ASSIGNMENT 3

The objective of this project is to build a 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.

**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.

**Brief description of files uploaded in GitHub**

Data Cleaning Scripts

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.

**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(

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

);

**USE-CASE**

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

1. **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

1. **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

1. **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

1. **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

1. **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

1. **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

1. **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.

1. **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

1. **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;

1. 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;

1. 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;

1. 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;

1. 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;

1. 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;

1. 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;

1. 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;

1. 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;

1. 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;