**Personal Portfolio Template**

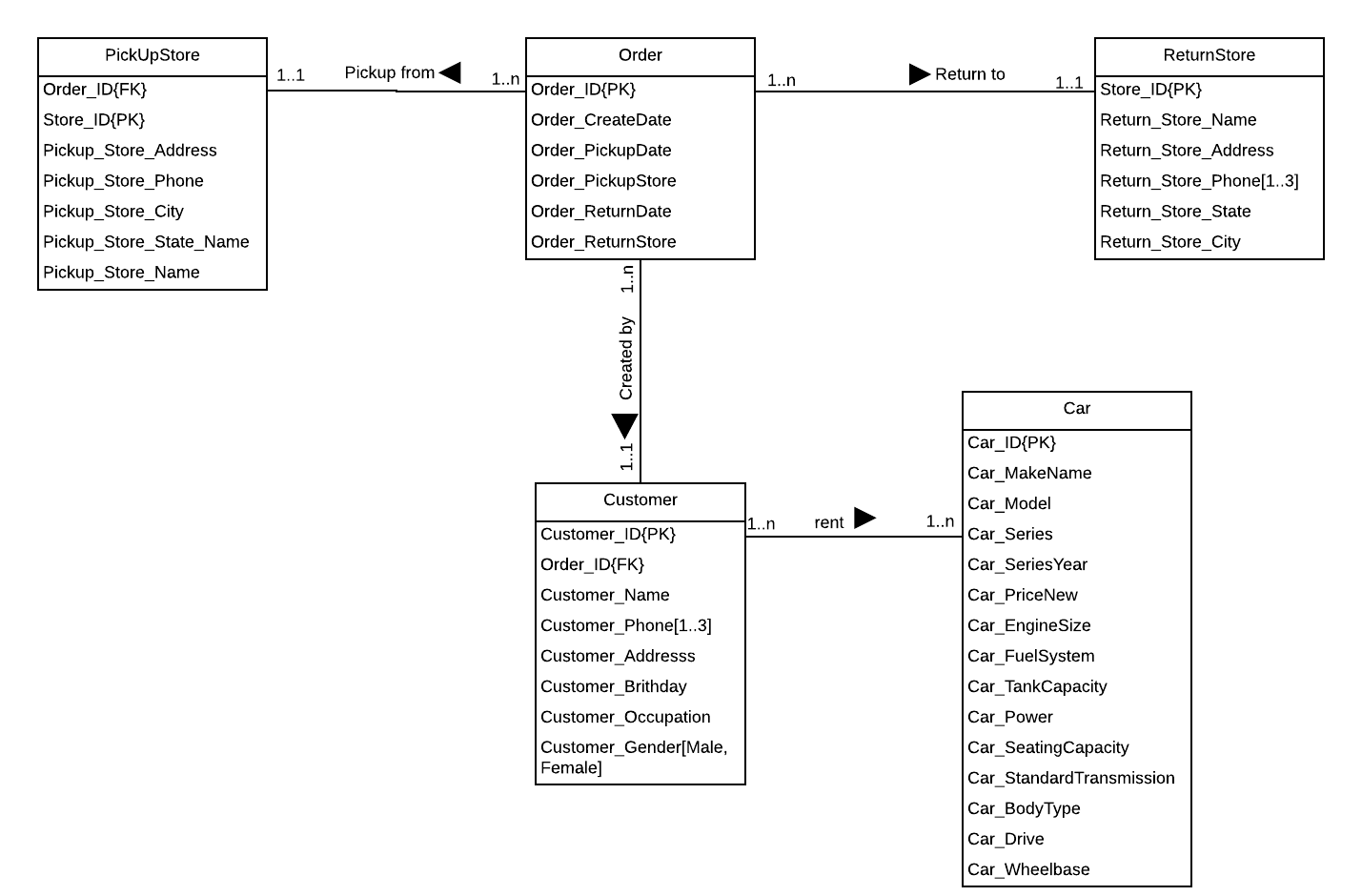
(Group number: 43)

(Anqi Yan– n9919180)

(Github: git clone https://anqiyan@bitbucket.org/anqiyan/ifb299carrental.git)

**Artefact 1 – (Entity Relationship Diagram design)**

This artefact shows work I have done related to database design which is design Entity Relationship Diagram (ERD). ERD is used to show relationship between entities which is very important to build up database, it shows action between entities. For example, order created by Customer then Customer rent the Car. It also shows multiplicity between entities. For instance, each Order can be created by only one customer and each Customer can create more than one Order. ERD also shows attribute domain which limits the data in attribute, for example customer needs at least one and up to three phone number. ERD is very helpful for building database as I mentioned above, with ERD, database can be built very efficiently and conveniently, in this project we need a lot of database, so it is most important part in this project, we will continue to improve this in next few weeks to make it better. (This Artefact is done by myself : Anqi Yan).

****

**Artefact 2 – (Build Database by using Django framework)**

This artefact is related to building database based on Django framework. In this project, we need to create database in order to perform tasks which are required by our customer, so my work in this project is to build database by using Django framework to store data, by using Django framework, I have created some tables such as Order, Customer, and Car by using models built in framework, I have set attributes, default value, data limitations, foreign keys in these tables as well, it is important, because we need a lot interaction between these data in this project, with this artefact, our project can be done based on client’s requirement. We will continue to improve this in next few weeks to make it better. (This Artefact is done by myself: Anqi Yan)

1. We could also use SQL queries to generate table rather than using model, for example, if we want to create customer table, we could use following query:

CREATE TABLE carRental\_Customer(

Customer\_id int,

CustomerName varchar (255),

PhoneNumber varchar(15),

Address varchar (255),

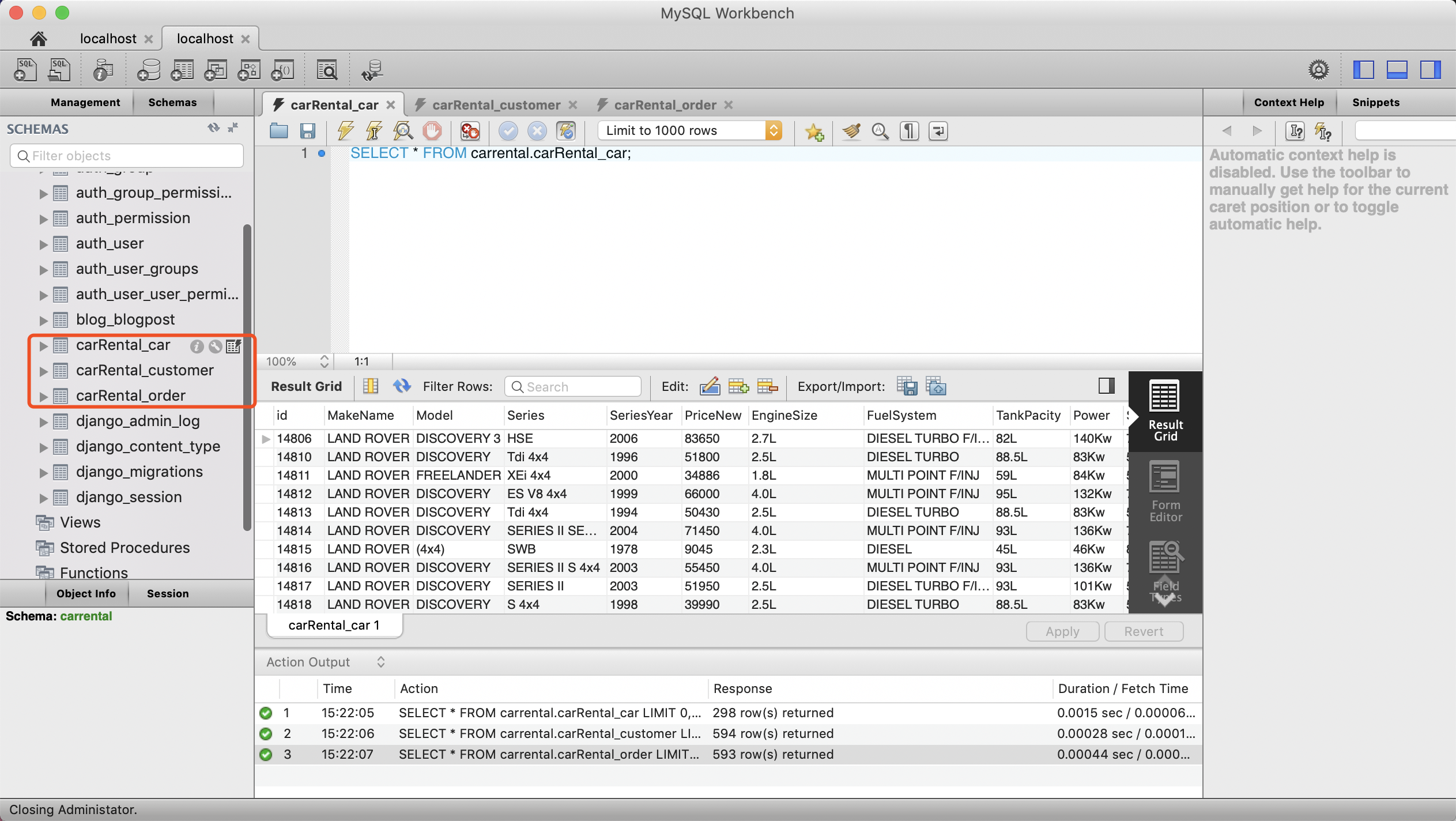
Birthday varchar (8),

Occupation varchar(20),

Gender char (1) enum(‘M’, ‘F’),

)

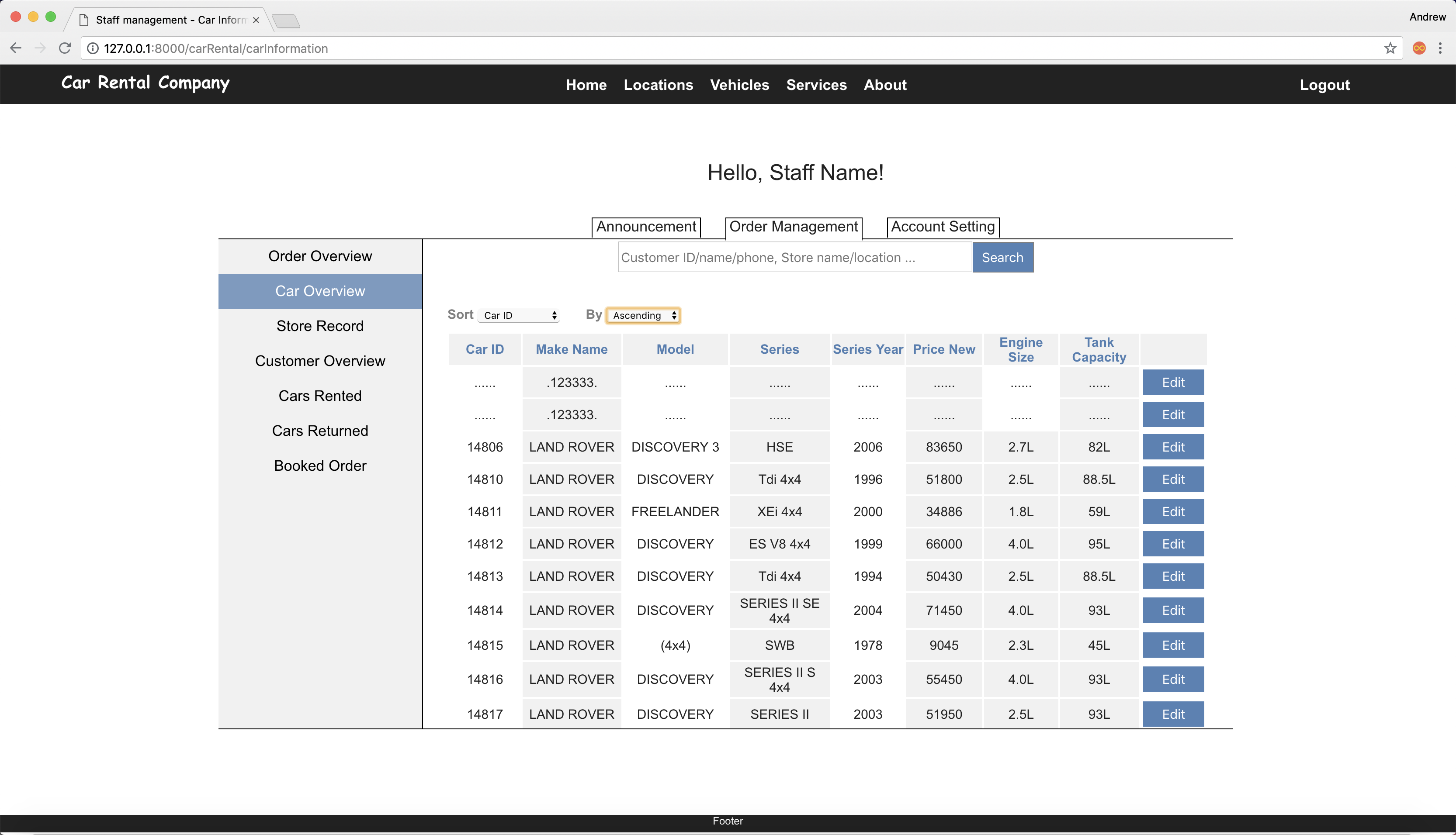
(Models.py)



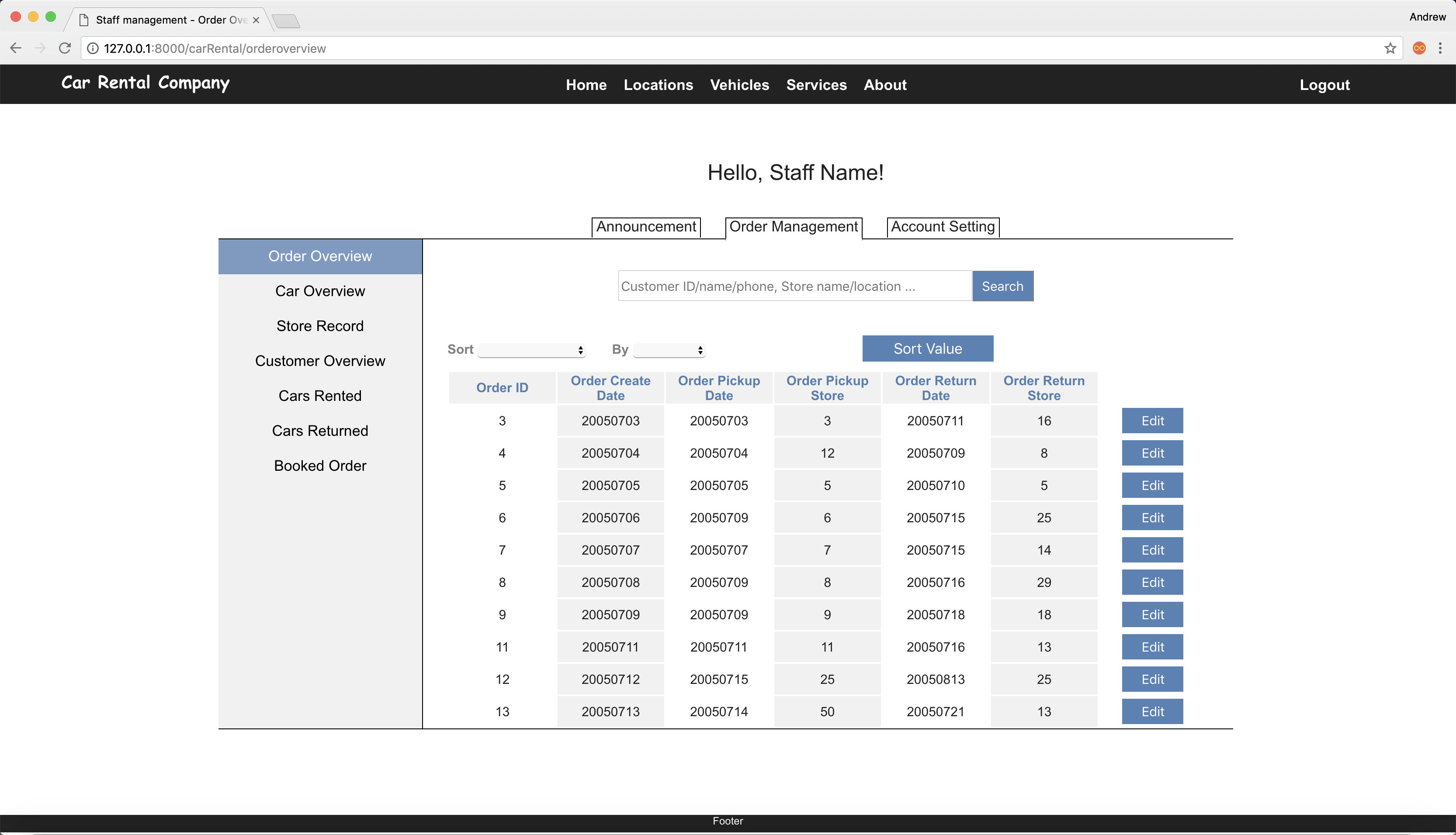
(Database in MySQL Workbench)

**Artefact 3 – (Retrieve Data From database and show in the page based on Django Framework)**

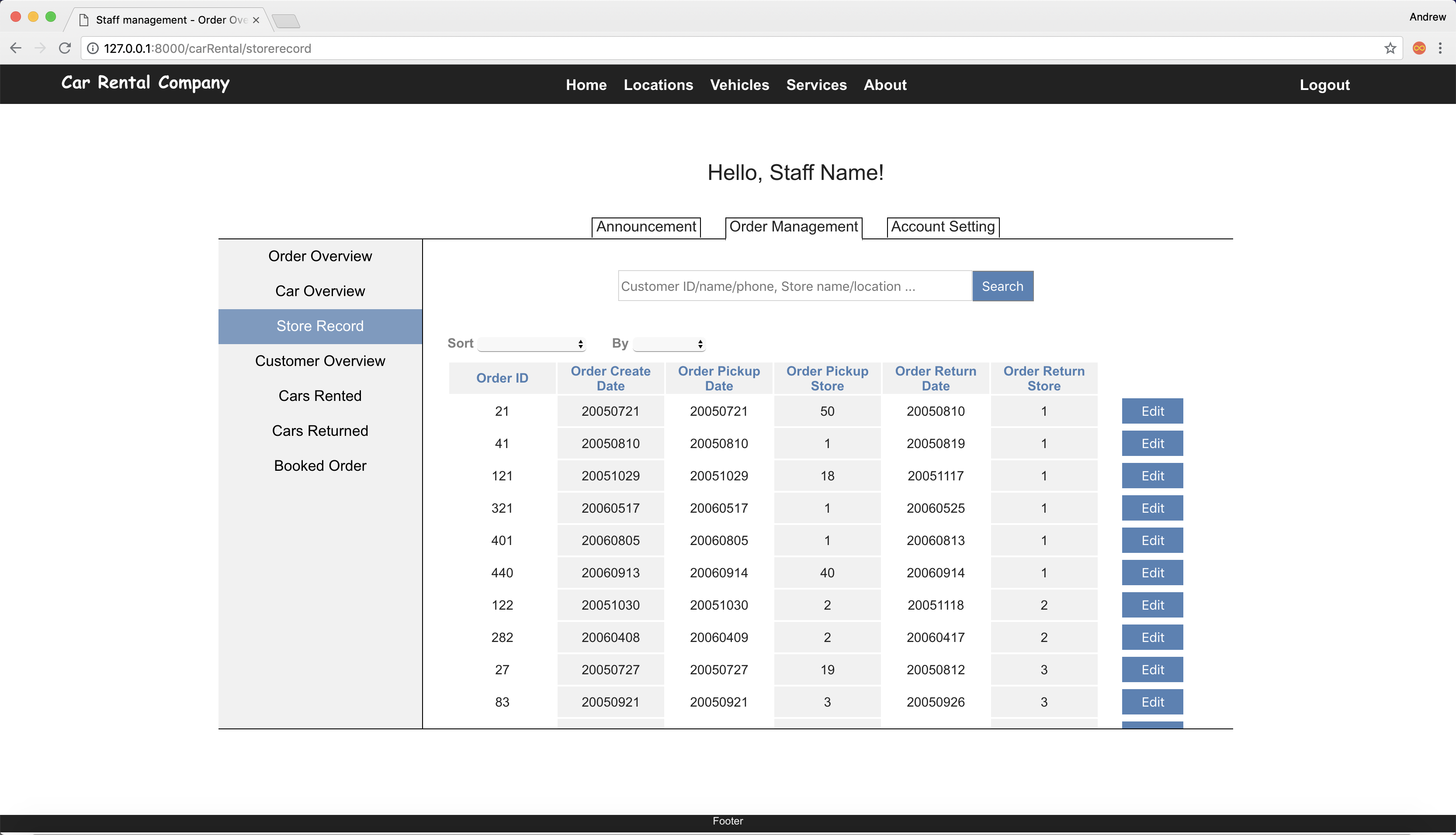
This artefact shows work I have done which is retrieve data from database and display in different pages (Order page, Customer detail page, Car information page, we will create more data in the next few weeks \**not including layout design*). In this artefact i have created some pages to show relative data based on requirement from our customer for them to do further data analysis, each page contains some data based on selection, there are several selections: Order Overview, Car overview, Store Record, Customer overview, Cars rented, Cars returned, booked order. It is easy to check data, for example, if we select Order overview, page will show all of order in the database. This artefact is important because in this project our customer wants to check data and to do further analysis, with this artefact, our team can do further development based on this artefact. (Data part in this Artefact is done by me \*not including layout design: Anqi Yan)

****

**Car Overview**

****

**(Order Overview)**

**(Sort by store)**

**Artefact4(SQL queries)**

This artefact I have done is related to SQL queries, it contains three main parts, input data, output data and test data. This artefact is important in this project, for example, for data input, our client may have new cars and want to store that information in the database, with these SQL queries, they can easily input information in the text fields and click submit button, then page will execute SQL queries and data will be stored in the database. I have created some INSERT queries following to do required tasks: Insert value into car table, insert value into Order table and Insert value into customer table. For data output, our client want to generate report based on some specific values, so I have made some queries to get data from database based on client’s requirements, for example, get how many types of cars have been picked up from specific store in specific month, which is 4th SQL queries(SELECT COUNT(Car\_ID), Return\_Store\_City FROM carrental.carRental\_order WHERE Order\_ReturnDate > 20050700 and Order\_ReturnDate < 20050800 and Return\_Store\_City = “Sydney” group by Return\_Store\_City Order by COUNT(Car \_ID) asc; Third part of SQL queries are related to test data, I have designed to insert different type of invalid value and all correct value into each table to check if database is been set correctly so that all of data in our database can be stored correctly. (List of SQL Queries following is done by myself: Anqi Yan).

**SQL Queries used to generate required results**

1. CREATE TABLE carRental\_Customer(

Customer\_id int,

CustomerName varchar (255),

PhoneNumber varchar(15),

Address varchar (255),

Birthday varchar (8),

Occupation varchar(20),

Gender char (1) enum(‘M’, ‘F’),

) (Example of using SQL queries to create table)

1. INSERT INTO carrental.carRental\_car(MakeName, Model, Series, SeriesYear, PriceNew, EngineSize, FuelSystem, TankPacity,Power, SeatingCapacity, StandardTransmission, BodyType, Drive, WheelBase) VALUES(‘LAND ROVER’, ‘4\*4’, ‘SWB’, ‘1978’, ‘54204’, ’4.0L’, ‘MULTI POINT F/INJ’, ‘150KW’, ‘7’, ‘4A’, ‘4D WAGON’, ‘4WD’, ‘2800mm’); (Insert value into car table with values. \*this is for company management to insert values to Car table)
2. INSERT INTO carrental.carRental\_order(Order\_ID, Order\_CreateDate, Order\_PickupDate, Order\_PickupStore, Order\_ReturnDate, Order\_ReturnStore) VALUES (‘3’, ‘20050703’, ‘3’, ‘20050711’, ‘16’) (Insert value into order table with values. \*this is for company management to insert values to Order table)
3. INSERT INTO carrental.carRental\_customer(Customer\_id, CustomerName, PhoneNumber, Address, Birthday, Occupation, Gender) VALUES (‘11014’, ‘Sydney B’, ‘431-432-133’, ’20 Armentieres Street’, ‘08/05/1980’, ‘Labour’, ‘M’).( Insert value into Customer table with values. \*this is for company management to insert values to Customer table)
4. SELECT COUNT(Car\_ID), Return\_Store\_City FROM carrental.carRental\_order WHERE Order\_ReturnDate > 20050700 and Order\_ReturnDate < 20050800 and Return\_Store\_City = “Sydney” group by Return\_Store\_City Order by COUNT(Car \_ID) asc; (Count how many order returned to specific store monthly. \*this is for company management to get monthly report based on which store)
5. SELECT COUNT(Order\_ID), Pickup\_Store\_City FROM carrental.carRental\_order WHERE Order\_PickupDate > 20050700 and Order\_ PickupDate < 20050800 and Pickup\_Store\_City = “Sydney” group by Pickup \_Store\_City Order by COUNT(Order\_ID) asc; (Count how many order picked up from specific store monthly. \*this is for company management to get monthly report based on which store)
6. SELECT COUNT(Order\_ID), Order\_PickupStore FROM carrental.carRental\_order GROUP By Order\_PickupStore order by COUNT(Order\_ID) asc; (Check how many record picked up in each store in total and order by ascending sequence. \*this is for company management to know which store has more customer to pick up the car)
7. SELECT \* FROM carrental.carRental\_car WHERE WheelBase >2800 and SeatingCapacity >5; (Get car information which has more space and more seat. \*this is for customer who want more space and more people to use)
8. SELECT COUNT(Car\_BodyType), Order\_PickupStore, FROM carrental.carRental\_order WHERE Order\_PickupStore = “Sydney” and Order\_PickupDate > 20050700 and Order\_ PickupDate < 20050800 GROUP By Order\_PickupStore order by COUNT(Car\_BodyType) asc;( \*this is for customer recommend service enables to recommend customer which car type is most popular in which store monthly)
9. SELECT DISTINCT \* FROM carrental.carRental\_car; (Used DISTINCT to eliminate duplicates records in the data. \*this is for company management to eliminate invalid data record)
10. SELECT \* FROM carrental.carRental\_order WHERE Order\_PickupStore = 1 (Check the record of pick up store. \*this is for company management to check data based on which store);

**SQL queries to Test Data**

1. INSERT INTO carrental.carRental\_car(MakeName, Model, Series, SeriesYear, PriceNew, EngineSize, FuelSystem, TankPacity,Power, SeatingCapacity, StandardTransmission, BodyType, Drive, WheelBase) VALUES(‘LAND ROVER’, ‘4\*4’, ‘SWB’, ‘1978’, ‘54204’, ’4.0L’, ‘MULTI POINT F/INJ’, ‘150KW’, ‘7’, ‘4A’, ‘4D WAGON’, ‘4WD’, ‘2800mm’); (Test if all correct value can be insert to table or not)
2. INSERT INTO carrental.carRental\_car(MakeName, Model, Series, SeriesYear, PriceNew, EngineSize, FuelSystem, TankPacity,Power, SeatingCapacity, StandardTransmission, BodyType, Drive, WheelBase) VALUES(‘@#$$’, ‘4\*4’, ‘NULL’, ‘1978’, ‘54204’, ’4.0L’, ‘MULTI POINT F/INJ’, ‘150’, ‘7’, ‘4A’, ‘4D WAGON’, ‘4WD’, ‘2800mm’); (Test if some incorrect value can be insert to table or not).
3. INSERT INTO carrental.carRental\_car(MakeName, Model, Series, SeriesYear, PriceNew, EngineSize, FuelSystem, TankPacity,Power, SeatingCapacity, StandardTransmission, BodyType, Drive, WheelBase) VALUES(‘LAND ROVER’, ‘ ‘ , ‘SWB’, ‘1978’, ‘54204’, ’4.0L’, ‘MULTI POINT F/INJ’, ‘150KW’, ‘7’, ‘4A’, ‘ ’, ‘ ‘, ‘ ’); (Check if empty value can be set to ‘NULL’ automatically in the table or not)
4. INSERT INTO carrental.carRental\_car(MakeName, Model, Series, SeriesYear, PriceNew, EngineSize, FuelSystem, TankPacity,Power, SeatingCapacity, StandardTransmission, BodyType, Drive, WheelBase) VALUES(‘&\*(#’, ‘$$@S’, ‘!@#’, ‘1978’, ‘54204’, ’4.0L’, ‘MULTI POINT F/INJ’, ‘150KW’, ‘#$@’, ‘4A’, ‘4D WAGON’, ‘4WD’, ‘2800mm’); (Test if some incorrect value can be insert into table or not).
5. INSERT INTO carrental.carRental\_order(Order\_ID, Order\_CreateDate, Order\_PickupDate, Order\_PickupStore, Order\_ReturnDate, Order\_ReturnStore) VALUES (‘3’, ‘20050703’, ‘3’, ‘20050711’, ‘16’) (Test if all correct value can be insert into table or not)
6. INSERT INTO carrental.carRental\_order(Order\_ID, Order\_CreateDate, Order\_PickupDate, Order\_PickupStore, Order\_ReturnDate, Order\_ReturnStore) VALUES (‘ ’, ‘20050703’, ‘3’, ‘ ‘, ‘16’)(Test if empty value can be set to ‘NULL’ automatically into the table or not)
7. INSERT INTO carrental.carRental\_order(Order\_ID, Order\_CreateDate, Order\_PickupDate, Order\_PickupStore, Order\_ReturnDate, Order\_ReturnStore) VALUES (‘@@#$’, ‘@$#@@, ‘3’, ‘, ‘16’) (Test if some incorrect value can be insert into table or not).
8. INSERT INTO carrental.carRental\_order(Order\_ID, Order\_CreateDate, Order\_PickupDate, Order\_PickupStore, Order\_ReturnDate, Order\_ReturnStore) VALUES (‘ABC’, ‘20050703’, ‘3’, ‘20050711’, ‘16’) (Test if incorrect integer value can be insert into table or not).
9. INSERT INTO carrental.carRental\_customer(Customer\_id, CustomerName, PhoneNumber, Address, Birthday, Occupation, Gender) VALUES (‘11014’, ‘Sydney B’, ‘431-432-133’, ’20 Armentieres Street’, ‘08/05/1980’, ‘Labour’, ‘M’).(Test if all correct value can be insert into table or not)
10. INSERT INTO carrental.carRental\_customer(Customer\_id, CustomerName, PhoneNumber, Address, Birthday, Occupation, Gender) VALUES (‘#$#@’, ‘Sydney B’, ‘@#)\*UI’, ’20 Armentieres Street’, ‘19850830’, ‘Labour’, ‘M’).(Test if some incorrect value can be insert into table or not)
11. INSERT INTO carrental.carRental\_customer(Customer\_id, CustomerName, PhoneNumber, Address, Birthday, Occupation, Gender) VALUES (‘’, ‘Sydney B’, ‘431-432-133’, ’20 Armentieres Street’, ‘19850830’, ‘Labour’, ‘M’).(Test if Customer\_id can be generated automatically or not)
12. INSERT INTO carrental.carRental\_customer(Customer\_id, CustomerName, PhoneNumber, Address, Birthday, Occupation, Gender) VALUES (‘’, ‘ ’, ‘ ’, ’ ’, ‘19850830’, ‘Labour’, ‘M’).(Test if empty value can be set to “NULL” or not)

(Continue…)

Artefact5(UI Design)

This Artefact is related to UI design, it shows what user interface I have designed in our project. In this artefact, I have designed several pages: Car information overview page, Store order overview page, store record page, customer information page and staff management page. Each page shows which functionalities we have with annotations. It is important for further development because our team can develop project based on these UI design and see interaction between pages. (Following pages are done by myself)

****

****

