

## DATASET:

Link to dataset: <https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents>

- This dataset on car accidents in the USA includes data from all 49 states. Data on accidents are gathered between February 2016 and December 2021.
- The dataset contains 47 columns with different data types, including integer, float, varchar, and Boolean, and more than 2.8 million records. The organized dataset is 1.15 GB in size.

## DESIGN A DATABASE:

### 1. SCHEMA DESIGN:

- a. Find entities, their attributes, their primary keys, and relationships between them.

ENTITY: ACCIDENT	
COLUMN NAME	DATA TYPE
ID	VARCHAR
Severity	INT
Start_Time	DATETIME
End_Time	DATETIME
Description	VARCHAR
Location_id	INT
ENTITY: LOCATION	
COLUMN NAME	DATA TYPE
Location_id	INT
City	VARCHAR
State	VARCHAR
County	VARCHAR
Zipcode	VARCHAR
Start_Lat	FLOAT
Start_Lng	FLOAT
End_Lat	FLOAT
End_Lng	FLOAT
ENTITY: WEATHER	
COLUMN NAME	DATA TYPE
Location_id	INT
Weather_Timestamp	DATETIME
Temperature(F)	FLOAT
Humidity (%)	FLOAT
Pressure(in)	FLOAT
Weather_Condition	VARCHAR
ID	VARCHAR

#### i. Accident table

- Entity: Accident
- Attributes: ID, Severity, Start\_Time, End\_Time, Description, Location\_id (foreign key from the Location table)
- Primary key: ID
- Relationships:
  - A foreign key relation between the accident table and Location table using location\_id.

**ii. Location table**

- Entity: Location
- Attributes: Location\_id, City, State, County, Zipcode, Start\_Lat, Start\_Lng, End\_Lat, End\_Lng
- Primary key: Location\_id
- Relationships:
  - A foreign key relation between Accident table and Location table using Location\_id.
  - A foreign key relation between the Weather table and Location table using Location\_id.

**iii. Weather table**

- Entity: Weather
- Attributes: Location\_id, Weather\_Timestamp, Temperature(F), Humidity (%), Pressure(in), Weather\_Condition, ID (foreign key from the Accident table)
- Primary key: Location\_id, Weather\_Timestamp
- Relationships:
  - A foreign key relation between Weather table and Location table using Location\_id.
  - A foreign key relation between Weather table and Accident table using ID.

**b. Model all the constraints you believe should be there in your schema.**

**i. Key Constraints:**

- The Accident table should have a primary key constraint on the ID attribute.
- The Weather table should have a primary key constraint on the Location\_Id and Weather\_timestamp attributes.
- The Location table should have a primary key constraint on the Location\_Id attribute.

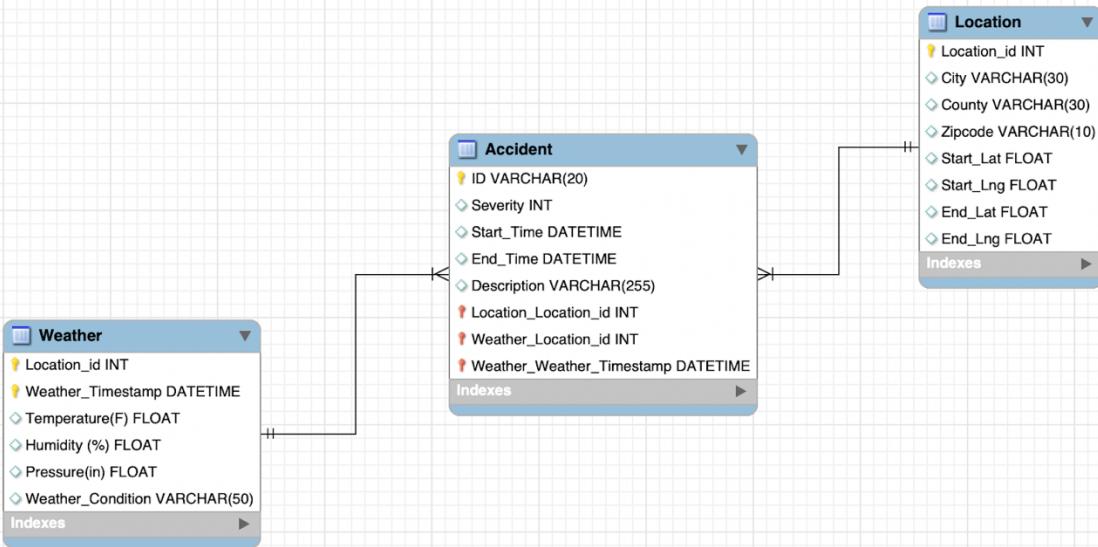
**ii. Single-Value Constraints:**

- The Start\_Time, End\_Time, and Description attributes in the Accident table should be single-valued (i.e., there should be only one value for each ID).
- The County and Zipcode combined attributes in the Location table should be single-valued (i.e., there should be only one value for each Location\_id).

**iii. Referential Constraints:**

- The Location\_Id attribute in the Accident table should have a referential constraint that references the Location\_Id primary key attribute in the Location table.
- The ID attribute in the Weather table should have a referential constraint that references the ID primary key attribute in the Accident table.
- The Location\_Id attribute in the Weather table should have a referential constraint that references the Location\_id primary key attribute in the Location table.

c. Draw the ER diagram of your dataset.



d. Translate your ER diagram into relations.

i. Accident table:

- **One-to-Many Relationship:** While an accident can be linked to a single location, a location can be linked to several accidents. This is because the Location\_id column in the Accident table is a foreign key that refers to the Location table's primary key Location\_id. Because the Location\_id field in the Accident database may only have one value per record, the Location table and the Accident table have a one-to-many connection.
- As a result, the foreign key constraint Location\_id establishes a one-to-many relation between the Location and Accident table.

ii. Weather table:

- The Accident and Weather tables have a one-to-many relationship, which means that one accident might have numerous weather data connected with it. This is because the Accident table's primary key (ID) is referenced as a foreign key in the Weather table, and the Weather table's primary key (Location\_id, Weather\_Timestamp) uniquely identifies each weather report connected with an accident. As a result, whereas one accident might be related with several weather records, each weather record can only be associated with one accident.

## 2. SCHEMA NORMALIZATION:

a. Find all the functional dependencies you can from your schema and list them.

iii. Functional dependencies

- {Zipcode} → {City, State, County}
- {State} → {County}
- {County, City} → {State, Zipcode}
- {Start\_Lat, Start\_Lng, End\_Lat, End\_Lng} → {City, County, State, Zipcode}
- {ID} → {Severity, Start\_Time, End\_Time, Description, Location\_id}

**b. Check if the keys you have chosen for your relations are minimal (prove)**

i. **Accident table:**

- Primary key: ID
- Foreign key: Location\_id
- There are no functional dependencies mentioned in the Accident table, and the primary key ID uniquely identifies each accident. Therefore, the primary key is minimal.

ii. **Location table:**

- Primary key: Location\_id
- Foreign key: Location\_id (from Accident table), Location\_id (from Weather table)
- The given functional dependencies in the Location table are as follows:
  - {Zipcode}  $\rightarrow$  {City, State, County}
  - {State}  $\rightarrow$  {County}
  - {County, City}  $\rightarrow$  {State, Zipcode}
- In the given functional dependencies, the attribute 'Zipcode' is not functionally dependent on any other attribute, and therefore, it can be a candidate key. The primary key 'Location\_id' is also a candidate key, and it is minimal because no other attribute can be removed from it without affecting the functional dependencies.

iii. **Weather table:**

- Primary key: (Location\_id, Weather\_Timestamp)
- Foreign key: Location\_id (from Location table), ID (from Accident table)
- The given functional dependencies in the Weather table are as follows:
  - {Start\_Lat, Start\_Lng, End\_Lat, End\_Lng}  $\rightarrow$  {City, County, State, Zipcode}
- The primary key (Location\_id, Weather\_Timestamp) uniquely identifies each row in the Weather table, and no other attribute can be removed from it without affecting the functional dependencies. Therefore, the primary key is minimal.
- In conclusion, the keys chosen for the given relations are all minimal, and no attribute can be removed from them without affecting the functional dependencies.

**c. Check if your schema is in BCNF (Boyce-Codd Normal Form) (prove)**

- The tables are in first normal form because every column is atomic, and every value recorded in each column is one value.
- Because there is no partial dependence, the table also meets the second normal form.
- Transitive Dependency: The functional dependency {Zipcode}  $\rightarrow$  {City, State, County} implies a transitive dependency between the non-prime attribute 'Zipcode' and the other non-prime attributes {City, State, County}. This transitive dependency violates the 3NF, which requires that each non-prime attribute should be functionally dependent on the primary key only, and not on any other non-prime attribute. Therefore, it fails to meet the third normal form requirement.

**d. If your schema violates BCNF, bring it to BCNF by decomposing it (show)**

location_id	City	State	County	start_lat	start_lng	end_lat	end_lng
-------------	------	-------	--------	-----------	-----------	---------	---------

City	County	Zip code	Location id
------	--------	----------	-------------

City	State
------	-------

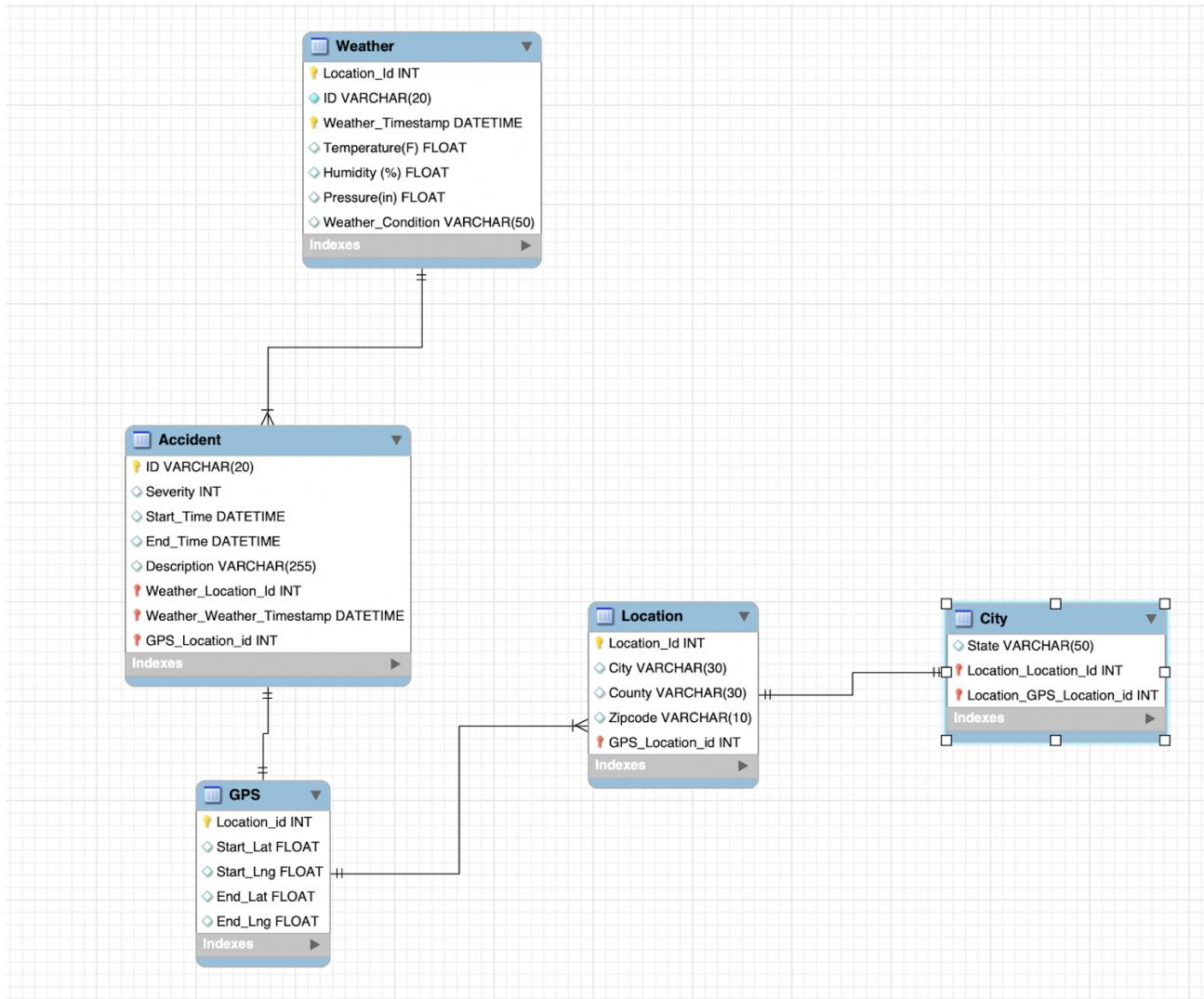
start_lat	start_long	end_lat	end_lat
-----------	------------	---------	---------

All non-key attributes are now completely functional, as can be seen in both tables, with the primary key only needing to satisfy the third normal form. A table must meet the two requirements listed below to satisfy the Boyce-Codd Normal Form:

- It must be in the Third Normal Form,
- for any  $A \rightarrow B$  dependency.,A must be a super key

The predicate is true for our schema. Our schema is therefore in BCNF.

**e. Update your ER diagram with the latest schema.**



**3. Create your database in MySQL using the latest version of your schema.**

In MySQL, I successfully built a new database called DB and imported a dataset as a table called "P".

```
CREATE DATABASE DB;
```

Then I begin querying the table's data with SQL statements.

To demonstrate that the table was created, I used the following SQL statement to retrieve all the data from the "Project" table:

```
SELECT * FROM Project;
```

**4. Import the data into your database.**

**a. If there are errors while importing, document these errors in your report and mention how you dealt with them.**

- i. Error Code: 1290. The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

Reason: The variable secure\_file\_priv is used to limit the effect of data import and export operations and these operations are allowed only to users who have the FILE privilege.

Fix: We may use SHOW VARIABLES LIKE "secure\_file\_priv"; to see the directory that has been configured. We can fix this by moving the file to the directory specified by secure-file-priv.

```
LOAD DATA INFILE '/Users/HarikaManapuram/Desktop/project.csv'  
INTO TABLE project  
FIELDS TERMINATED BY ','  
ENCLOSED BY '\"'  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

- ii. Error Code: 1062. Duplicate entry'

Fix: adding 'Ignore' while inserting data into table resolved the issue for us

- iii. Error Code: 2013. Lost connection while loading data to MySQL server during query

Reason: This error appears when the connection between your MySQL client and database server times out. Essentially, it took too long for the query to return data, so the connection gets dropped

Fix: We can increase your MySQL client's timeout values by editing the SQL Editor preferences in MySQL Workbench:

- In the application menu, select Edit > Preferences > SQL Editor.
- Look for the MySQL Session section and increase the DBMS connection read time out value.
- Save the settings, quite MySQL Workbench and reopen the connection.

We have changed it to 9000 seconds.

## DATA CLEANING AND DATABASE TESTING:

1. For each table in your database, check all the columns and the values they contain.

```
13 •  DESCRIBE p;
14
15
16
```

100% 1:13 | 2 errors found

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
ID	text	YES		NULL	
severity	float	YES		NULL	
Start_Time	text	YES		NULL	
End_Time	text	YES		NULL	
Description	text	YES		NULL	
City	text	YES		NULL	
State	text	YES		NULL	
country	text	YES		NULL	
Zipcode	float	YES		NULL	
Start_Lat	double	YES		NULL	
Start_Lng	double	YES		NULL	
End_Lat	double	YES		NULL	
End_Lng	double	YES		NULL	
Weather_Timestamp	text	YES		NULL	
Weather_Condition	text	YES		NULL	
Temperature	float	YES		NULL	
Humidity	float	YES		NULL	

Result 157

2. For numeric columns, provide all the statistics, check them, and see what you find.

- a. Check them against the information you found in step 3 of project 1.

```
16 •  SELECT
17      AVG(pressure) AS mean,
18      STDDEV(pressure) AS standard_deviation,
19      MAX(pressure) AS maximum_value,
20      MIN(pressure) AS minimum_value
21  FROM
22    p;
23
```

100% 5:22 | 3 errors found

Result Grid Filter Rows: Search Export:

mean	standard_deviation	maximum_value	minimum_value
30.034608729986562	0.22682138527076515	30.92	20.41

- b. You should be looking for missing values, values that seem to be outliers (typically far away from the mean), or data errors or any values that does not seem to be valid (like a typo)

```

25 •   SELECT ID,Severity,Start_Time,End_Time,Description,
26   City,State,County,Zipcode,Start_Lat,Start_Lng,End_Lat,
27   End_Lng,Weather_Condition,`Temperature.F.`, `Humidity...`, `Pressure.in.`,Location_Id
28   FROM project;
29

```

Result Grid | Filter Rows: | Search | Export: | Fetch rows: | Result Grid | Form Editor | Field Types | Query Stats | Read Only

ID	Severity	Start_Time	End_Time	Description	City	State	county	Zipcode	Start_Lat	Start_Lng
A-1	2	2016-02-08 06:15:39	2016-02-08 12:15:39	At I-71/US-50/Exit 1 - Accident.	Dayton	OH	Montgomery	45421	39.98812	-84.03630
A-3	2	2016-02-08 06:51:45	2016-02-08 12:51:45	At Dart Ave/Exit 21 - Accident.	Cincinnati	OH	Hamilton	45203	39.10266	-84.52461
A-4	2	2016-02-08 07:53:43	2016-02-08 13:53:43	At Mitchell Ave/Exit 6 - Accident.	Akron	OH	Summit	44311	41.06213	-81.53787
A-5	3	2016-02-08 08:16:57	2016-02-08 14:16:57	At Dela Palma Rd - Accident.	Cincinnati	OH	Hamilton	45217	39.172393	-84.49275
A-6	2	2016-02-08 08:15:41	2016-02-08 14:15:41	At OH-4/Exit 54 - Accident.	Williamsburg	OH	Clermont	45176	39.06324	-84.03245
A-7	2	2016-02-08 09:51:46	2016-02-08 17:51:46	At Bagley Rd/Exit 235 - Accident.	Dayton	OH	Montgomery	45404	39.77565	-84.18603
A-8	2	2016-02-08 14:19:57	2016-02-08 20:19:57	At OH-65/Exit 122 - Accident.	Cleveland	OH	Cuyahoga	44130	41.37531	-81.82011
A-9	2	2016-02-08 15:16:43	2016-02-08 21:16:43	At I-71/Exit 26 - Accident.	Lima	OH	Allen	45806	40.702247	-84.07581
A-10	2	2016-02-08 16:50:57	2016-02-08 22:50:57	At OH-4/Paddock Rd/Exit 9 - Accident.	Westerville	OH	Franklin	43081	40.10931	-82.96843
A-11	2	2016-02-08 17:27:39	2016-02-08 23:27:39	At US-42/Exit 170 - Accident.	Cincinnati	OH	Hamilton	45216	39.19288	-84.47722
A-12	2	2016-02-08 18:30:18	2016-02-08 23:30:18	Between OH-72/Exit 58 and US-35/Exit 65 - Acc...	Cincinnati	OH	Hamilton	45225	39.13877	-84.53391
A-13	2	2016-02-08 18:11:11	2016-02-09 00:11:11	At Shipley Rd - Accident.	Cleveland	OH	Cuyahoga	44113	41.4739	-81.70422
A-14	3	2016-02-08 19:47:42	2016-02-09 01:47:42	At OH-16/Broad St/Exit 39 - Accident.	Jamestown	OH	Greene	45335	39.582242	-83.6778
A-15	2	2016-02-08 20:13:22	2016-02-09 02:13:22	Between Weber Rd/Exit 113 and Hudson St/Exit...	Freepo	OH	Guernsey	43973	40.151785	-81.31265
A-16	3	2016-02-08 21:00:17	2016-02-09 02:00:17	Closed between I-75/Exit 204 and US-24/Dale...	Columbus	OH	Franklin	43213	39.97241	-82.8469
A-17	2	2016-02-08 21:47:42	2016-02-09 01:47:42	At I-270 - Accident.	Columbus	OH	Franklin	43213	39.9838	-82.8565
A-18	3	2016-02-08 22:00:17	2016-02-09 02:00:17	Between Weber Rd/Exit 113 and Hudson St/Exit...	Columbus	OH	Franklin	43224	40.02664	-82.9944
A-19	4	2016-02-08 22:00:17	2016-02-09 02:00:17	Closed between I-75/Exit 204 and US-24/Dale...	Toledo	OH	Lucas	43610	41.670261	-82.5720

We created a new table named "P" using the required columns from the table "Project".

- c. Make sure all the values of these columns are from the same type (all numeric)

```

90      ALTER TABLE p
91      modify COLUMN Severity float;
92 •   ALTER TABLE p
93      modify COLUMN start_lat float;
94 •   ALTER TABLE p
95      modify COLUMN start_lng float;
96 •   ALTER TABLE p
97      modify COLUMN end_lat float;
98 •   ALTER TABLE p
99      modify COLUMN end_lng float;
100 •  ALTER TABLE p
101     modify COLUMN Temperature float;

```

100% | 15:88 | 2 errors found

Result Grid | Filter Rows: | Search | Export: | Read Only

Field	Type	Null	Key	Default	Extra
Severity	float	YES		NULL	
Start_Time	text	YES		NULL	
End_Time	text	YES		NULL	
Description	text	YES		NULL	
City	text	YES		NULL	
State	text	YES		NULL	
county	text	YES		NULL	
Zipcode	float	YES		NULL	
start_lat	float	YES		NULL	
start_lng	float	YES		NULL	
end_lat	float	YES		NULL	
end_lng	float	YES		NULL	
Weather_Timestamp	text	YES		NULL	
Weather_Condition	text	YES		NULL	
Temperature	float	YES		NULL	
Humidity	float	YES		NULL	
Pressure	float	YES		NULL	

Result 171

**d. Document the problems you find; fix them and explain how you dealt with them.**

Missing values: We dealt with missing values by deleting the rows with missing values if the number of missing values is small.

**3. For character columns, provide frequency table of some of the values they contain, then do a quick check on these values.**

**a. Check them against the information you found in step 3 of project 1.**

→ Frequency of city

Miami	Los Angeles	Orlando	Dallas	Houston
106966	68956	54691	41979	39448

→ Frequency of State

State	CA	FL	TX	OR	VA
Frequency	795868	401388	149037	126341	113535

→ Frequency of Country

Country	Los Angeles	Miami-Dade	Orange	San Bernardino	Dallas
Frequency	234122	143939	114917	55018	50050

→ Frequency of Zip Code

ZipCode	91761	33186	92407	92507	91706
Frequency	6162	5248	4528	4527	4471

**b. You should be looking for missing values or data errors or values that does not seem to be valid (e.g., sometimes there are white spaces in some of the cells either before or after the value)**

**c. Make sure all the values are from the same type and domain.**

```
68 • ALTER TABLE p
69   RENAME COLUMN `temperature.f.` TO temperature;
70 • ALTER TABLE p
71   RENAME COLUMN `humidity...` TO humidity;
72 • ALTER TABLE p
73   RENAME COLUMN `pressure.in` TO pressure;
```

100% 41:73 2 errors found

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
start_time	text	YES	NONE		
End_Time	text	YES	NONE		
Description	text	YES	NONE		
City	text	YES	NONE		
State	text	YES	NONE		
country	text	YES	NONE		
Zipcode	float	YES	NONE		
start_lat	float	YES	NONE		
start_lng	float	YES	NONE		
end_lat	float	YES	NONE		
end_lng	float	YES	NONE		
Weather_Timestamp	text	YES	NONE		
Weather_Condition	text	YES	NONE		
Temperature	float	YES	NONE		
Humidity	float	YES	NONE		
Pressure	float	YES	NONE		
Location_id	float	YES	NONE		

- There were no missing values as they were handled while cleaning the data.
- There were no white spaces in some of the cells either before or after the value.

**d. Document the problems you find, fix them, and explain how you dealt with them.**

```

76 • CREATE TABLE Accident AS
77   SELECT id ,severity, start_time, end_time, description, location_id
78   FROM P;
79
80 • CREATE TABLE location
81   SELECT Location_id , city, county, zipcode
82   FROM P;
83
84 • CREATE TABLE Weather AS
85   SELECT Location_id, weather_timestamp, temperature, humidity, pressure, weather_condition, ID
86   FROM P;
87
88 • CREATE TABLE GPS AS
89   SELECT Start_Lat,Start_Lng,End_Lat,End_Lng,Location_Id
90   FROM P;
91
92 • CREATE TABLE City AS
93   SELECT City,State
94   FROM P;
95

```

100% 1:81 | 2 errors found

Action	Output	Time	Action	Response	Duration / Fetch Time
CREATE TABLE Accident AS SELECT id ,severity, start_time, end_time, description, location_id FROM P;	420	00:00:40	CREATE TABLE ACCIDENT AS SELECT ID ,SEVERITY, START_TIME, END_TIME, DESCRIPTION, LOCATION_ID FROM P;	75843 row(s) affected Records: 75843 Duplicates:... 0.919 sec	0.919 sec
CREATE TABLE location AS SELECT Location_id , city, county, zipcode FROM P;	421	00:05:21	CREATE TABLE LOCATION AS SELECT LOCATION_ID , CITY, COUNTY, ZIPCODE FROM P;	75843 row(s) affected Records: 75843 Duplicates:... 0.871 sec	0.871 sec
CREATE TABLE Weather AS SELECT Location_id, weather_timestamp, temperature, humidity, pressure, weather_condition, ID FROM P;	422	00:05:25	CREATE TABLE WEATHER AS SELECT LOCATION_ID, WEATHER_TIMESTAMP, TEMPERATURE, HUMIDITY, PRESSURE, WEATHER_CONDITION, ID FROM P;	75843 row(s) affected Records: 75843 Duplicates:... 1.259 sec	1.259 sec
CREATE TABLE GPS AS SELECT Start_Lat,Start_Lng,End_Lat,End_Lng,Location_Id FROM P;	423	00:05:29	CREATE TABLE GPS AS SELECT START_LAT,START_LNG,END_LAT,END_LNG,LOCATION_ID FROM P;	75843 row(s) affected Records: 75843 Duplicates:... 0.854 sec	0.854 sec
CREATE TABLE City AS SELECT City,State FROM P;	424	00:05:34	CREATE TABLE CITY AS SELECT CITY,STATE FROM P;	75843 row(s) affected Records: 75843 Duplicates:... 0.944 sec	0.944 sec

We have created 5 tables according to the entities we have decided to work on.

- 4. Try to query your database especially using more than one table (by joining them) to see if the results make sense or not.**
- a. Try at least 3 join queries to see if the joins are working properly. Explain how you believe the joins are working. Check if the results of these queries match what you expect.**

**i. Id is a primary key and must be unique**

```

3 • insert into accident values
4   ('A-1',3,'2016-02-08 00:37:08','2016-02-08 06:37:08',
5   'Between Sawmill Rd/Exit 20 and OH-315/Olentangy Riv Rd/Exit 22 - Accident.', '9419884');
6

```

90:105 | 1 error found

Action	Output	Time	Action	Response	Duration / Fetch Time
insert into accident values ('A-1',3,'2016-02-08 00:37:08','2016-02-08 06:37:08','Between Sawmill Rd/Exit 20 and OH-315/Olentangy Riv Rd/Exit 22 - Accident.', '9419884');	347	22:39:32	SELECT * FROM accident LIMIT 0, 1000	Error Code: 1146. Table 'ud.accident' doesn't exist	0.00047 sec
	348	22:40:06	use p	0 row(s) affected	0.00052 sec
	349	22:40:09	SELECT * FROM accident LIMIT 0, 1000	1000 row(s) returned	0.0056 sec / 0.0032...
	350	22:41:24	insert into accident values ('A-1',3,'2016-02-08 00:37:08','2016-02-08 06:37:08','Between Sawmill Rd/Exit 20 and OH-315/Olentangy Riv Rd/Exit 22 - Accident.', '9419884');	Error Code: 1062. Duplicate entry 'A-1' for key 'accid...' 0.0044 sec	0.0044 sec

**ii. Id is primary and must be not null**

```

103 • insert into accident values
104   (NULL,3,'2016-02-08 00:37:08','2016-02-08 06:37:08',
105   'Between Sawmill Rd/Exit 20 and OH-315/Olentangy Riv Rd/Exit 22 - Accident.', '9419884');
106

```

7:104 | 1 error found

Action	Output	Time	Action	Response	Duration / Fetch Time
insert into accident values (NULL,3,'2016-02-08 00:37:08','2016-02-08 06:37:08','Between Sawmill Rd/Exit 20 and OH-315/Olentangy Riv Rd/Exit 22 - Accident.', '9419884');	346	22:40:00	use p	0 rows affected	0.00002 sec
	349	22:40:09	SELECT * FROM accident LIMIT 0, 1000	1000 row(s) returned	0.0056 sec / 0.0032...
	350	22:41:24	insert into accident values ('A-1',3,'2016-02-08 00:37:08','2016-02-08 06:37:08','Between Sawmill Rd/Exit 20 and OH-315/Olentangy Riv Rd/Exit 22 - Accident.', '9419884');	Error Code: 1062. Duplicate entry 'A-1' for key 'accid...' 0.0044 sec	0.0044 sec
	351	22:44:26	insert into accident values (NULL,3,'2016-02-08 00:37:08','2016-02-08 06:37:08','Between Sawmill Rd/Exit 20 and OH-315/Olentangy Riv Rd/Exit 22 - Accident.', '9419884');	Error Code: 1048. Column 'ID' cannot be null 0.00063 sec	0.00063 sec

- iii. Foreign key relationships involve a parent table that holds the central data values and a child table with identical values pointing back to its parent. The FOREIGN KEY clause is specified in the child table. It will reject any INSERT or UPDATE operation that attempts to create a foreign key value in a child table if there is no matching candidate key value in the parent table

```

126
127 • | insert into accident values
128   ('A1',3,'2016-02-08 00:37:08','2016-02-08 06:37:08',
129     'Between Sawmill Rd/Exit 20 and OH-315/Olentangy Riv Rd/Exit 22 - Accident.', '123');
130
100% 1:127 | 2 errors found

Action Output
Time Action Response Duration / Fetch Time
384 23:40:19 SELECT * FROM City c RIGHT JOIN Location l ON c.City = l.City LIMIT 0, 1000 1000 row(s) returned 0.00039 sec / 0.0077...
385 23:41:49 SELECT distinct city,state,location_id,county FROM City c RIGHT JOIN Locatio... Error Code: 1052. Column 'city' in field list is ambigu... 0.00068 sec
386 23:42:52 SELECT distinct c.city,c.state,l.location_id,l.county FROM City c RIGHT JOIN L... 1000 row(s) returned 1.370 sec / 0.00068 s...
387 00:02:41 insert into accident values ('A1',3,'2016-02-08 00:37:08','2016-02-08 06:37:08...', Error Code: 1452. Cannot add or update a child row:... 0.015 sec

```

- iv. It is not possible to drop a table if there is a foreign key constraint that references this table

```

132 • | drop table location;
100% 21:132

Action Output
Time Action Response Duration / Fetch Time
188 22:44:05 insert into accident(id) values (1) Error Code: 1062. Duplicate entry '1' for key 'acciden... 0.0067 sec
189 22:58:23 drop table location Error Code: 3730. Cannot drop table 'location' refere... 0.032 sec

```

- b. Try at least 3 queries with regard to the constraints you have in your database to see if those constraints are working properly. Explain how you believe they are working.

- i. To get city,state,county against the location id we are joining city and location

```

SELECT distinct c.city,c.state,l.location_id,l.county
FROM City c
RIGHT JOIN Location l ON c.City = l.City;

```

100% 42:104 | 2 errors found

**Result Grid** Filter Rows: Search Export: Fetch rows:

city	state	location_id	county
Libertyville	IL	55	Lake
Knoxville	TN	187	Knox
Knoxville	MD	187	Knox
Romulus	MI	444	Wayne
Corona	CA	597	Riverside
Corona	NY	597	Riverside
Corona	NM	597	Riverside

ii. To get all the weather conditions against the accident

```

115
116 •   SELECT a.ID, a.Severity, w.Temperature, w.Humidity, w.Pressure, w.Weather_Condition
117   FROM Weather w
118   RIGHT JOIN Accident a ON a.Location_Id = w.Location_Id;

```

100% 56:118 2 errors found

Result Grid Filter Rows: Search Export: Fetch rows:

ID	Severity	Temperature	Humidity	Pressure	Weather_Conditi...
A-1	3	42.1	58	29.76	Light Rain
A-10	2	32	100	29.59	Snow
A-100	2	12	77	30.17	
A-1000	2	39	93	30.27	Clear
A-10000	2	63	72	30.06	Overcast
A-10001	2	66.2	56	30.07	Overcast
A-10002	2	70	51	30.05	Partly Cloudy
A-10003	2	69.1	51	30.05	Mostly Cloudy
A-10004	2	64.9	56	30.05	Partly Cloudy
A-10005	2	66	54	30.02	Clear
A-10006	2	64.4	52	30.08	Clear
A-10007	2	64.4	52	30.08	Clear
A-10008	2	68	49	30.09	Scattered Clouds
A-10009	2	69.1	57	30.04	Mostly Cloudy
A-1001	2	48.9	77	30.3	Scattered Clouds
A-10010	2	69.1	57	30.04	Mostly Cloudy
A-10011	3	70	53	30.04	Overcast

iii. To fetch all the gps details against an accident we are joining accident table and gps table.

```

108 •   SELECT a.ID, a.Severity, g.Start_Lat, g.Start_Lng, g.End_Lat, g.End_Lng
109   FROM Accident a
110   JOIN GPS g ON a.Location_Id = g.Location_id;
111

```

100% 45:110 1 error found

Result Grid Filter Rows: Search Export: Fetch rows:

ID	Severity	Start_Lat	Start_Lng	End_Lat	End_Lng
A-1	3	40.10891	-83.09286	40.11206	-83.03187
A-2	2	39.86542	-84.0628	39.86501	-84.04873
A-3	2	39.10266	-84.52468	39.10209	-84.52396
A-4	2	41.06213	-81.53784	41.06217	-81.53547
A-5	3	39.172393	-84.492792	39.170476	-84.501798
A-6	2	39.06324	-84.03243	39.06731	-84.05851
A-7	2	39.77565	-84.18603	39.77275	-84.18805
A-8	2	41.37531	-81.82017	41.36786	-81.82174
A-9	2	40.702247	-84.075887	40.69911	-84.084293
A-10	2	40.10931	-82.96849	40.11078	-82.984
A-11	2	39.19288	-84.47723	39.19615	-84.47335
A-12	2	39.13877	-84.53394	39.13977	-84.5343
A-13	2	41.4739	-81.704233	41.47388	-81.70559

