**DADT MIDTERM REPORT**

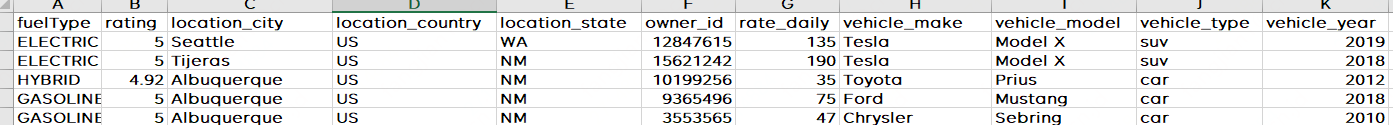
### Part1: About the dataset:

Data Resource: <https://www.kaggle.com/datasets/kushleshkumar/cornell-car-rental-dataset>

* **Quality:** This is an open dataset on car rental information in major US cities in July 2020. It was scraped from the web, but the author didn’t offer its original source. This dataset may not be authoritative and accurate enough, but it is fully capable of being used as a source for machine learning, data analysis, and database testing.
* **Detail:** This dataset has information of More than 5500 Rental Cars, including Make, Model, Year, Fare, Location and Fuel Type.
* **Documentation:** This dataset is standard tabular data. The author has marked the detailed description of the data for each column of data, so that users can accurately grasp the meaning of each data. All the data are marked prominently on the webpage, and there are corresponding text and graphic descriptions. When users encounter problems, it is very easy to find the corresponding reference content.
* **Interrelation:** The data contains the detailed geographic location of the vehicle, including latitude and longitude information. This information can be connected with databases such as Google Maps to obtain accurate positioning. The data also includes city, state, and country information of the city where the vehicle is located, which can be connected with other geographic location (city, state, etc.) databases to obtain similar information such as local population, average income level, and weather etc. More access to data information will allow us to provide more comprehensive and useful data information for further use by database users.
* **Use:** I would like to analyze the preferences of the US car rental market from this database.
  + What is main stream cars by fuel type? Gasoline or electric?
  + Which car is the most popular one? Made by which car maker?
  + Which city/state owns most renting cars?
  + Which car model has the highest renting fee rate?
* **Discoverability:** Car rental is a business. Large car rental companies such as HERZ and DAVIS have more detailed car rental data, but these data involve commercial confidentiality, and unless the above companies voluntarily provide some data sets, it is not easy to obtain.

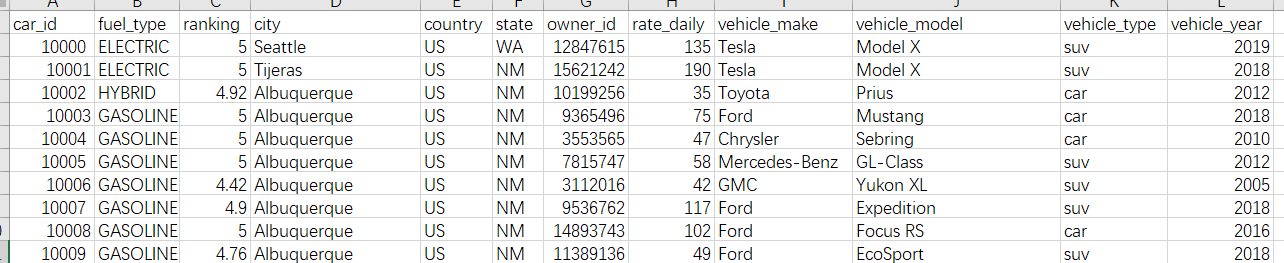
### Part2:Data Modeling

**A glimpse of the data:**



1. This data is almost in 1NF because:
   1. There are no mixing data types within the same column.
   2. There are no repeating rows.
   3. Row order does not convey any information.

But we still need to decide a Primary Key. As each row is about the rental summary of a specific car, we should add a car\_id as the primary key for it, and it is finally in 1NF.



1. To make it 2NF compliant, we need to make sure the non-key attribute in the table depend on the entire primary key.

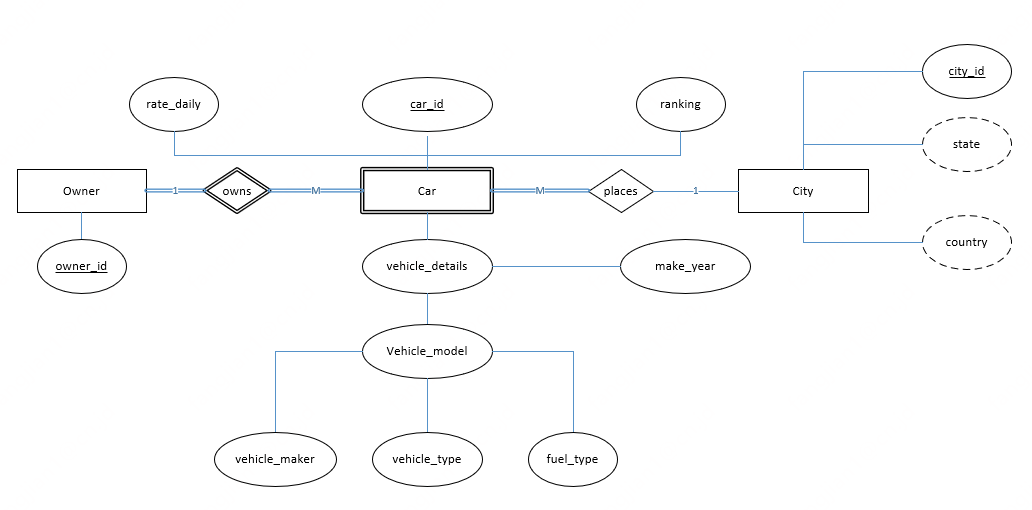
As we have 1 single primary key, the other attributes are indeed dependent on it, so it is in 2NF

1. Let’s move on to see if it is 3NF compliant. We need to make sure every non-key attribute in the table depend on the key, the whole key, and nothing but the key.

And we can see several attributes do not meet this standard in this table.

* 1. We can deduce vehicle\_model from the car\_id, then from the vehicle\_model, we further get information about the vehicle\_maker, vehicle\_type. So these attributes are indirectly dependent on the car\_id. It violates the rule that it must depend on the key, nothing but the key. We can fix it by create a new vehicle\_model table to avoid it.
  2. Each car must locate in some place(here we have City, State,Country). But we can deduce State from City and thus Country from State(not directly from the car\_id). So again it violates the rule. We can fix it by create a City table, State table and a Country(optional as there is only one country involved here) table to avoid the violation.

And we can create E/R model based on the above principles:



It is now 3NF compliant and as there’s no multivalued dependency and each table can not be describe as being the logical result of joining other tables, so there’s no need to further check 4NF and 5NF.

### Part3: Create the database

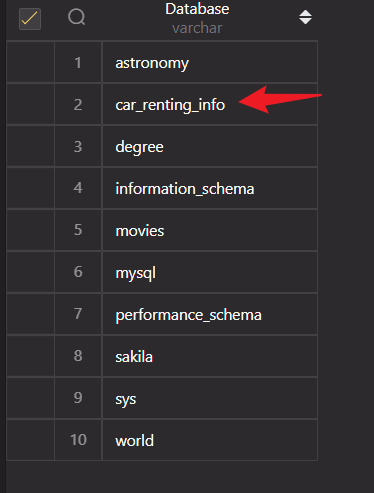
It is time to create the required data table based on the ER model.

**1.Create a new database named car\_renting\_info**



Show databases;





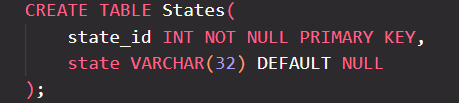
And use it



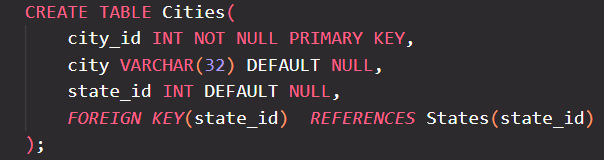
**2. Create 4 tables, named Cars, Vehicle\_details,Cities and States**

As some tables contain foreign keys constraints, so we need to create tables without constraints first.

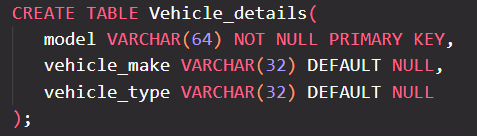
2.01: Create table States, which includes state\_id as PK and state name



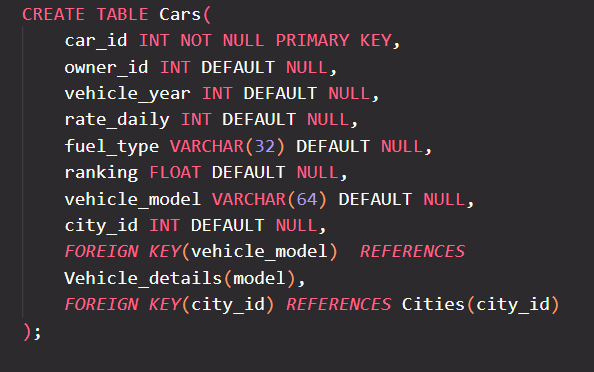
**2.02:Create table Cities, which has city\_id as PK and state\_id as foreign key**



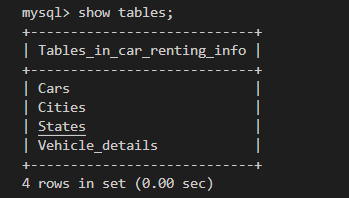
**2.03:Create table Vehicle\_details:**



**2.04:Create table Cars:**

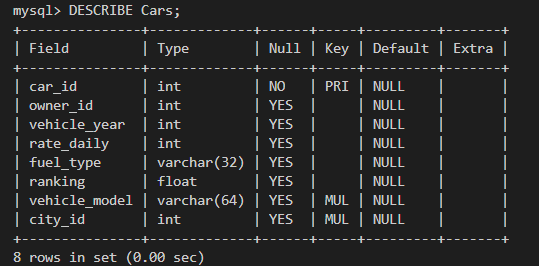


**The final result:**

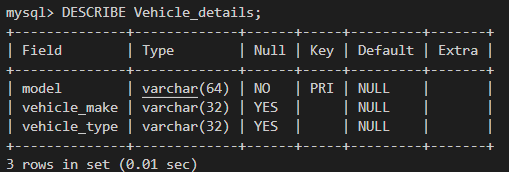


And table description:

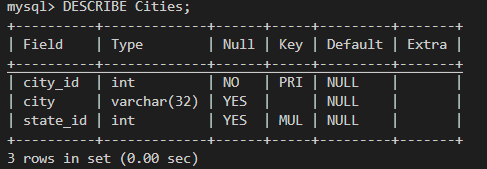
**Cars:**



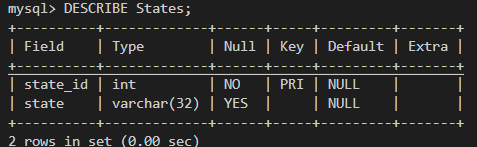
**Vehicle\_details:**



**Cites:**



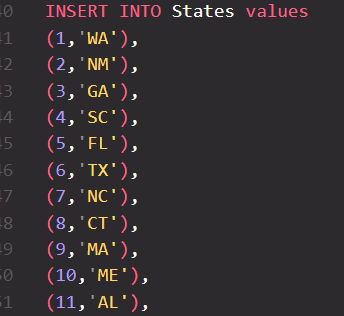
**States:**



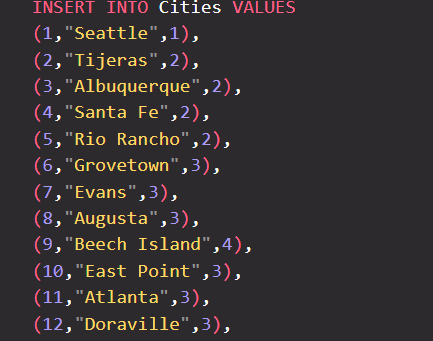
**3.Populate data into the right table.**

There are several ways of populating data into tables, such as using **INSERT** statement or load the .csv file into the table. As we don't have permission to upload files to the cloud project file, here I use INSERT and I give part of the code screenshots of how I did it.

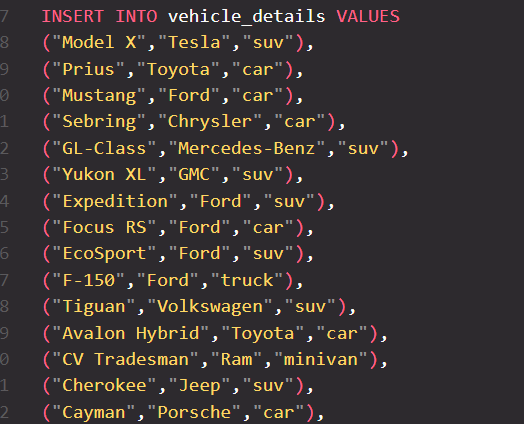
**States table:**



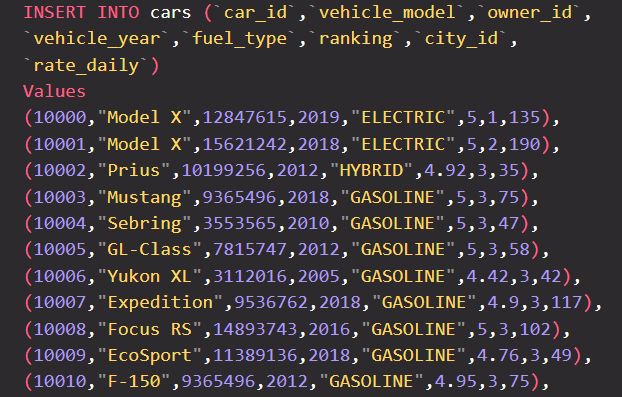
**Cities table:**



**Vehicle\_details table:**



**Cars table:**





**Now let’s answer the questions regarding car renting market with the sql scripts.**

1. **What is main stream cars by fuel type? Gasoline or electric?**

"SELECT COUNT(\*) as car\_counts ,fuel\_type from Cars GROUP BY fuel\_type  ORDER BY car\_counts DESC",

1. **What car model is the most popular? Made by which car maker?**

"SELECT AVG(Cars.ranking) as car\_ranking, Vehicle\_details.vehicle\_make,Cars.vehicle\_model,COUNT(\*) as count FROM Cars INNER JOIN Vehicle\_details ON Cars.vehicle\_model = Vehicle\_details.model GROUP BY Cars.vehicle\_model HAVING count > 50 ORDER BY car\_ranking DESC;",

1. **Which city/state owns most renting cars?**

"SELECT COUNT(\*) as car\_counts,state from Cars INNER JOIN Cities ON Cars.city\_id = Cities.city\_id    INNER JOIN States  ON Cities.state\_id = States.state\_id    GROUP BY States.state    ORDER BY car\_counts DESC     LIMIT 10",

**4. Which car model has the highest renting fee rate?**

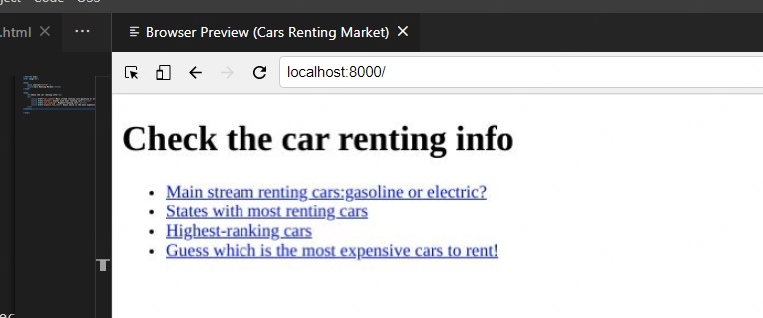
"SELECT AVG(Cars.rate\_daily) as rate\_daily\_avg,Vehicle\_details.vehicle\_make,Vehicle\_details.model FROM Cars INNER JOIN Vehicle\_details ON Cars.vehicle\_model = Vehicle\_details.model GROUP BY Vehicle\_details.model ORDER BY rate\_daily\_avg DESC LIMIT 20;"

I actually would like to do further analysis on the users or owners of the cars, however, as the data is limited(we do not have any user or the owner information), I had to give it up.

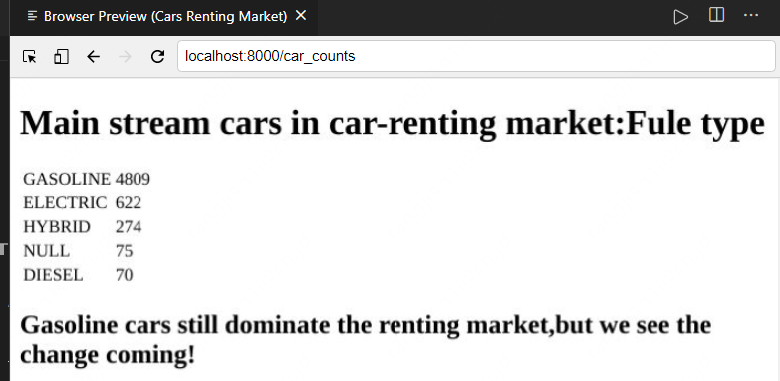
### Part 4. Create a simple web application

I created a main page(default page) for this project. User can click the info they want and it will route to the right page.

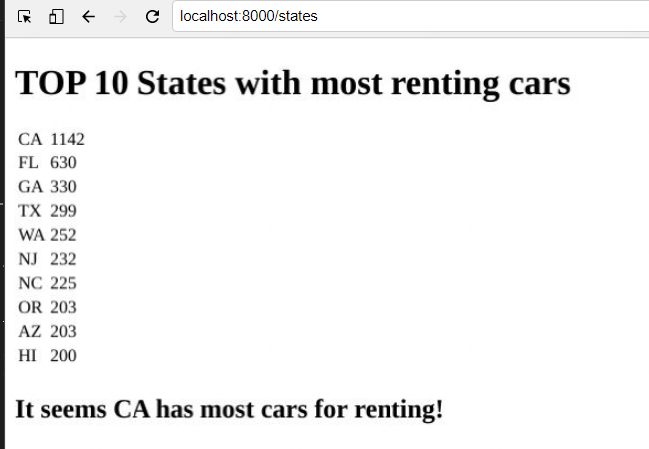
**Default page:**



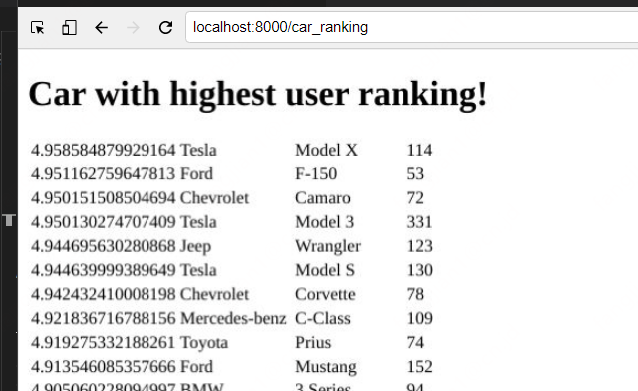
**Main stream Fuel type:**



**States with most cars for renting**



**Most popular cars to rent**

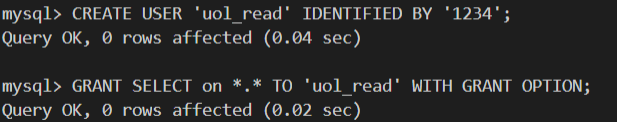


**Most expensive cars to rent**



**Grant only SELECT permission**

The admin user(**root**) has all permissions and can perform operations such as create, add, delete, and update etc. I also created a new account(**uol\_read**), and limit the account permissions to only **SELECT** with **GRANT OPTION**, but not making any changes to the data to assure security of the database and tables.



**Test on UOL LAB with the shared link**

https://hub.labs.coursera.org:443/connect/sharedkartwzya?forceRefresh=false&path=%2F%3Ffolder%3D%2Fhome%2Fcoder%2Fproject

