the SQL query with line numbers 45 through 52. Below the code editor is a toolbar with various icons for file operations, search, and export. The main area shows the results of the query in a grid format. The grid has three columns: 'Num', 'Code', and 'Description'. There is one row of data: Num is 10456, Code is 22350, and Description is 'No person shall drive a vehicle upon a highway at a speed greater than is reasonable or prudent having...'. The bottom right corner of the grid has a small ellipsis (...).

Num	Code	Description
10456	22350	No person shall drive a vehicle upon a highway at a speed greater than is reasonable or prudent having...

2. Use Case: Based on the roadway surface display distance of the crash

```
SELECT DISTINCT RoadwaySurface,Distance  
FROM Crashes_Table.Factors  
INNER JOIN Crashes_Table.crashes  
ON crashes.Name = Factors.Crashname  
ORDER BY RoadwaySurface ASC, Distance;
```

Query Output



The screenshot shows a database query interface with the following details:

- Query Text:

```
59 •  SELECT DISTINCT RoadwaySurface,Distance  
60      FROM Crashes_Table.Factors  
61      INNER JOIN Crashes_Table.crashes  
62      ON crashes.Name = Factors.Crashname  
63      ORDER BY RoadwaySurface ASC, Distance;  
64
```
- UI Elements:
 - Zoom: 100%
 - Time: 39:63
 - Result Grid: Selected
 - Filter Rows: Available
 - Search: Available
 - Export: Available
 - Fetch rows: Available
- Result Grid Data:

RoadwaySurface	Distance
Dry	0
Dry	1
Dry	2
Dry	3
Dry	4
Dry	5
Dry	6
Dry	7

3. Use Case: Display the Age groups involved in accidents between 2021 to present

```
SELECT Age, count(Age)
FROM Crashes_Table.Vehicles
LEFT JOIN Crashes_Table.Factors
ON Vehicles.Crashname = Factors.Crashname
WHERE Date BETWEEN '2021-01-01' AND '2022-01-01'
GROUP BY Age
ORDER BY count(Age) DESC
LIMIT 10;
```

Query output

	Age	count(Age)	
▶	26	661	
	35	229	
	22	124	
	29	122	
	21	119	
	23	117	
	27	116	
	25	113	
	24	109	
	32	105	

4. Use Case: Display CollisionType occurrences and the severity of the accident according to Sobriety of the Driver involved.

```
SELECT CollisionType, count(CollisionType), VehicleDamage,
Sobriety
FROM Crashes_Table.crashes
INNER JOIN Crashes_Table.Vehicles
ON crashes.Name = Vehicles.Crashname
GROUP BY CollisionType, VehicleDamage, Sobriety
ORDER BY CollisionType, VehicleDamage ASC;
```

Query Output

	CollisionType	count(CollisionTy...)	VehicleDamage	Sobriety
▶	Broadside	42	Major	Had Been Drinking - Impairment Unknown
	Broadside	65	Major	Had Been Drinking - Not Under Influence
	Broadside	283	Major	Had Been Drinking - Under Influence
	Broadside	3801	Major	Had Not Been Drinking
	Broadside	1106	Major	Impairment Not Known
	Broadside	8	Major	Impairment Physical
	Broadside	7	Major	Sleepy/Fatigued
	Broadside	8	Major	Under Drug Influence
	Broadside	18	Minor	Had Been Drinking - Impairment Unknown
	Broadside	38	Minor	Had Been Drinking - Not Under Influence
	Broadside	53	Minor	Had Been Drinking - Under Influence
	Broadside	1799	Minor	Had Not Been Drinking
	Broadside	491	Minor	Impairment Not Known
	Broadside	4	Minor	Impairment Physical
	Broadside	3	Minor	Sleepy/Fatigued
	Broadside	4	Minor	Under Drug Influence
	Broadside	30	Moderate	Had Been Drinking - Impairment Unknown
	Broadside	78	Moderate	Had Been Drinking - Not Under Influence
	Broadside	165	Moderate	Had Been Drinking - Under Influence
	Broadside	4154	Moderate	Had Not Been Drinking
	Broadside	986	Moderate	Impairment Not Known
	Broadside	1	Moderate	Impairment Physical
	Broadside	6	Moderate	Sleepy/Fatigued

5. Use Case: Display the number of crashes involved considering the weather conditions impact on Roadway conditions

```
SELECT b.Crashname, count(b.Crashname) as c, a.Weather,
a.RoadwayCondition
FROM Crashes_Table.Factors a
LEFT JOIN Crashes_Table.Vehicles b
ON b.Crashname = a.Crashname
GROUP BY b.Crashname, a.Weather, a.RoadwayCondition
ORDER BY a.RoadwayCondition, a.Weather ASC;
```

Query Output

	Crashname	c	Weather	RoadwayCondition
▶	CR-0000093902	2	Clear	Construction - Repair Zone
	CR-0000094414	2	Clear	Construction - Repair Zone
	CR-0000093776	2	Clear	Construction - Repair Zone
	CR-0000093748	2	Clear	Construction - Repair Zone
	CR-0000092626	1	Clear	Construction - Repair Zone
	CR-0000092369	1	Clear	Construction - Repair Zone
	CR-0000092051	3	Clear	Construction - Repair Zone
	CR-0000091724	1	Clear	Construction - Repair Zone
	CR-0000089218	2	Clear	Construction - Repair Zone
	CR-0000088720	1	Clear	Construction - Repair Zone
	CR-0000088324	1	Clear	Construction - Repair Zone
	CR-0000084482	2	Clear	Construction - Repair Zone
	CR-0000084908	1	Clear	Construction - Repair Zone

6. Use Case: Relationship between Collision Type and Movement PrecedingCollision

```
SELECT DISTINCT CollisionType, MovementPrecedingCollision
FROM Crashes_Table.crashes
INNER JOIN Crashes_Table.Vehicles
ON Crashes.Name = Vehicles.Crashname;
```

Query Output

CollisionType	MovementPrecedingCollision
► Other	Proceeding Straight
Head On	Other Unsafe Turning
Rear End	Proceeding Straight
Broadside	Proceeding Straight
Broadside	Making Left Turn
Rear End	Slowing/Stopping
Other	Slowing/Stopping
Broadside	Other Unsafe Turning
Broadside	Making Right Turn
Head On	Proceeding Straight
Other	Merging
Broadside	Making U-Turn
Other	Changing Lanes
Other	Ran Off Road
Head On	Making Left Turn
Other	Other Unsafe Turning
Other	Making Right Turn
Other	Entering Traffic
Other	Making U-Turn
Sideswipe	Changing Lanes
Other	Passing Other Vehicles
Rear End	Making Right Turn
Other	Backing

7. Use Case: Number of crash occurrences according to the lighting conditions

```
SELECT f.Lighting, f.Crashname, count(f.Crashname)
FROM Crashes_Table.Factors f
JOIN Crashes_Table.Crashes
ON f.Crashname = Crashes.Name
JOIN Crashes_Table.Vehicles
ON Crashes.Name = Vehicles.Crashname
GROUP BY Lighting, Crashname;
```

Query Output

Lighting	Crashname	count(f.Crashname)
Daylight	CR-0000029283	1
Dark - Street Light	CR-0000029063	1
Dark - Street Light	CR-0000027161	1
Daylight	CR-0000029064	1
Dark - Street Light	CR-0000029290	1
Daylight	CR-0000029468	1
Daylight	CR-0000028983	1
► Dark - Street Light	CR-0000028985	1
Dark - Street Light	CR-0000029078	1
Dark - Street Light	CR-0000029079	1
Daylight	CR-0000029304	1
Daylight	CR-0000029306	1
Dark - Street Light	CR-0000028991	1
Dark - Street Light	CR-0000028997	1
Daylight	CR-0000029310	1
Daylight	CR-0000029311	1
Daylight	CR-0000029486	1
Daylight	CR-0000029089	1
Daylight	CR-0000029315	1
Daylight	CR-0000029004	1
Dark - Street Light	CR-0000029495	1
Dark - Street Light	CR-0000029496	1
Dusk - Dawn	CR-0000029326	2

8. Use Case: Party involved in the crash and the vehicle responsible for it.

```
SELECT PartyCategory, VehicleInvolvedWith,  
count(PartyCategory)  
FROM Crashes_Table.crashes  
JOIN Crashes_Table.Vehicles  
ON Crashes.Name = Vehicles.Crashname  
GROUP BY PartyCategory, VehicleInvolvedWith  
ORDER BY PartyCategory ASC;
```

Query Output

PartyCategory	VehicleInvolvedWith	count(PartyCategory)
Bicycle	Bike	1085
Bicycle	Fixed Object	2
Bicycle	Other Object	1
Bicycle	Other Vehicle	21
Bicycle	Parked Vehicle	5
Bicycle	Pedestrian	5
Bicycle	Scooter Motorized	74
Bicycle	Train	1
Driver	Animal	13
Driver	Bike	1130
Driver	Fixed Object	4843
Driver	Ice Cream Truck	2
Driver	Light Rail Vehicle	73

9. Use Case: Display the count of crashes that have involved drunk driving and sober drivers for the past 2 years

```
SELECT (SELECT count(*) AS sober_crashes
FROM Crashes_Table.crashes c
JOIN Crashes_Table.Vehicles v
JOIN Crashes_Table.Factors f
ON c.Name = v.Crashname
AND f.Crashname = v.Crashname
WHERE f.Date BETWEEN '2021-01-01' AND '2022-01-01'
AND v.Sobriety LIKE '%Had Not Been Drinking%'),
(SELECT COUNT(*) AS drunk_crashes
FROM Crashes_Table.crashes c
JOIN Crashes_Table.Vehicles v
JOIN Crashes_Table.Factors f
ON f.Crashname = v.Crashname
AND c.name = v.Crashname
WHERE f.Date BETWEEN '2021-01-01' AND '2022-01-01'
AND v.Sobriety LIKE '%Had Been Drinking%')
FROM dual;
```

Query Output

(SELECT count(*) AS sober_crashes FROM Crashes_Table.crashes c	(SELECT COUNT(*) AS drunk_crashes FROM Crashes_Table.crashes c
► 2660	446

10. Use Case: Relationship between Pedestrian activity before collision and Movement preceding collision

```
SELECT PedestrianAction, MovementPrecedingCollision
FROM Crashes_Table.crashes
JOIN Crashes_Table.Vehicles
ON crashes.Name = Vehicles.Crashname
GROUP BY PedestrianAction, MovementPrecedingCollision
ORDER BY PedestrianAction ASC, MovementPrecedingCollision
ASC;
```

Query Output

PedestrianAction	MovementPrecedingCollision
Crossing - Not In Crosswalk	Backing
Crossing - Not In Crosswalk	Changing Lanes
Crossing - Not In Crosswalk	Crossing Into Opposing Lane
Crossing - Not In Crosswalk	Entering Traffic
Crossing - Not In Crosswalk	Making Left Turn
Crossing - Not In Crosswalk	Making Right Turn
Crossing - Not In Crosswalk	Merging
Crossing - Not In Crosswalk	Other (Bike)
Crossing - Not In Crosswalk	Other (Ped)
Crossing - Not In Crosswalk	Other Unsafe Turning
Crossing - Not In Crosswalk	Parked
Crossing - Not In Crosswalk	Proceeding Straight
Crossing - Not In Crosswalk	Ran Off Road
Crossing - Not In Crosswalk	Stopped
Crossing - Not In Crosswalk	Traveling Wrong Way
Crossing In Crosswalk - At...	Backing
Crossing In Crosswalk - At...	Entering Traffic
Crossing In Crosswalk - At...	Making Left Turn
Crossing In Crosswalk - At...	Making Right Turn
Crossing In Crosswalk - At...	Making U-Turn
Crossing In Crosswalk - At...	Merging
Crossing In Crosswalk - At...	Other (Bike)
Crossing In Crosswalk - At...	Other (Ped)

Git Hub link

https://github.com/sonalibandi/DMDD_Final_Project/