600.315 Databases Final Project

Tianyi Lin (tlin44@jhu.edu)
Yang Cao (ycao29@jhu.edu)

1) Description of application domain: Our database describes the locations of crashes involving bikes and cars in the Chapel Hill Region of North Carolina.We download the data online, and the original source is "from police-reported bicycle-motor vehicle and pedestrian-motor vehicle collisions that occurred on the public roadway network, public vehicular areas and private properties (if reported)", as stated on the website.

2) View our reseults (all steps start from the Crashes_NC folder):

- Please follow the instruction in section 6 (User's Guide) to populate the database and add stored procedures
- To view the website, first initiate apache server, and go to webpage:
  - "chmod 704 index.html"
  - "php -S 127.0.0.1:8080"
  - open Google Chrome (browser), type in the url :"http://localhost:8080/index.html"
  - now you can play with the questions we listed and try out each options in the drop down list :)
- To return to the main page (index.html), you could simply click the return button on the left top corner of the browser (for Chrome)

3.Change after phase 1

- We attach our new phase 1 at the end of the report, with the updated sql entries, database table examples, etc. But for questions that are overlapped with the questions asked here, we explain more details in here. Thus, we hope you can based mostly on this phase 2 report.
- After looking more in depth with our data sets, we decided to use the two out of three candidate datasets. Instead of including bike crashes, pedestrian crashes, and criminal records, we choose only the first two, as they both come from the same city in North Carolina, and the entries within are very similar. Therefore, we choose to analyze these two types of crashes and their factors over a more general yet too-scattered data.

4. Source of data & extraction

- We download the csv files from these online sources:https://catalog.data.gov/dataset/bicycle-crashes, https://catalog.data.gov/dataset/pedestrian-crashes and save them in the folder "raw_data".
- First, we use "Numbers" (an apple spreadsheet software that comes with mac) to change the csv files to encoding of UTF-8, and save them as "pedestrianData.csv" and "bikeCrash.csv"
- Then, we take 2 approaches to extract the data from the csvs.

- 1. We use java to extract data from the csv file for tables: PedInjParser and ReasonPed.
    - The reason we choose these two tables is because we want to add some columns to both tables, which are not originally in the csv but require some extra calculation. So we use java to pre-process the datasets.
    - We write our own Java classes for each object in the table. For example, for table pedInjure, we first write the file PedInjure.java which is the object class, and the variables within are the columns in the table. Then we use PedInjParser.java to read in the csv, parse the table, store the variables we care into the PedInjure object, and finally write the corresponding sql statement to create the table PedInjure as well as populating the table.
- 2. Considering the size and amounts of our tables, we use the online csv converter for the rest of datasets, the link to the webpage is http://www.convertcsv.com/csv-to-sql.htm.

5. Software & hardware
- We use the MySQL dbase on our JHU ugrad machine.

6. User's guide
- Step 1, for our data pre-processing part, please follow the instructions below:
    - "cd dataProcess"
    - "javac *.java", then run "java PedInjParser" and "java ReasonPedParser" to construct "pedInjure.sql and " reasonPed.sql"
    - pipe in the two sql files to the database with
        - "mysql -h dbase.cs.jhu.edu -u ycao29 -D cs41518_ycao29_db -p -t -f -vvv < ./pedInjure.sql"
        - "mysql -h dbase.cs.jhu.edu -u ycao29 -D cs41518_ycao29_db -p -t -f -vvv < ./reasonPed.sql"
        - Note: the password is "wyxjaycqli"
    - Now the two tables PedInjure and ReasonPed are stored in our database.
- Step 2, in order to populate the rest of the dataset, please follow the instructions below:
    - "cd sql"
    - fill our database by inserting all the entries with
        - "mysql -h dbase.cs.jhu.edu -u ycao29 -D cs41518_ycao29_db -p -t -f -vvv < ./BikeCrash.sql"
        - "mysql -h dbase.cs.jhu.edu -u ycao29 -D cs41518_ycao29_db -p -t -f -vvv < ./PedestrianCrash.sql"
    - split the datasets into smaller tables that we'll make operations on:
        - "mysql -h dbase.cs.jhu.edu -u ycao29 -D cs41518_ycao29_db -p -t -f -vvv < ./CleanTableProcedures.sql"
    - load in the 15 procedures we write:
        - "mysql -h dbase.cs.jhu.edu -u ycao29 -D cs41518_ycao29_db -p -t -f -vvv < ./Procedures.sql"
- Step 3, initiate apache server, and go to webpage:
    - "chmod 704 index.html"
    - "php -S 127.0.0.1:8080"

    - open Google Chrome (browser), type in the url :"http://localhost:8080/index.html"
    - now you can play with the questions we listed and try out each options in the drop down list :)
- Note: to return to the main page (index.html), you could simply click the return button on the left top corner of the browser (for Chrome)

7. Major/minor areas of specialization
- Preprocessing of data
    - We downloaded original data in CSV format. For BikeCrash data, we populated our database table BikeCrash by converting CSV data to SQL file through online conversion website http://www.convertcsv.com/csv-to-sql.htm
    - For PedestrianCrash data, we applied knowledge in object-oriented design to parse the original CSV data. We create objects for each table, and use the dot operation (as explained in class) to access the variables.
    - We also manually created primary key for our big tables before dividing them into smaller ones, using the *AUTO_INCREMENT* SQL statement.
    - We then wrote stored procedures (see CleanTableProcedures.sql) such as *DivideBikeCrashTable()* to divide a huge table into smaller ones with related attributes. Before division, we also checked all attributes related to our future queries carefully, deleting and combining columns if necessary (see CombineColumns() in CleanTableProcedures.sql)
- Complex stored procedures
    - Although we have only 15 queries, each query usually requires 2-3 mysql queries, where one retrieves information from PedestrianCrash data, one retrieves information from BikeCrash data, and another one returns a table that acts like a legend should information be too complex for users to interpret.
    - Additionally, we attempted to analyze and retrieve complex information. For example, for some queries, we have to display the most frequent value for multiple attributes during the user-specified time (from and to) in one single table. We also attempted and succeeded in displaying two lists, one displaying pedestrian crash types and one displaying bike crash types, as two columns in one table.
- MySQLi Language
    - At first, when we use MySQL database system with our php files, we had trouble displaying multiple, separate tables in one php. After doing some research, we learned that MySQLi extension supports multiple statements (and prepared statements, etc.), so we quickly switched to MySQLi and successfully displayed as many tables as we would like in one php.

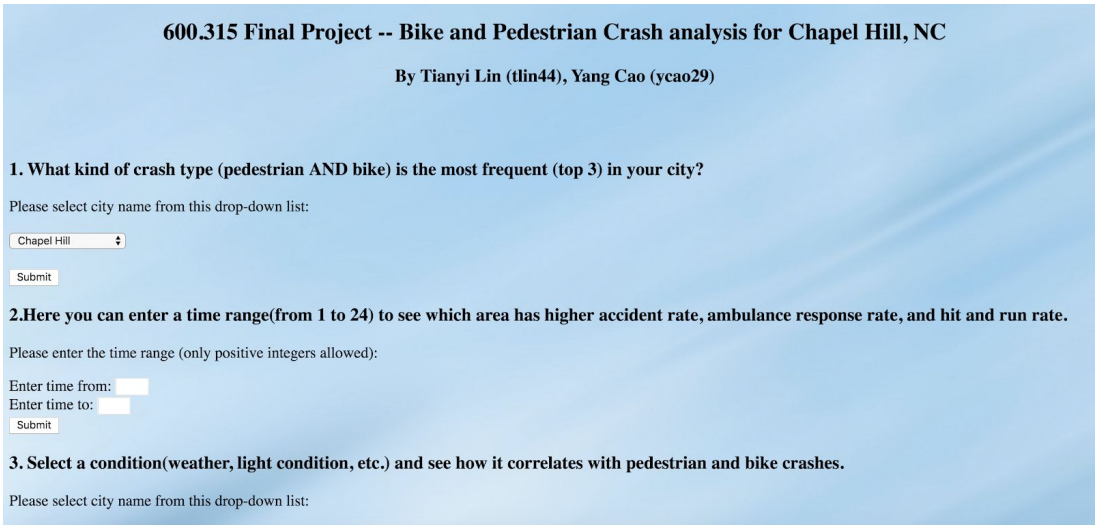
8. Strengths
- Combined queries. When we wrote our queries, we realized that some queries require the same type of user input, such as time range and age range, so we combined queries with the same type of user input together so that the user only has to enter the input once and get all corresponding query results together.
- Complex stored procedures. As mentioned in part (7), we wrote many procedures that, although only return a few lines, require complex sql manipulations.

- Check valid input using html. For hw3, we do check on php and sql side, but this time, as we do some more research on html, we use different forms and constraints on them to check whether the input is valid.
- Provide several related factors for each question. For instance, in question 3 (Select a condition(weather, light condition, etc.) and see how it correlates with pedestrian and bike crashes), we not only present the data related to user's input, but also give the most frequent type of other factors. If the user choose "road surface", we also shows the "most frequent victim severity" and "Most Frequent Weather" for each type of road surface.

9. Limitations
   - More polished user interface. We only had time to use the most basic html language that displays a very simple interface for users to interact with.
   - Graph analysis. Since both our pedestrian and bike crash data contain time information, we could draw and display graphs that shows patterns of how time correlates with certain attributes.
   - Server. Now our project can only run on ugrad server. If we had more time, we might be able to setup a public server through Heroku.

10. Everything was done on our own and NOT for any other project/course/research.

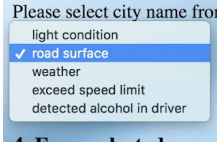
11.
   - Our main html page:



   - The user can choose which factor they want to see, and the factor's correlation with our dataset. As for the output, we not only present the road surface type and their number of crashes & rate, but also the most frequent severity and the most frequent weather. This intuitively makes sense, since a user might be curious about under what weather does each surface condition causes crash the most, or what's the most probable injure one will get when being hit on a certain type of road.

3. Select a condition(weather, light condition, etc.) and see how it correlates with pedestrian and bike crashes.

Please select city name from this drop-down list:



Pedestrian crashes:

Surface Condition	Count	Percentage	Most Frequent Severity	Most Frequent Weather
Smooth Asphalt	222	69.3750	B: Evident Injury	Clear
Coarse Asphalt	81	25.3125	B: Evident Injury	Clear
Concrete	11	3.4375	B: Evident Injury	Clear
Gravel	5	1.5625	B: Evident Injury	Clear
Unknown	1	0.3125	B: Evident Injury	Clear

Bike crashes:

Surface Condition	Count	Percentage	Most Frequent Severity	Most Frequent Weather
Smooth Asphalt	131	79.3939	B: Evident Injury	Clear
Coarse Asphalt	29	17.5758	B: Evident Injury	Clear
Concrete	3	1.8182	B: Evident Injury	Clear
Other	1	0.6061	B: Evident Injury	Clear
Gravel	1	0.6061	B: Evident Injury	Clear

- Here, we display the age group and the pedestrian position, injury type, etc. for each victim in the database.

Note1: Since the entries within each age group is very limited, we present the entire dataset below.

Note2: If the table is empty, it means there are no crashes within this age group.
Info about pedestrian crash victims:

Count	Age Group	Pedestrian Position	Pedestrian Race	Pedestrian Injury	Pedestrian Sex
11	6-10	Non-Roadway - Parking Lot / Other	White	B: Evident Injury	Female
11	6-10	Non-Roadway - Parking Lot / Other	Hispanic	C: Possible Injury	Male
11	6-10	Driveway / Alley	Asian	C: Possible Injury	Female
11	6-10	Non-Roadway - Parking Lot / Other	Black	C: Possible Injury	Male
11	6-10	Crosswalk Area	White	C: Possible Injury	Male
11	6-10	Travel Lane	Black	B: Evident Injury	Female
11	6-10	Non-Roadway - Parking Lot / Other	Black	C: Possible Injury	Female
11	6-10	Travel Lane	White	A: Disabling Injury	Male
11	6-10	Non-Roadway - Parking Lot / Other	White	O: No Injury	Female
11	6-10	Non-Roadway - Parking Lot / Other	Hispanic	C: Possible Injury	Female
11	6-10	Crosswalk Area	Black	B: Evident Injury	Male

Info about bike crash victims:

Count	Age Group	Biker Position	Biker Race	Biker Injury	Biker Sex	Biker Drink Alcohol
6	6-10	Travel Lane	Black	B: Evident Injury	Male	No
6	6-10	Driveway / Alley	White	C: Possible Injury	Male	No
6	6-10	Driveway / Alley	White	B: Evident Injury	Male	No
6	6-10	Sidewalk / Crosswalk / Driveway Crossing	White	C: Possible Injury	Male	No
6	6-10	Travel Lane	Black	B: Evident Injury	Male	No
6	6-10	Sidewalk / Crosswalk / Driveway Crossing	Black	B: Evident Injury	Female	No

- For question 5 (Display analysis on both bike and pedestrian crash related to the severity of injury, frequency of crashes at intersection/non-intersection and traffic control), we combine several stored procedure in the output of this one question. And we label the parts with corresponding potential questions. In addition, we list out all injury types for user's reference,

since we only show the highest crash severity type for pedestrian and bike, it'll be better for user to understand the condition better when knowing the whole criteria.

-

Part 1: which type of crash (pedestrian or bike) has higher severity of injury?
Pedestrian crash:

Crash Severity Level	Percentage
B: Evident Injury	40.6250
C: Possible Injury	40.6250

Bike crash:

Crash Severity Level	Percentage
B: Evident Injury	51.5152

A list of all injury types:

Crash Severity Level
A: Disabling Injury
B: Evident Injury
C: Possible Injury
K: Killed
O: No Injury

Part 2: for both pedestrian and bike crashes, do crashes happen more often at intersection/non-intersection?

Crash Location	Percentage	Most Frequent Weather	Most Frequent Light Condition
Intersection	36.2887	Clear	Daylight
Non-Intersection	36.0825	Clear	Daylight
Non-Roadway	19.3814	Clear	Daylight

- For the last problem, we present the driver information. We show the category of driver on the left, and the counts on the right. Thus, the user can easily compare the number within each category (ex. sex, race, etc.)

To give the idea of distribution of driver's sex, race, age, etc. we display the count of each category below:

Category	Count For Each Category
Female	125
Male	148
Unknown	47
Asian	15
Black	58
Hispanic	20
Native American	1
Other	6
Unknown/Missing	47
White	173
0-19	25
20-24	37
25-29	28
30-39	52
40-49	43
50-59	35
60-69	24
70+	27
Unknown	49
Assault with Vehicle	2

12.

1)

```
CREATE TABLE BikeCrashTime (
        BikeCrashID     INT NOT NULL,
        crash_time      VARCHAR(5) NOT NULL,
        crash_hour      DECIMAL(4,1) NOT NULL,
        crashday        VARCHAR(9) NOT NULL,
        crash_mont      VARCHAR(9) NOT NULL,
        crash_year      INT(11) NOT NULL,
        PRIMARY KEY (BikeCrashID)
);
# INSERT INTO
BikeCrashTime(BikeCrashID,crash_time,crash_hour,crashday,crash_mont,crash_year) VALUES (1,
“10:12”, 10.0, “Saturday”, “July”, 2011);
```

BikeCrashID	crash_time	crash_hour	crashday	crash_mont	crash_year
1	“10:12”	10.0	“Saturday”	“July”	2011

2)

```
CREATE TABLE BikeCrashLoc (
        BikeCrashID     INT NOT NULL,
        lat             DECIMAL(12,10) NOT NULL,
        lon             DECIMAL(12,10) NOT NULL,
        county          VARCHAR(7) NOT NULL,
        city            VARCHAR(18) NOT NULL,
        rural_urba      VARCHAR(5) NOT NULL,
        crash_loc       VARCHAR(20) NOT NULL,
        developmen      VARCHAR(22) NOT NULL,
        PRIMARY KEY (BikeCrashID)
);
# INSERT INTO BikeCrashLoc(BikeCrashID,lat,lon,county,city,rural_urba,crash_loc,developmen)
VALUES (1, 35.9100670923, -79.0745027481, “Orange”, “Carrboro”, “Urban”, “Non-Intersection”,
“Commercial”);
```

BikeCrashID	lat	lon	county	city	rural_urba	crash_loc	developmen
1	35.9100670923	-79.0745027481	“Orange”	“Carrboro”	“Urban”	“Non-Intersection”	“Commercial

3)

```
CREATE TABLE BikeCrashRdCond (
        BikeCrashID     INT NOT NULL,
        rd_defects      VARCHAR(7) NOT NULL,
        rd_feature      VARCHAR(23) NOT NULL,
        rd_charact      VARCHAR(20) NOT NULL,
        rd_surface      VARCHAR(14) NOT NULL,
        rd_conditi      VARCHAR(24) NOT NULL,
        speed_limi      VARCHAR(12) NOT NULL,
        traff_cntr      VARCHAR(35) NOT NULL,
        weather         VARCHAR(6) NOT NULL,
        rd_config       VARCHAR(41) NOT NULL,
        num_lanes       VARCHAR(15) NOT NULL,
        developmen      VARCHAR(22) NOT NULL,
        light_cond      VARCHAR(26) NOT NULL,
        PRIMARY KEY (BikeCrashID)
);
# INSERT INTO BikeCrashRdCond
(BikeCrashID,rd_defects,rd_feature,rd_charact,rd_surface,rd_conditi,speed_limi,traff_cntr,weather,rd
_config,num_lanes,developmen,light_cond) VALUES (1, “None”, “No Special Feature”, “Straight -
Level”, “Smooth Asphalt”, “Dry”, “20 - 25 MPH”, “No Control Present”, “Clear”, “Two-Way, Not
Divided”, “Unknown”, “Commercial”, “Daylight”);
```

BikeCrashID	rd_defects	rd_feature	rd_charact	rd_surface	rd_conditi	speed_limi	traff_cntr	weather	rd_config	num_lanes	developmen	light_cond
1	“None”e	“No Special Feature”	“Straight - Level”	“Smooth Asphalt”	“Dry”	“20 - 25 MPH”	“No Control Present”	“Clear”	“Two-Way, Not Divided”	“Unknown”	“Commercial”	“Daylight”

4)

```
CREATE TABLE BikeCrashResult (
        BikeCrashID     INT NOT NULL,
        ambulancer      VARCHAR(3) NOT NULL,
        crash_type      VARCHAR(62) NOT NULL,
        crsh_sevri      VARCHAR(19) NOT NULL,
        hit_run         VARCHAR(3) NOT NULL,
        bike_injur      VARCHAR(19) NOT NULL,
        drvr_injur      VARCHAR(18) NOT NULL,
        PRIMARY KEY (BikeCrashID)
);
```

INSERT INTO BikeCrashResult
(BikeCrashID,ambulancer,crash_type,crsh_sevri,hit_run,bike_injur,drvr_injur) VALUES (1, “Yes”, “Motorist Overtaking - Bicyclist Swerved”, “K: Killed”, “No”, “K: Killed”, “O: No Injury”);

BikeCrashID	ambulancer	crash_type	crsh_sevri	hit_run	bike_injur	drvr_injur
1	“Yes”	“Motorist Overtaking - Bicyclist Swerved”	“K: Killed”	“No”	“K: Killed”	“O: No Injury”

5)
CREATE TABLE Biker (
BikeCrashID INT NOT NULL,
bike_injur VARCHAR(19) NOT NULL,
bike_race VARCHAR(15) NOT NULL,
bike_dir VARCHAR(14) NOT NULL,
bike_age VARCHAR(7) NOT NULL,
bikeage_gr VARCHAR(7) NOT NULL,
bike_sex VARCHAR(7) NOT NULL,
bike_pos VARCHAR(40) NOT NULL,
bike_alc_d VARCHAR(7) NOT NULL,
PRIMARY KEY (BikeCrashID)
);
INSERT INTO Biker
(BikeCrashID,bike_injur,bike_race,bike_dir,bike_age,bikeage_gr,bike_sex,bike_pos,bike_alc_d)
VALUES (1, “K: Killed”, “White”, “With Traffic”, “70+”, “70+”, “Male”, “Travel Lane”, “No”);

BikeCrashID	bike_injur	bike_race	bike_dir	bikeage_gr	bike_age	bike_sex	bike_pos	bike_alc_d
1	“K: Killed”	“White”	“With Traffic”	“70+”	“70+”	“Male”	“Travel Lane”	“No”

6)
CREATE TABLE Driver_BikeCrash(
BikeCrashID INT NOT NULL,
drvr_vehty VARCHAR(34) NOT NULL,
drvr_injur VARCHAR(18) NOT NULL,
drvr_sex VARCHAR(7) NOT NULL,
drvr_race VARCHAR(15) NOT NULL,

```
        drvr_age        VARCHAR(7) NOT NULL,
        drvrage_gr      VARCHAR(7) NOT NULL,
        drvr_estsp      VARCHAR(9) NOT NULL,
        drvr_alc_d      VARCHAR(7) NOT NULL,
        PRIMARY KEY (BikeCrashID)
);
# INSERT INTO Driver_BikeCrash
(BikeCrashID,drvr_vehty,drvr_injur,drvr_sex,drvr_race,drvr_age,drvrage_gr,drvr_estsp,drvr_alc_d)
VALUES (1, “Passenger Car”, “O: No Injury”, “Male”, “White”, “70+”, “70+”, “11 - 15 mph”,
“No”);
```

BikeCrashID	drvr_vehty	drvr_injur	drvr_sex	drvr_race	drvr_age	drvrage_gr	drvr_estsp	drvr_alc_d
1	“Passenger Car”	“O: No Injury”	“Male”	“White”	“70+”	“70+”	“11 - 15 mph”	“No”

7)

```
CREATE TABLE BikeCrashReason(
        BikeCrashID     INT NOT NULL,
        crashalcoh      VARCHAR(3) NOT NULL,
        excsspdind      VARCHAR(3) NOT NULL,
        drvr_alc_d      VARCHAR(7) NOT NULL,
        bike_alc_d      VARCHAR(7) NOT NULL,
        bike_pos        VARCHAR(40) NOT NULL,
        bike_dir        VARCHAR(14) NOT NULL,
        drvr_estsp      VARCHAR(9) NOT NULL,
        on_rd           VARCHAR(23),
        PRIMARY KEY (BikeCrashID)
);
# INSERT INTO BikeCrashReason
(BikeCrashID,crashalcoh,excsspdind,drvr_alc_d,bike_alc_d,bike_pos,bike_dir,drvr_estsp,on_rd)
VALUES (1, “No”, “No”, “No”, “No”, “Travel Lane”, “With Traffic”, “11 - 15 mph”, NULL);
```

BikeCrashReason

BikeCrashID	crashalcoh	excsspdind	drvr_alc_d	bike_alc_d	bike_pos	bike_dir	drvr_estsp	on_rd
1	“No”	“No”	“No”	“No”	“Travel Lane”	“With Traffic”	“11 - 15 mph”	NULL

8)

```
CREATE TABLE PedCrashRdCond (
        BikeCrashID     INT NOT NULL,
```

```
rd_defects      VARCHAR(22) NOT NULL,
rural_urba      VARCHAR(5) NOT NULL,
city            VARCHAR(18) NOT NULL,
locality        VARCHAR(28) NOT NULL,
rd_feature      VARCHAR(32) NOT NULL,
light_cond      VARCHAR(26) NOT NULL,
rd_charact      VARCHAR(20) NOT NULL,
rd_surface      VARCHAR(14) NOT NULL,
developmen      VARCHAR(22) NOT NULL,
traff_cntr      VARCHAR(35) NOT NULL,
rd_conditi      VARCHAR(7) NOT NULL,
```

```
    region          VARCHAR(8) NOT NULL,
    rd_class        VARCHAR(22) NOT NULL,
    weather         VARCHAR(40) NOT NULL,
    num_lanes       VARCHAR(15) NOT NULL,
    rd_config       VARCHAR(41) NOT NULL,
    PRIMARY KEY (BikeCrashID)
);
# INSERT INTO PedCrashRdCond
(BikeCrashID,rd_defects,rural_urba,city,locality,rd_feature,light_cond,rd_charact,rd_surface,develop
men,traff_cntr,rd_conditi,region,rd_class,weather,num_lanes,rd_config) VALUES (1, “None”,
“Urban”, “Chapel Hill”, “Urban (>70% Developed)”, “No Special Feature”, “Dark - Roadway Not
Lighted”, “Straight - Level”, “Smooth Asphalt”, “Commercial”, “No Control Present”, “Dry”,
“Piedmont”, “Public Vehicular Area”, “Clear”, “Unknown”, “Unknown”);
```

PedCrashID	rd_defects	rural_urba	city	locality	rd_feature	light_cond	rd_charact
1	“None”	“Urban”	“Chapel Hill”	“Urban (>70% Developed)”	“No Special Feature”	“Dark - Roadway Not Lighted”	“Straight - Level”

rd_surface	developmen	traff_cntr	rd_conditi	region	rd_class	weather	num_lanes	rd_config
“Smooth Asphalt”	“Commercial”	“No Control Present”	“Dry”	“Piedmont”	“Public Vehicular Area”	“Clear”	“Unknown”	“Unknown”

9)

```
CREATE TABLE DiverBiker_PedCrash (
    BikeCrashID     INT NOT NULL,
    drvr_age        VARCHAR(7) NOT NULL,
    drvrage_gr      VARCHAR(7) NOT NULL,
    drvr_estsp      VARCHAR(9) NOT NULL,
    speed_limi      VARCHAR(12) NOT NULL,
    drvr_vehty      VARCHAR(36) NOT NULL,
    drvr_injur      VARCHAR(19) NOT NULL,
    drvr_sex        VARCHAR(7) NOT NULL,
    drvr_race       VARCHAR(15) NOT NULL,
    drvr_alc_d      VARCHAR(7) NOT NULL,
    PRIMARY KEY (BikeCrashID)
);
```

INSERT INTO DiverBiker_PedCrash
(BikeCrashID,drvr_age,drvrage_gr,drvr_estsp,speed_limi,drvr_vehty,drvr_injur,drvr_sex,drvr_race,drvr_alc_d) VALUES (2, “46”, “40-49”, “Unknown”, “30 - 35 MPH”, “Passenger Car”, “O: No Injury”, “Female”, “White”, “No”);

BikeCrashID	drvr_age	drvrage_gr	drvr_estsp	speed_limi	drvr_vehty	drvr_injurdrvr_sex	drvr_alc_d	drvrage_gr	drvr_estsp
1	“46”	“40-49”	“Unknown”	“30 - 35 MPH”	“Passenger Car”	“O: No Injury”	“Female	“White”	“No”

10)
CREATE TABLE PedCrashDetail(

BikeCrashID	INT NOT NULL,
crsh_sevri	VARCHAR(19),
ambulancer	VARCHAR(3) NOT NULL,
crash_time	VARCHAR(5) NOT NULL,
crash_year	INT(11) NOT NULL,
county	VARCHAR(7) NOT NULL,
longitude	DECIMAL(5,1) NOT NULL,
latitude	DECIMAL(4,1) NOT NULL,
crash_mont	VARCHAR(9) NOT NULL,
crash_type	VARCHAR(50) NOT NULL,
city	VARCHAR(18) NOT NULL,
locality	VARCHAR(28) NOT NULL,
ped_pos	VARCHAR(46) NOT NULL,
drvr_injur	VARCHAR(19) NOT NULL,
crashday	VARCHAR(9) NOT NULL,
crash_loc	VARCHAR(20) NOT NULL,
crash_hour	DECIMAL(4,1) NOT NULL,
geo_shape	VARCHAR(74) NOT NULL,
crash_date	DATE NOT NULL,
crash_grp	VARCHAR(45) NOT NULL,
hit_run	VARCHAR(3) NOT NULL,
PRIMARY KEY (BikeCrashID)	

);
INSERT INTO PedCrashDetail
(BikeCrashID,crsh_sevri,ambulancer,crash_time,crash_year,county,longitude,latitude,crash_mont,crash_type,city,locality,ped_pos,drvr_injur,crashday,crash_loc,crash_hour,geo_shape,crash_date,crash_grp,hit_run) VALUES (1, “B: Evident Injury”, “Yes”, “1:52”, 2007, “Orange”, -79.0, 36.0,

“November”, “Assault with Vehicle”, “Chapel Hill”, “Urban (>70% Developed)”, “Non-Roadway - Parking Lot / Other”, “Unknown Injury”, “Saturday”, “Non-Roadway”, 1.0, {"type": "Point", "coordinates": [-79.02140273340797, 35.93761709952935]}, 0000-00-00, “Unusual Circumstances”, “No”);

PedCrashID	crsh_sevri	ambulancer	crash_time	crash_year	county	longitude	latitude	crash_mont	crash_type
1	“B: Evident Injury”	“Yes”	“1:52”	2007	“Orange”	-79.0	36.0	“November”	“Assault with Vehicle

city	locality	ped_pos	drvr_injur	crash day	crash_loc	crash_hour	geo_shape	crash_date	crash_grp	hit_run
“Chapel Hill”	“Urban (>70% Developed)”	“Non-Roadway - Parking Lot / Other”	“Unknown Injury”	“Saturday”	“Non-Roadway”	1.0	{"type": "Point", "coordinates": [-79.02140273340797, 35.93761709952935]}	0000-00-00	“Unusual Circumstances”	“No”

11)

```
CREATE TABLE PedInjure (
        BikeCrashID     INT NOT NULL,
        ped_pos         VARCHAR(46) NOT NULL,
        ped_race        VARCHAR(15) NOT NULL,
        pedage_grp      VARCHAR(7) NOT NULL,
        ped_age         VARCHAR(7) NOT NULL,
        ped_injury      VARCHAR(19) NOT NULL,
        ped_sex         VARCHAR(7) NOT NULL,
        PRIMARY KEY (BikeCrashID)
```

);
INSERT INTO PedInjure
(BikeCrashID,ped_pos,ped_race,pedage_grp,ped_age,ped_injury,ped_sex) VALUES (1,
“Non-Roadway - Parking Lot / Other”, “Black”, “25-29”, “29”, “B: Evident Injury”, “Male”);

PedCrashID	ped_pos	ped_race	pedage_grp	ped_age	ped_injury	ped_sex
1	“Non-Roadway - Parking Lot / Other”	“Black”	“25-29”	“29”	“B: Evident Injury”	“Male”

12)

CREATE TABLE ReasonPed(

BikeCrashID	INT NOT NULL,
crashalcoh	VARCHAR(60) NOT NULL,
excsspdind	VARCHAR(30) NOT NULL,
ped_pos	VARCHAR(60) NOT NULL,
drvr_injur	VARCHAR(30) NOT NULL,
hit_run	VARCHAR(5) NOT NULL,
drvr_estsp	VARCHAR(30) NOT NULL,
exceedSpeed	VARCHAR(30) NOT NULL,
speed_limi	INT(11) NOT NULL,
PRIMARY KEY (BikeCrashID)	

);
INSERT INTO ReasonPed
(BikeCrashID,crashalcoh,excsspdind,ped_pos,drvr_injur,hit_run,drvr_estsp,exceedSpeed,speed_limi)
VALUES (1, “No”, “No”, “Non-Roadway - Parking Lot”, “Unknown Injury”, “No”, “Unknown”, -1,
5);

BikeCrashID	crashalcoh	excsspdind	ped_pos	drvr_injur	hit_run	drvr_estsp	exceedSpeed	speed_limi
1	“No”	“No”	“Non-Roadway - Parking Lot / Other ”	“Unknown Injury”	“No”	“Unknown”	-1	5

Tianyi Lin
Yang Cao

Database Phase 1

1.Members: Tianyi Lin, Yang Cao
2.Target domain: crimes in Baltimore area (and other cities) from 2001 to present
3.

1) What is the crime rate for a specific neighborhood?
2) What areas have higher crime rate at night and what areas have higher crime rate at day?
3) What kind of crime (e.g. murder, rape, robbery, b & e) is the most frequent around certain area?
4) What areas have higher accident rate (e.g. bicycle crash)?
5) What kind of accident is the most frequent around certain area?
6) What areas have higher accident rate at night and what areas have higher accident rate at day?
7) What neighborhood have caught more gun offenders?
8) Does weather correlate with accident rate?
9) Does weather correlate with bicycle crash rate?
10) Does light condition correlate with bicycle crash rate?
11) Does road condition correlated with bicycle crash rate?
12) What is the rate of car accidents that are caused by driver exceeding speed limit?
13) For bicycle crashes, how many of the Drivers have Alcohol Detected?
14) For people who committed crimes, how many of them are also gun offenders?
15) What age range (10 years, for example) has highest number of gun offenders?

4.
Crime_data

accident_id	States	Crime_type	Time	Police_station	People_involved	First name of People commited to crime	Last name of People commited to crime
123	MD	Robbery	19	Baltimore	Stan Lee, Eddie Brook	Eddie	Brook

Gun_offenders

caseNumber	created_date	modified_date	lastName	firstName	Date_Of_Birth	sex	full_address	district	neighborhood	Police_station	casualties	race
123	11/15/10028	11/18/2018	Fake	Name	1/2/2000	M	Room 123, Street 456.	Bad County	Hopkins	Baltimore Police Station	1 injured	Black

Crime_mapping

accident_id	crime_category	district	map_reference	location_category	lat	lon	location
123	Robbery	Bad County	123	Highway	111.222	333.444	123 Street 345

Bike_crash

City	Crash Date	Crash Location	Crash Time	Crash Severity	accident_id	Ambulance Response	Light Condition	Number of Lanes	Road Characteristics/Class/Condition/Configuration	Road Defects/Features
baltiomre	1/1/2018	Highway	11:23 pm	2 injured	123123	Yes	OK	3	No Special Feature	None

Bike_crash_people

accident_id	Bike/Pedestrian Age Group	Bike/Pedestrian Sex	Driver Age Group	Driver Estimated Speed	Speed Limit	Driver Alcohol Detected	Driver Injury	Crash Type
123	20	M	30	27	20	No	No	Motorist Overtaking - Bicyclist Swerved

5.

SQL statments for 15 questinos

1) What is the crime rate for a specific neighborhood?
    - select neighborhood with the highest count of crime for a neighborhood
2) What areas have higher crime rate at night and what areas have higher crime rate at day?
    - night: select top 3 neighborhoods that have the highest count of crime at night
    - day: select top 3 neighborhoods that have the highest count of crime at day
3) What kind of crime (e.g. murder, rape, robbery, b & e) is the most frequent around certain area?
    - group by crime type (from all tuples with the input neighborhood) and select the crime type that has max count
4) What areas have higher accident rate (e.g. bicycle crash)?
    - Combine Bike_crash with Crime_mapping to get the neighborhood in which a bike crash happens. Group by neighborhood and select the one with max count of bike crash
5) Rate of severe injury bike crashing people accident?
    - select count of severe injury accident / count of all accident from Bike_crash_people
6) What areas have higher accident rate at night and what areas have higher accident rate at day?
    - Night: group by neighborhood, select the neighborhood from Bike_crash and Crime_mapping that has max count of night accidents
    - Day: group by neighborhood, select the neighborhood from Bike_crash and Crime_mapping that has max count of day accidents
7) What neighborhood have caught more gun offenders?
    - Group by neighorbood, select max count of tuples from Gun_offenders

8) Does weather correlate with accident rate? (bad weather accident : all accident)
    - select count of accidents happening under bad weather / total count of accidents

9) Does weather correlate with bicycle crash rate? (bad weather bike crash : all crashes)
    - select count of bike crashes under bad weather / total count of bike crashes
10) Does light condition correlate with bicycle crash rate?
    - select count of bike crashes under bad light condition / total count of bike crashes
11) Does road condition correlated with bicycle crash rate?
    - select count of bike crashes under bad road condition / total count of bike crashes
12) What is the rate of car accidents that are caused by driver exceeding speed limit?
    - select exceed_count / totalCount from (select count of bike crashes where drive_speed > speed limit) as exceed_count, (select count of accidents) as totalCount
13) For bicycle crashes, how many of the Drivers have Alcohol Detected?
    - select count(*) where Driver_Alcohol_Detected = 'Yes'
14) For people who committed crimes, how many of them are also gun offenders?
    - select count(*) where Crime_data.First name of People commited to crime = Gun_offenders.FirstName AND Crime_data.Last name of People commited to crime = Gun_offenders.LastName

15) What age range (10 years, for example) has highest number of gun offenders?
    - select max from (select count(*) from gun_offenders group by age)

6. how to load database
    - We first put all the csv files into Json objects and put it in a json file. Then, along with the raw data that is originally in Json format, we use JDBC to handle the object and establish connection with a heroku server. Then within the java file, we populate the data into the database.

7.
    - We expect the output to be table of data, and our work includes combine multiple rows of data and present it to the user in a user friendly view. Possibly some design on the table format to make the information easier to read.

8.
    - Since we are in section 315, we choose to minorly focus on complex data extraction. So far, our raw data consists of both csv and json files, and we also plan to do some statistical analysis/calculation on these files before we push all of the datasets to the database.
    - We also plan to touch on JDBC to handle database connections and Json objects.

Database Phase 1 -- updated ver.

1.Members: Tianyi Lin, Yang Cao
2.Target domain: Our database describes the locations of crashes involving bikes and cars in the Chapel Hill Region of North Carolina.We download the data online, and the original source is "from police-reported bicycle-motor vehicle and pedestrian-motor vehicle collisions that occurred on the public roadway network, public vehicular areas and private properties (if reported)", as stated on the website.

3. List of queries (Note: for completion of phase 2, we also write the input & output and stored procedures name for each question)

1) What kind of crash type (pedestrian AND bike) is the most frequent (top 5) around certain area?
    a) input: city (drop down)
    b) output:
        i) tables (1 and 2): most frequent time of the crash, severity of ped/biker
        ii) table 3 (legend): kind of crash type
    c) procedures: CrashTypeRate_Bike(city VARCHAR(18)), CrashTypeRate_Ped(city VARCHAR(18)), ShowCrashTypes_Comb()

2) What areas have higher accident rate (e.g. bicycle crash)?What areas have higher accident rate at night and what areas have higher accident rate at day (based on time range)?
    a) input: time range
    b) output: accident rate, city, crash type, severity of injury
    c) procedures: AccidentRate_Bike(t_from NUMERIC(4,1), t_to NUMERIC(4,1)), AccidentRate_Ped(t_from NUMERIC(4,1), t_to NUMERIC(4,1))

3) Does light condition correlate with bicycle/pedestrian crash rate? (light_bike, light_ped)
    a) input:condition
    b) output: both table with columns of: type of light condition && corresponding # of crashes, % of light condition = && crash most frequent severity

4) Does road surface correlated with bicycle/pedestrian crash rate? (road_charact_bike, road_charact_ped)
    a) input:condition
    b) output: road surface type, count of crash, weather, severity

5) Does weather correlated with bicycle/pedestrian crash rate? (weather_bike,weather_ped)
    a) input:condition
    b) output: type of weather & # of crashes, several frequent month…

6) Does exceeding the speed limit (driver) correlate with crash rate? (output: coordinate with highest rate, link: other factors, navigate to corresponding pages)?
    a) input:condition
    b) output: percentage of exceedLim/all, percentage of belowLim/all
    c) procedures: ExceedSp_Bike(), ExceedSp_Ped()

7) For bicycle crashes, does Alcohol Detected for driver correlate with crash rate?
    a) input:
    b) output: percentage of alco/all for biker, percentage of alco/all for driver, time of the day, location

8) Do bike/pedestrian crashes have higher severity of injury?
    a) input:
    b) output:
        i) all types of injury
        ii) 2 tables, percentage of severe injury (level B/C), time, weather, alcohol, county
    c) procedures: ShowInjuryTypes(), Injury_Bike(), Injury_Ped()

9) Correlation between ambulance response and severity of injury
    a) input: time
    b) output:
        i) table 1 (ped): for ambulance=yes/no, the most frequent severity level
        ii) table 2 (bike): for ambulance=yes/no, the most frequent severity level
    c) procedures: AmbulanceSevri_Bike(t_from NUMERIC(4,1), t_to NUMERIC(4,1)), AmbulanceSevri_Ped(t_from NUMERIC(4,1), t_to NUMERIC(4,1))

10) For a selected age group, show all crashes data for the victim.
    a) variable: time
    b) input: age group (drop down?)
    c) output:
        i) table 1 (bike) count, Biker.bikeage_gr, bike_injur, bike_race, bike_dir, bike_sex, bike_pos, bike_alc_d
        ii) table 2 (ped) count, pedInjure.pedage_grp, ped_pos, ped_race, ped_injury, ped_sex #need to check

d) procedures: AgeGpAccidentRate_Bike(age VARCHAR(7)), AgeGpAccidentRate_Ped(age VARCHAR(7))

11) correlation between time(input) and hit and run rate
    a) input: time
    b) output: hit and run rate, weather
    c) procedures: HitRun_Bike(t_from NUMERIC(4,1), t_to NUMERIC(4,1)), HitRun_Ped(t_from NUMERIC(4,1), t_to NUMERIC(4,1))

12) which type of location (rural/urban) has more frequent crashes?
    a) input: rural/urban (drop down)
    b) output: hit and run rate, time, alcohol detected, weather, severity
    c) procedures: LocAccidentRate_Bike(loc VARCHAR(5)), LocAccidentRate_Ped(loc VARCHAR(5)) # need to check

13) driver information
    a) input: pedestrian/bike (drop down)
    b) output: summary of driver info: sex, race, age, crash type, driver severity
    c) procedures: DriverInfo(type VARCHAR(10))

14) do crashes happen more often at intersection/non-intersection
    a) input:
    b) output: 1 table (union ped and bike crashes), crash rate (intersection/all, or non-intersection/all), severity, weather, light cond, num_lanes
    c) procedures: IntersectAccidentRate()

15) do crashes happen more often when there's no traffic control?
    a) input:
    b) output: 2 tables, traffic control rate, severity, weather, light cond, num_lanes
    c) procedures: Traffic_Bike(), Traffic_Ped()

4.Relational data model:

- 1.PedCrashRdCond
    - rd_defects,rural_urba,city,locality,rd_feature,light_cond,rd_charact. ,rd_surface, developmen, traff_cntr,rd_conditi,region, rd_class, weather, num_lanes, rd_config

PedCrashID	rd_defects	rural_urba	city	locality	rd_feature	light_cond	rd_charact

1	“None”	“Urban”	“Chapel Hill”	“Urban (>70% Developed)”	“No Special Feature”	“Dark - Roadway Not Lighted”	“Straight - Level”

rd_surface	developmen	traff_cntr	rd_conditi	region	rd_class	weather	num_lanes	rd_config
“Smooth Asphalt”	“Commercial”	“No Control Present”	“Dry”	“Piedmont”	“Public Vehicular Area”	“Clear”	“Unknown”	“Unknown”

- 2.PedInjure
  - for injured people profile
  - ped_pos,ped_race,pedage_grp,ped_age, ped_injury,ped_sex

PedCrashID	ped_pos	ped_race	pedage_grp	ped_age	ped_injury	ped_sex
1	“Non-Roadway - Parking Lot / Other”	“Black”	“25-29”	“29”	“B: Evident Injury”	“Male”

- 3. DiverBiker_PedCrash
  - driver/biker profile
  - speed_limi, drvr_vehty, drvr_injur, drvr_sex, drvrage_gr, drvr_race, drvr_age, drvr_estsp, drvrage_gr, drvr_alc_d

BikeCrashID	drvr_age	drvrage_gr	drvr_estsp	speed_limi	drvr_vehty	drvr_injurdrvr_sex	drvr_alc_d	drvrage_gr	drvr_estsp
1	“46”	“40-49”	“Unknown”	“30 - 35 MPH”	“Passenger Car”	“O: No Injury”	“Female	“White”	“No”

- 4. PedCrashDetail
    - crash detail
    - crsh_sevri,“ambulancer”, crash_time, crash_year, county, longitude,latitude, crash_mont,crash_type,city,locality,ped_pos,drvr_injur,crashday,crash_loc, crash_hour,geo_shape, crash_date, crash_grp,hit_run

PedCrashID	crsh_sevri	ambulancer	crash_time	crash_year	county	longitude	latitude	crash_mont	crash_type
1	“B: Evident Injury”	“Yes”	“1:52”	2007	“Orange”	-79.0	36.0	“November”	“Assault with Vehicle

city	locality	ped_pos	drvr_injur	crashday	crash_loc	crash_hour	geo_shape	crash_date	crash_grp	hit_run
“Chapel Hill”	“Urban (>70% Developed)”	“Non-Roadway - Parking Lot / Other”	“Unknown Injury”	“Saturday”	“Non-Roadway”	1.0	{"type": "Point", "coordinates": [-79.021402733407 97, 35.93761709529 35]}	0000-00-00	“Unusual Circumstances”	“No”

- 5. ReasonPed
    - Bike/Pedestrian Alcohol Detected, Driver Estimated Speed, Speed Limit,Driver Alcohol Detected

BikeCrashID	crashalcoh	excsspdind	ped_pos	drvr_injur	hit_run	drvr_estsp	exceedSpeed	speed_limi
1	“No”	“No”	“Non-Roadway - Parking Lot / Other ”	“Unknown Injury”	“No”	“Unknown”	-1	5

- 6.BikeCrashTime

BikeCrashID	crash_time	crash_hour	crashday	crash_mont	crash_year
1	“10:12”	10.0	“Saturday”	“July”	2011

- 7.BikeCrashLoc

BikeCrashID	lat	lon	county	city	rural_urba	crash_loc	developmen
1	35.9100670923	-79.0745027481	“Orange ”	“Carrboro”	“Urban”	“Non-Intersection ”	“Commercial

- 8.BikeCrashRdCond

BikeCrashID	rd_defects	rd_feature	rd_charact	rd_surface	rd_conditi	speed_limi	traff_cntr	weather	rd_config	num_lanes	developmen	light_cond
1	“Non”e	“No Special Feature”	“Straight - Level”	“Smooth Asphalt ”	“Dry”	“20 - 25 MPH”	“No Control Present”	“Clear”	“Two-Way, Not Divided”	“Unknown”	“Commercial”	“Daylight”

- 9.BikeCrashResult

BikeCrashID	ambulancer	crash_type	crsh_sevri	hit_run	bike_injur	drvr_injur
1	“Yes”	“Motorist Overtaking - Bicyclist Swerved”	“K: Killed”	“No”	“K: Killed”	“O: No Injury”

- 10.Biker

BikeCrashID	bike_injur	bike_race	bike_dir	bikeage_gr	bike_age	bike_sex	bike_pos	bike_alc_d
1	“K: Killed”	“White”	“With Traffic”	“70+”	“70+”	“Male”	“Travel Lane”	“No”

- 11.Driver_BikeCrash

BikeCrashID	drvr_vehty	drvr_injur	drvr_sex	drvr_race	drvr_age	drvrage_gr	drvr_estsp	drvr_alc_d
1	“Passenger Car”	“O: No Injury”	“Male”	“White”	“70+”	“70+”	“11 - 15 mph”	“No”

- 12.BikeCrashReason

BikeCrashID	crashalcoh	excsspdind	drvr_alc_d	bike_alc_d	bike_pos	bike_dir	drvr_estsp	on_rd

5. Sql statements are in Procedures.sql
6. how to load database
   - We first put the variables and their corresponding value in a java object, we populate the data into the database.

7.

- We expect the output to be table of data, and our work includes combine multiple rows of data and present it to the user in a user friendly view. Possibly some design on the table format to make the information easier to read.

8.

- Since we are in section 315, we choose to minorly focus on complex data extraction. So far, our raw data consists of both csv and json files, and we also plan to do some statistical analysis/calculation on these files before we push all of the datasets to the database.