Mahya Jamshidian
9525133
HW4 DB1

1.

```
WITH RECURSIVE PT AS (
SELECT part_id, subpart_id, [count]
FROM subpart
WHERE part_id = 'P100'
UNION ALL
SELECT child.part_id, child.subpart_id, child.count
FROM subpart AS CHILD
JOIN part_tree AS parent ON parent.subpart_id = child.part_id)

SELECT *
FROM part_tree;
SELECT SUM(count)
FROM part_tree;
```

2. In this DB, there are Persons and Cars. People can own one or more cars. Also, each record of an accident consists of all the cars involved, a short description, and the cost.
   a. Meaning that the cost should be extractable whether by drivers, accidents, or cars. Hence, there should be a separate table describing the participation of drivers and cars in a given accident.
   b.

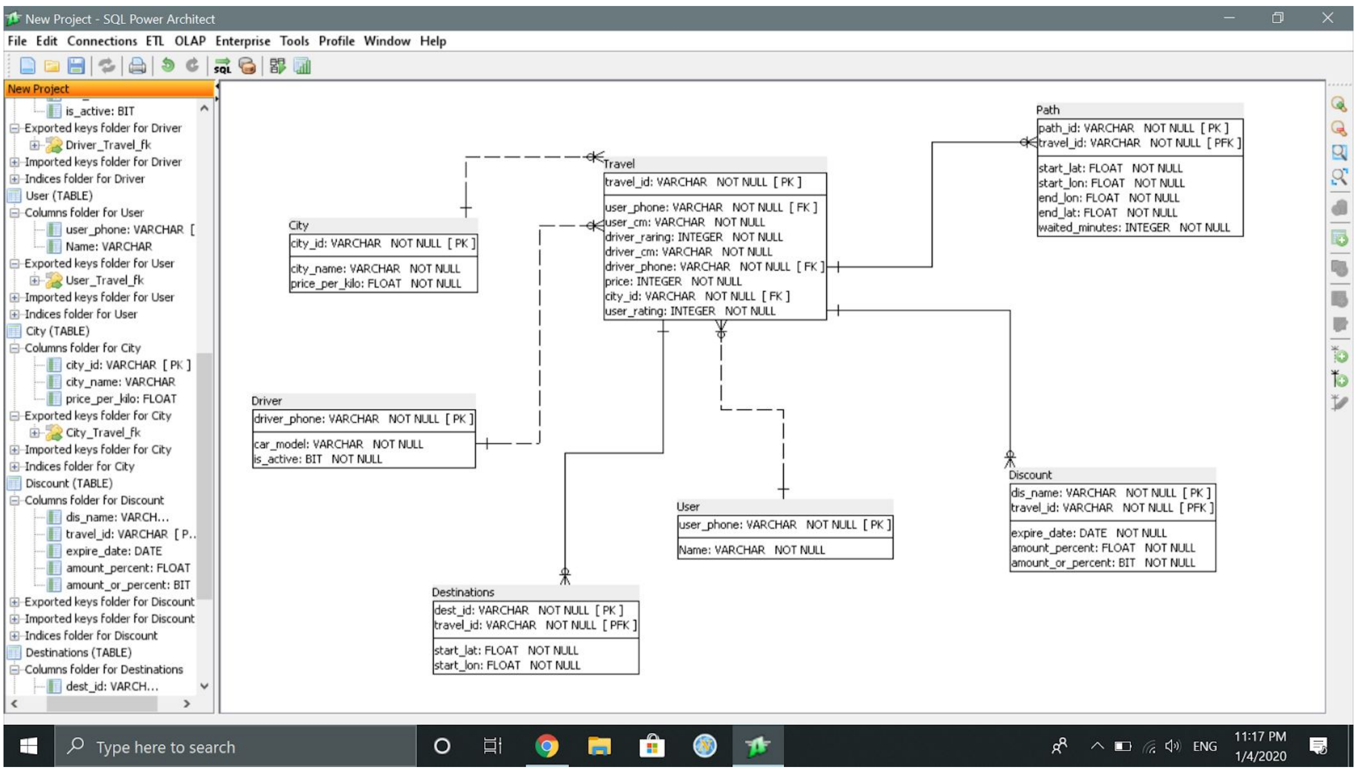
```
Person( driver_id, address, name)
Car( license, model, year)
Accident( report_number, date, location)
Owner( driver_id, license)
AccidentParticipation( report_number, driver_id, license, cost)
```

```
CREATE TABLE Person(
driver_id VARCHAR(20) PRIMARY KEY KEY,
address VARCHAR(200),
name VARCHAR(50) NOT NULL
);
CREATE TABLE Car(
license INT PRIMARY KEY,
model VARCHAR(50),
year DATE
);
CREATE TABLE Accident(
report_number INT PRIMARY KEY,
date DATE,
location VARCHAR(100)
);
CREATE TABLE Owner(
driver_id VARCHAR(20),
license INT,
PRIMARY KEY(driver_id, license),
FOREIGN KEY(driver_id) REFERENCES Person(driver_id) ,
FOREING KEY(license) REFERENCES Car(license)
);
CREATE TABLE AccidentParticipation(
driver_id VARCHAR(20),
license INT,
report_number INT,
damage_amount INT,
PRIMARY KEY(driver_id, license, report_number),
FOREIGN KEY(driver_id) REFERENCES person(driver_id) ,
FOREIGN KEY(license) REFERENCES Car(license),
FOREIGN KEY(report_number) REFERENCES Accident(report_number)
);
```

3. This database is to record academical activities regarding courses and credits across the university, by recording every student's information, his or her taken courses and respective grades, each instructor's teaching course, held courses, and prerequisites for each course.

```
Student( sid, name, program)
Enrols( sid(Student), secno(CourseOffering), grade)
CourseOffering( secno, time, room, year, semester)
Teaches( iid(Instructor), secno(CourseOffering))
Instructor( iid, name, title, dept)
isOffered( secno( CourseOffering), courseno( Course))
Course( courseno, syllabus, title, credits)
Prerequisites( courseno_1( Course), courseno_2( Course))
```

4. Here


New Project - SQL Power Architect
File Edit Connections ETL OLAP Enterprise Tools Profile Window Help
New Project
is_active: BIT
Exported keys folder for Driver
Driver_Travel_fk
Imported keys folder for Driver
Indices folder for Driver
User (TABLE)
Columns folder for User
user_phone: VARCHAR [
Name: VARCHAR
Exported keys folder for User
User_Travel_fk
Imported keys folder for User
Indices folder for User
City (TABLE)
Columns folder for City
city_id: VARCHAR [PK]
city_name: VARCHAR
price_per_kilo: FLOAT
Exported keys folder for City
City_Travel_fk
Imported keys folder for City
Indices folder for City
Discount (TABLE)
Columns folder for Discount
dis_name: VARCH...
travel_id: VARCHAR [P...
expire_date: DATE
amount_percent: FLOAT
amount_or_percent: BIT
Exported keys folder for Discount
Imported keys folder for Discount
Indices folder for Discount
Destinations (TABLE)
Columns folder for Destinations
dest_id: VARCH...
Type here to search
ENG
11:17 PM
1/4/2020

Travel
travel_id: VARCHAR NOT NULL [PK]
user_phone: VARCHAR NOT NULL [FK]
user_cm: VARCHAR NOT NULL
driver_raring: INTEGER NOT NULL
driver_cm: VARCHAR NOT NULL
driver_phone: VARCHAR NOT NULL [FK]
price: INTEGER NOT NULL
city_id: VARCHAR NOT NULL [FK]
user_rating: INTEGER NOT NULL

City
city_id: VARCHAR NOT NULL [PK]
city_name: VARCHAR NOT NULL
price_per_kilo: FLOAT NOT NULL

Path
path_id: VARCHAR NOT NULL [PK]
travel_id: VARCHAR NOT NULL [PFK]
start_lat: FLOAT NOT NULL
start_lon: FLOAT NOT NULL
end_lon: FLOAT NOT NULL
end_lat: FLOAT NOT NULL
waited_minutes: INTEGER NOT NULL

Driver
driver_phone: VARCHAR NOT NULL [PK]
car_model: VARCHAR NOT NULL
is_active: BIT NOT NULL

User
user_phone: VARCHAR NOT NULL [PK]
Name: VARCHAR NOT NULL

Discount
dis_name: VARCHAR NOT NULL [PK]
travel_id: VARCHAR NOT NULL [PFK]
expire_date: DATE NOT NULL
amount_percent: FLOAT NOT NULL
amount_or_percent: BIT NOT NULL

Destinations
dest_id: VARCHAR NOT NULL [PK]
travel_id: VARCHAR NOT NULL [PFK]
start_lat: FLOAT NOT NULL
start_lon: FLOAT NOT NULL

5.

a. All the tables are in the 1NF since none of them contains any multi-valued entry and all the values are single.

   All the tables are in the 2NF since none of them contains attributes that are partially dependent on the primary key.

   All the tables are in the 3NF since all the tables control the data redundancy by an adequate definition of the primary keys.

   All the tables are in the BCNF since the primary key for all of them is the same as their candidate key.

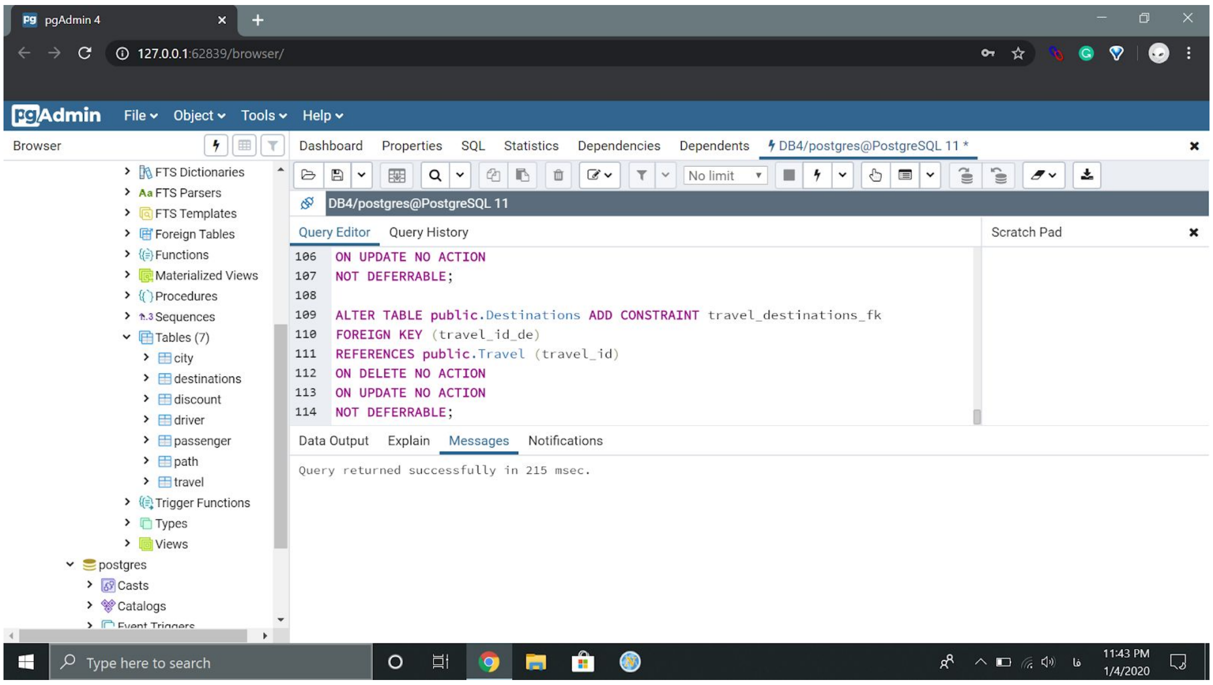
   All the tables are in the 4NF since none of them contains records that can present other records.

b. Transitive Dependency is when two relation p and q are dependant via another relation like r. This form of redundancy is further eliminated during 3NF by removing attributes relying on anything other than the primary key.

c.
   i. Since data is not duplicated, table joins are required. This makes queries more complicated, and thus read times are slower.

ii. Since joins are required, indexing does not work as efficiently. Again, this makes read times slower because the joins don't typically work well with indexing.

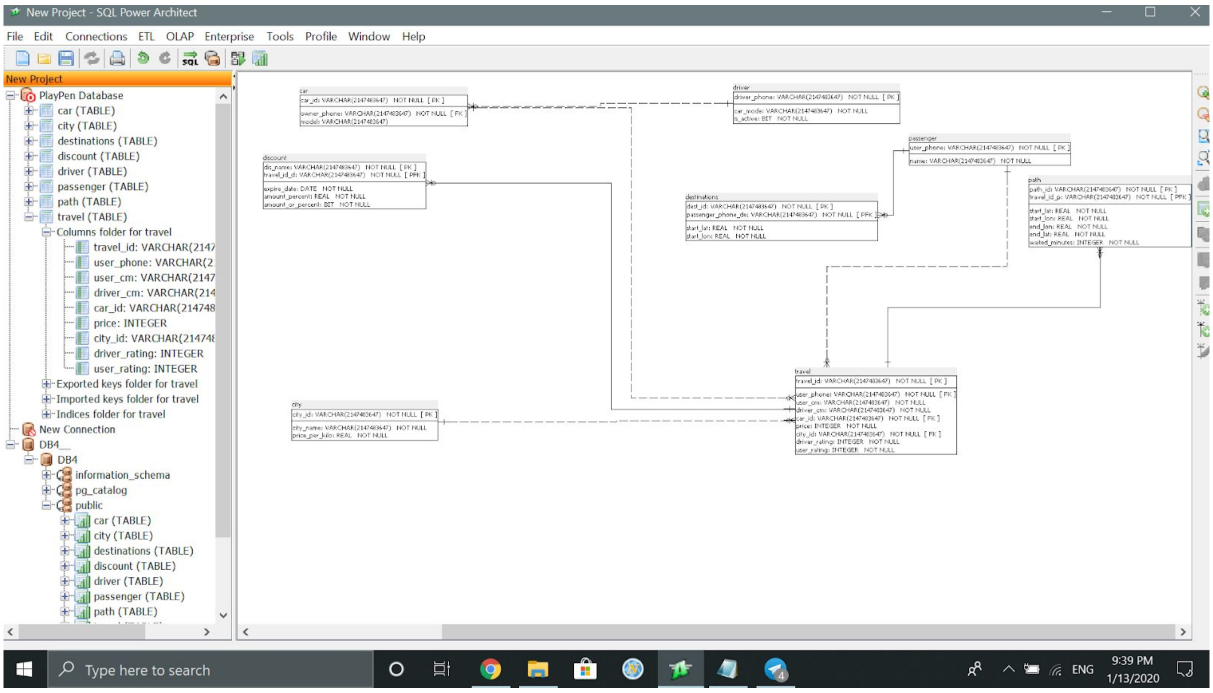
6. The SQL scripts are under the 4.sql file.


```
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.Destinations ADD CONSTRAINT travel_destinations_fk
FOREIGN KEY (travel_id_de)
REFERENCES public.Travel (travel_id)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
Query returned successfully in 215 msec.
```

7. Here



car
car_id: VARCHAR(2147483647) NOT NULL [PK]
owner_phone: VARCHAR(2147483647) NOT NULL [FK]
model: VARCHAR(2147483647)
discount
dis_name: VARCHAR(2147483647) NOT NULL [PK]
travel_id_d: VARCHAR(2147483647) NOT NULL [PFK]
expire_date: DATE NOT NULL
amount_percent: REAL NOT NULL
amount_or_percent: BIT NOT NULL
driver
driver_phone: VARCHAR(2147483647) NOT NULL [PK]
car_mode: VARCHAR(2147483647) NOT NULL
is_active: BIT NOT NULL
passenger
user_phone: VARCHAR(2147483647) NOT NULL [PK]
name: VARCHAR(2147483647) NOT NULL
destinations
dest_id: VARCHAR(2147483647) NOT NULL [PK]
passenger_phone_de: VARCHAR(2147483647) NOT NULL [PFK]
start_lat: REAL NOT NULL
start_lon: REAL NOT NULL
path
path_id: VARCHAR(2147483647) NOT NULL [PK]
travel_id_p: VARCHAR(2147483647) NOT NULL [PFK]
start_lat: REAL NOT NULL
start_lon: REAL NOT NULL
end_lon: REAL NOT NULL
end_lat: REAL NOT NULL
waited_minutes: INTEGER NOT NULL
city
city_id: VARCHAR(2147483647) NOT NULL [PK]
city_name: VARCHAR(2147483647) NOT NULL
price_per_kilo: REAL NOT NULL
travel
travel_id: VARCHAR(2147483647) NOT NULL [PK]
user_phone: VARCHAR(2147483647) NOT NULL [FK]
user_cm: VARCHAR(2147483647) NOT NULL
driver_cm: VARCHAR(2147483647) NOT NULL
car_id: VARCHAR(2147483647) NOT NULL [FK]
price: INTEGER NOT NULL
city_id: VARCHAR(2147483647) NOT NULL [FK]
driver_rating: INTEGER NOT NULL
user_rating: INTEGER NOT NULL

8. here

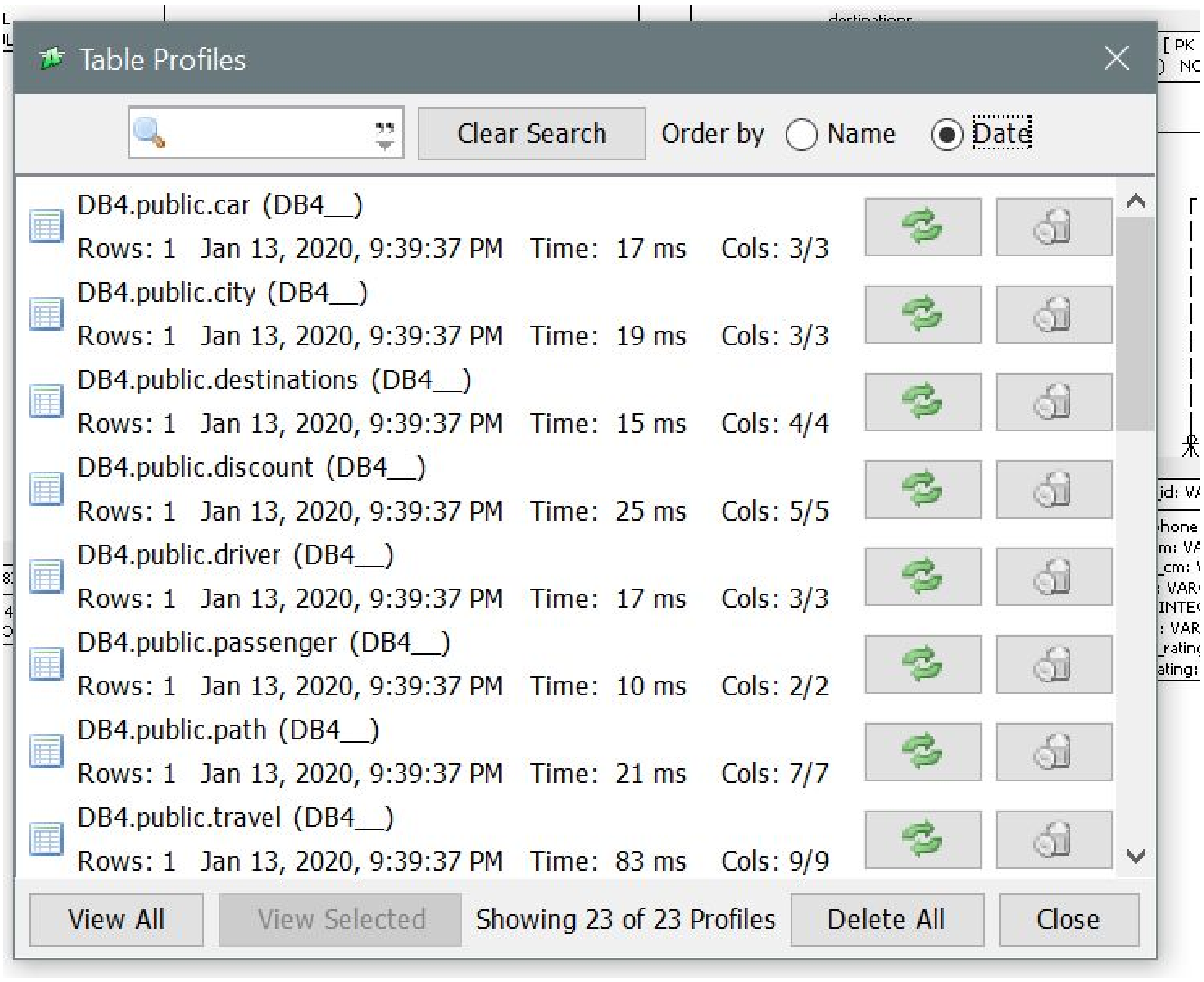


Table Profiles

Clear Search Order by ( ) Name (●) Date

```
DB4.public.car  (DB4__)
Rows: 1   Jan 13, 2020, 9:39:37 PM   Time:  17 ms   Cols: 3/3
DB4.public.city  (DB4__)
Rows: 1   Jan 13, 2020, 9:39:37 PM   Time:  19 ms   Cols: 3/3
DB4.public.destinations  (DB4__)
Rows: 1   Jan 13, 2020, 9:39:37 PM   Time:  15 ms   Cols: 4/4
DB4.public.discount  (DB4__)
Rows: 1   Jan 13, 2020, 9:39:37 PM   Time:  25 ms   Cols: 5/5
DB4.public.driver  (DB4__)
Rows: 1   Jan 13, 2020, 9:39:37 PM   Time:  17 ms   Cols: 3/3
DB4.public.passenger  (DB4__)
Rows: 1   Jan 13, 2020, 9:39:37 PM   Time:  10 ms   Cols: 2/2
DB4.public.path  (DB4__)
Rows: 1   Jan 13, 2020, 9:39:37 PM   Time:  21 ms   Cols: 7/7
DB4.public.travel  (DB4__)
Rows: 1   Jan 13, 2020, 9:39:37 PM   Time:  83 ms   Cols: 9/9
```

View All | View Selected | Showing 23 of 23 Profiles | Delete All | Close

9.