Database design Project Presentation:

Rentalcars.com

Agenda

Company Background
Business Objectives
Scope of Project
Functionality Requirements
External Views
Business Rules and Assumptions
ER Diagram
SQL Database (Table Design, Relational Database, Integrity Control, Sample Data)

Company Background

- TravelJigsaw, 2004
- 1000 Bookings Per Day Within 4 Years
- Join Priceline Group, 2010
  - Rebranding
- Today:
  - 8 Million Bookings Per Day
  - 160 Countries
  - 60 000 Locations

Business Objectives

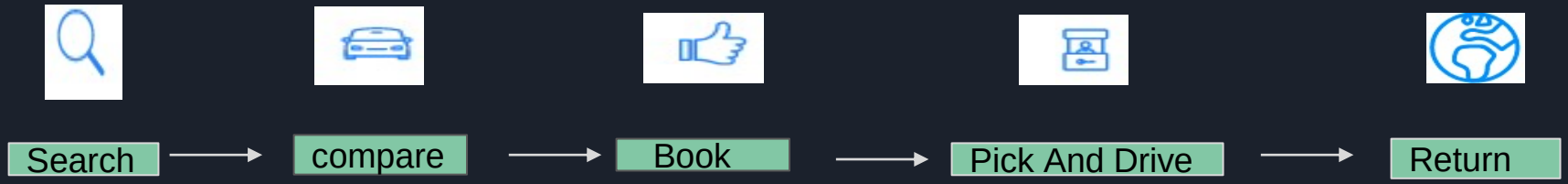
- Act as an intermediary between car rental companies and customers
- Database to help customers find optimal rental vehicle
  - Make/Model
  - Price
  - Location
- Customer reviews and feedback

Scope of Project

Create a database that will store rental car reservation orders as users rent cars through the website.

- **User Profile**
  - Name
  - Payment Information
  - Phone
  - Email
- **Selecting the Car Through a Search**
  - Make/model
  - Price
  - Supplier
- **Reserving a Car**
  - User Profile
  - Pick up/Drop off
- **Rating/Review**
  - Supplier/Car
  - RentalCars.com

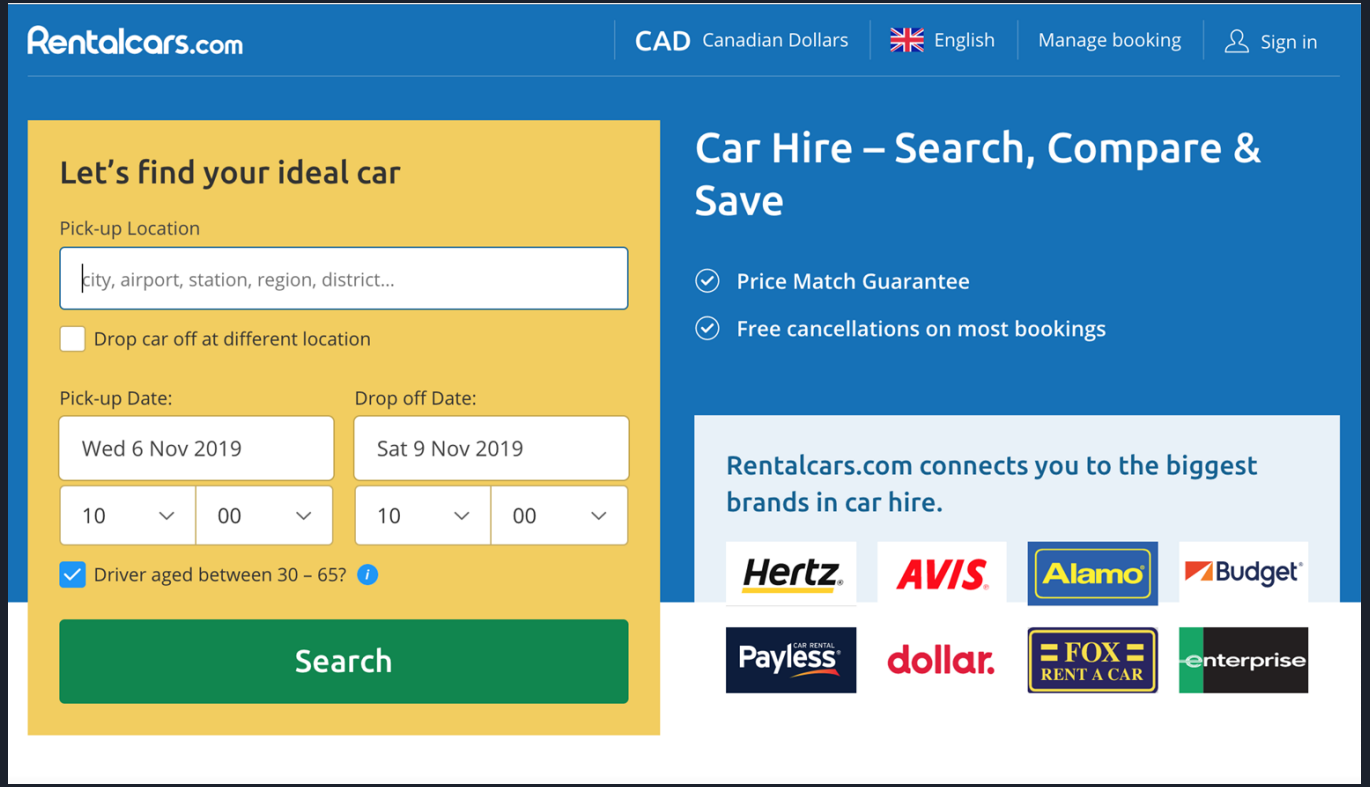
Functional Requirements



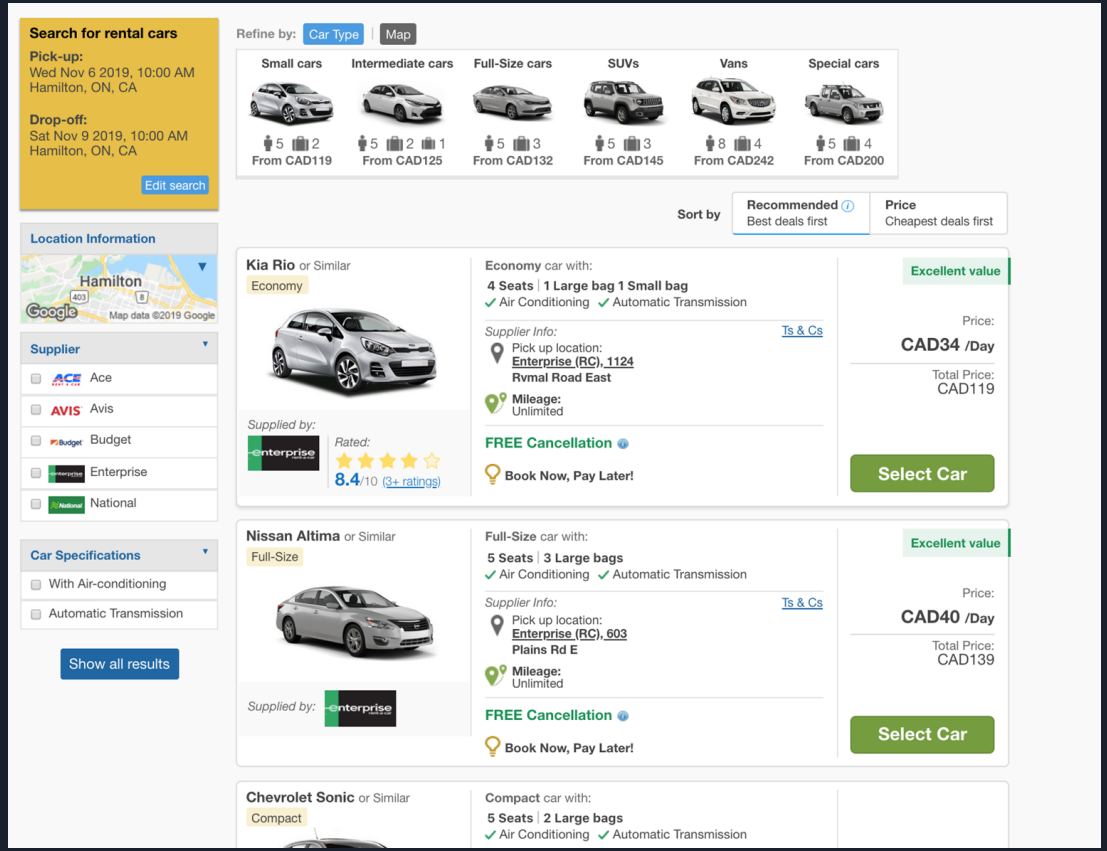
- Provide available rental dates to the customer
- As an agent between customer and car hire
- Payment options and retain voucher dates

24/7 Support

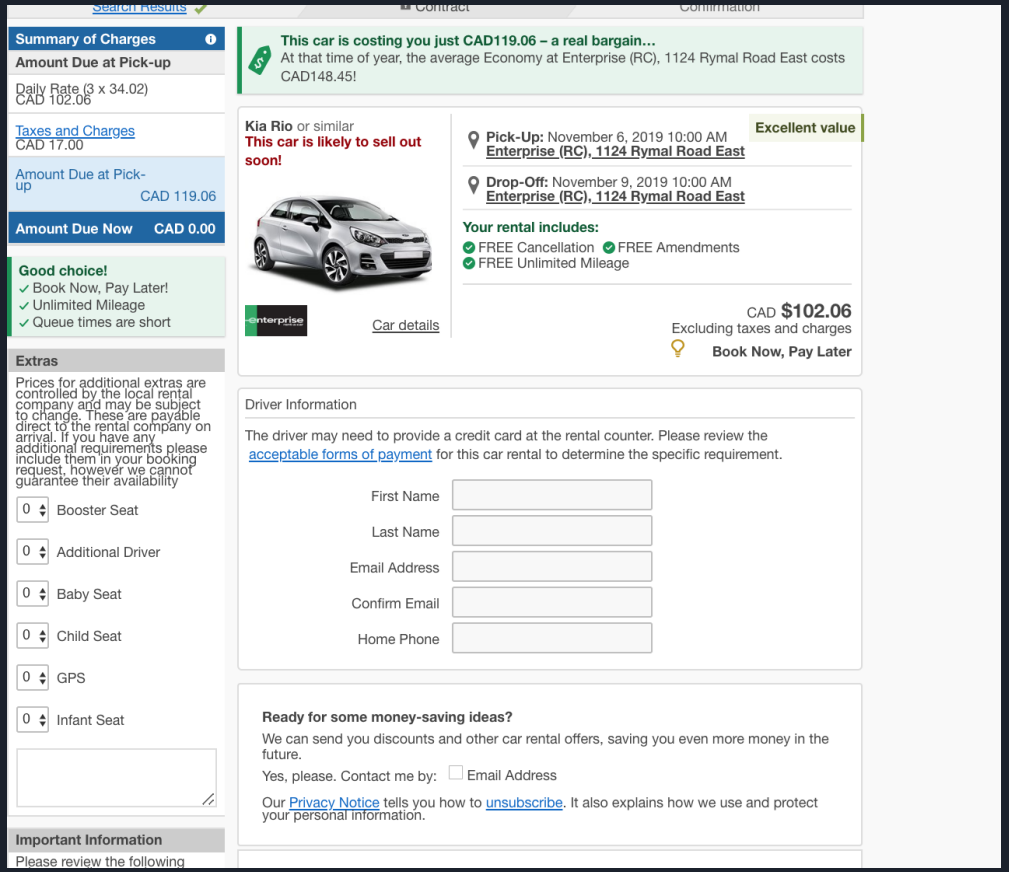
External Views



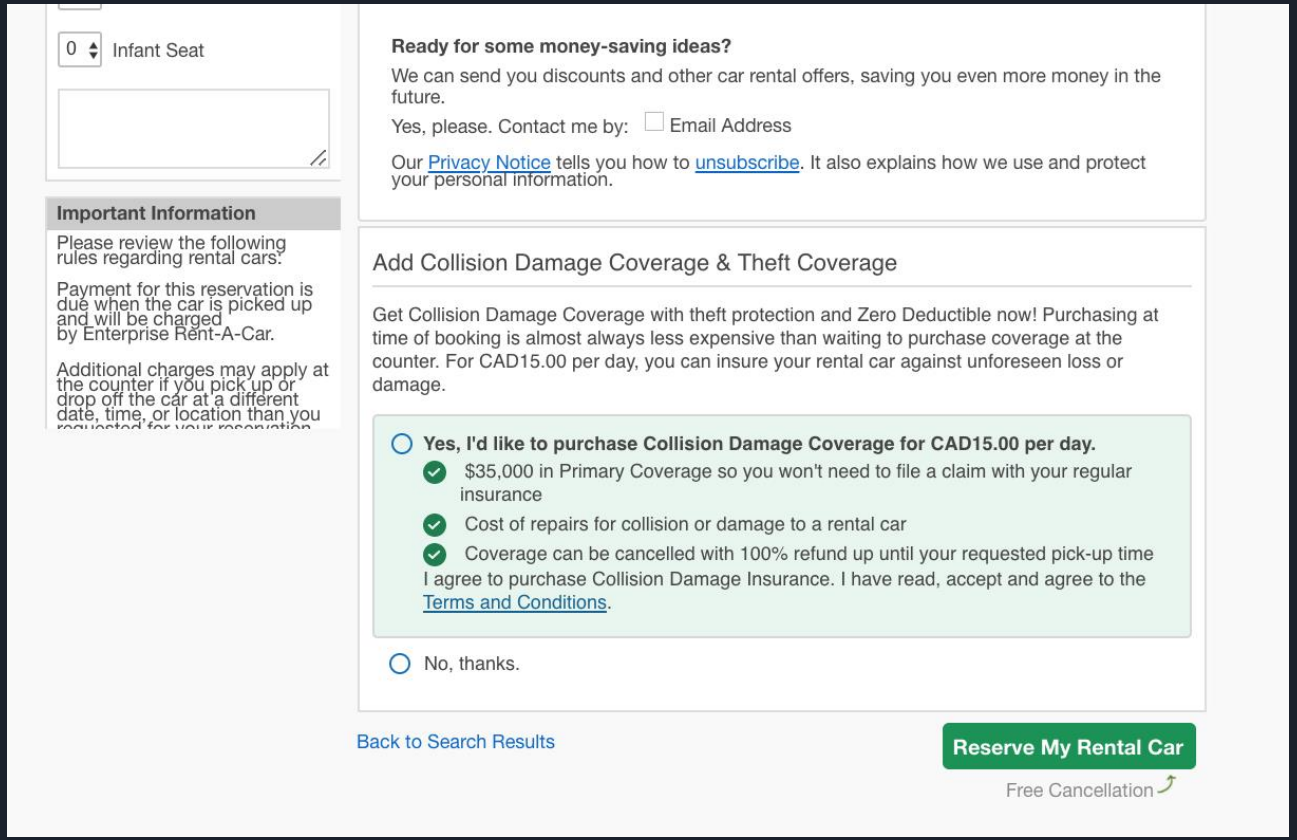
External Views



External Views



External Views



Business Rules/Assumptions

Website

1. Customers will be provided with a car by Website either directly or as an agent.

2. one user may use the website to find a right car base on their expectation

3.The site is available in 40 languages; the user can clear understanding of their required information

4.Rentalcars implement the price match guarantee for the customers.

Customer

1. To book a car, all customers need is a credit or debit card, but the website voucher must be presented to the car hire company when customers picking up the hire car.

2.Every driver must have a full, valid driving licence they have held for at least one year (minimum age is 21 years old or above)

ER Diagram

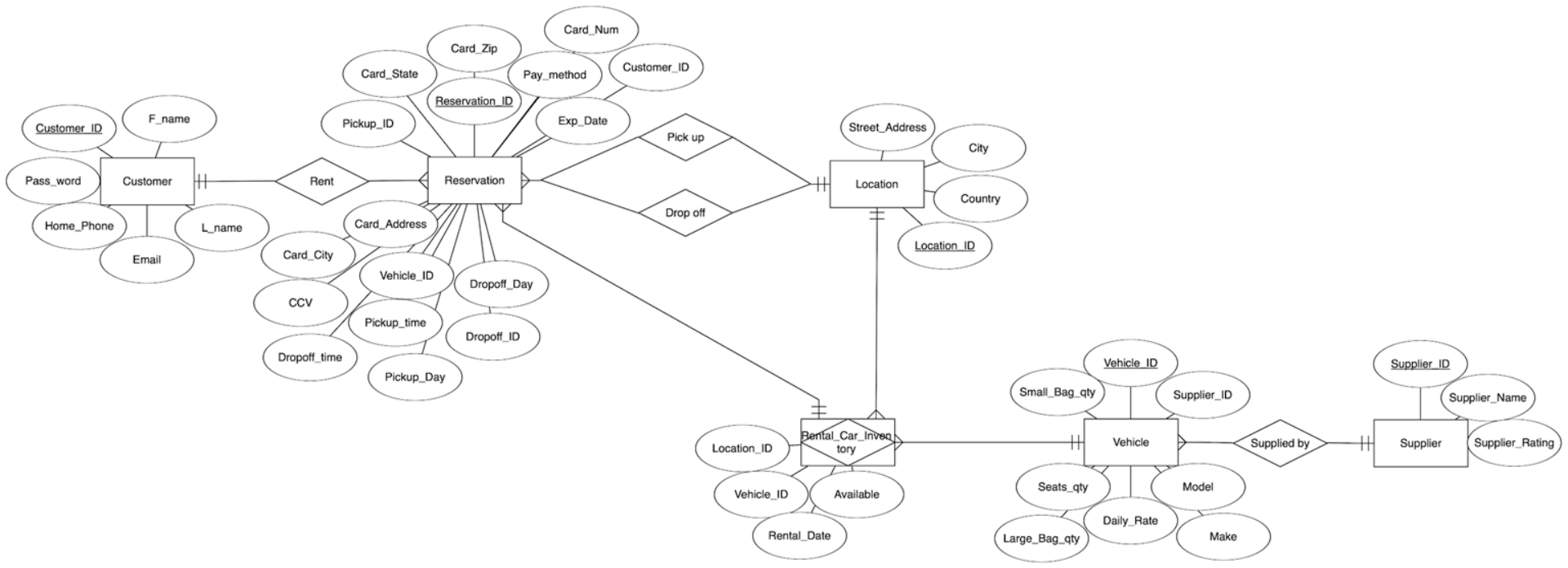
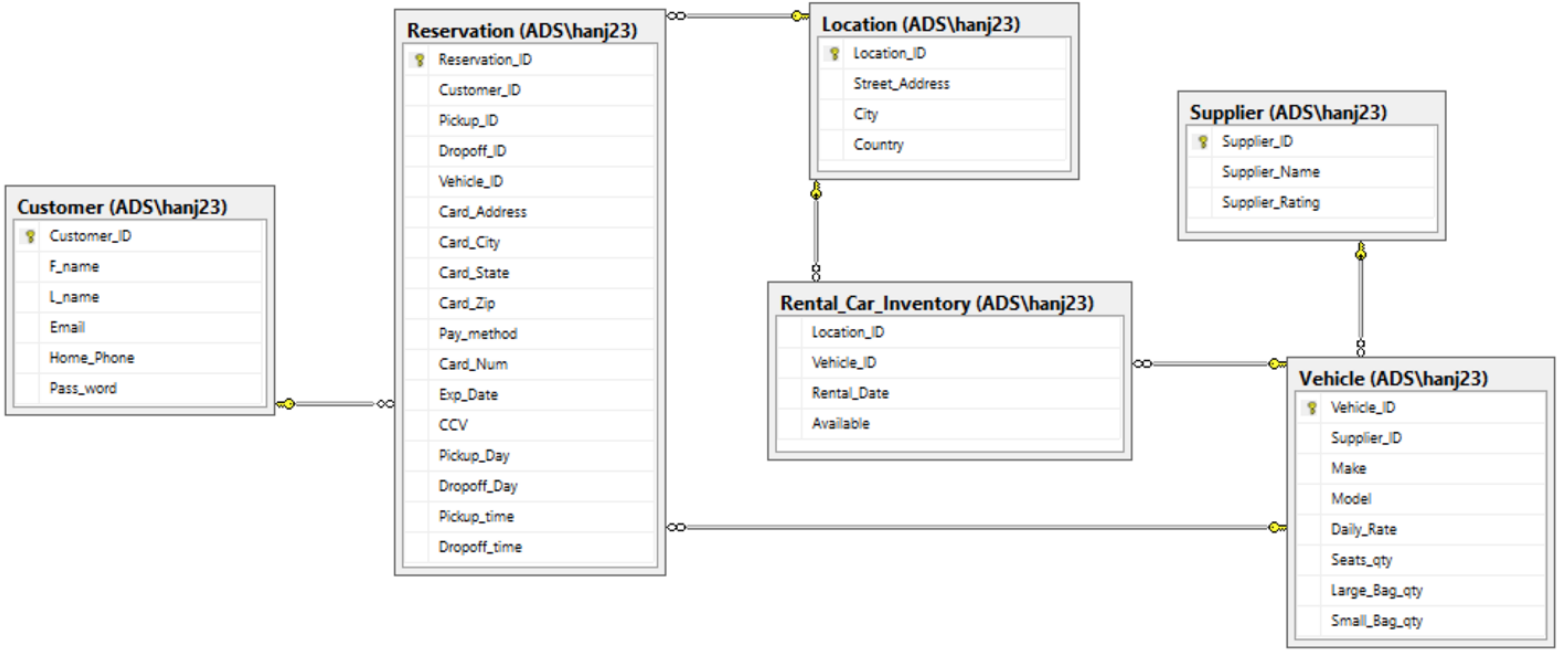

Card_Num
Card_Zip
Card_State
Pay_method
Customer_ID
Reservation_ID
Pickup_ID
Exp_Date
Customer_ID
F_name
Pass_word
Customer
Rent
Reservation
Pick up
Drop off
Street_Address
City
Location
Country
Location_ID
Home_Phone
L_name
Email
Card_Address
Card_City
CCV
Vehicle_ID
Dropoff_Day
Pickup_time
Dropoff_ID
Dropoff_time
Pickup_Day
Location_ID
Rental_Car_Inventory
Vehicle_ID
Available
Rental_Date
Vehicle_ID
Small_Bag_qty
Supplier_ID
Vehicle
Supplied by
Supplier_ID
Supplier_Name
Supplier
Supplier_Rating
Seats_qty
Model
Daily_Rate
Large_Bag_qty
Make

Table Design (SQL)

```
Create table Customer(     -- can't use 'user' as table name, built in function
        Customer_ID char(5) Not Null,
        Payment_ID char(5) Not null,
        F_name varchar(30),
        L_name varchar(30),
        Email varchar(30),
        Home_Phone varchar(15),
        Pass_word varchar(20), -- password is also built in function
        Primary key (Customer_ID),
);

Create Table Supplier (
        Supplier_ID char(5) not null,
        Supplier_Name varchar(20),
        Supplier_Rating decimal(3,2),   -- ratings as decimals
        Primary Key (Supplier_ID)
);

Create table Vehicle (
        Vehicle_ID char(5) not null,
        Supplier_ID char(5) not null,
        Make varchar(20),
        Model varchar(20),
        Daily_Rate varchar(20),
        Seats_qty varchar(2),
        Large_Bag_qty varchar(2),
        Small_Bag_qty varchar(2),
        size varchar(20),
        Primary Key (Vehicle_ID),
        Foreign Key (Supplier_ID) references Supplier
                on delete cascade
                on update cascade
);
```

```
Create table Location (
        Location_ID char(5) Not null,
        Address varchar(20),
        City varchar(20),
        Country varchar(20),
        Primary Key (Location_ID)
);

Create table Rental_Car_Inventory (
        Location_ID char(5) Not null,
        Vehicle_ID char(5) Not null,
        Rental_Date date,
        Available bit
);


Create table Reservation (
        Reservation_ID char(5) Not Null,
        Customer_ID char(5) Not Null,
        Pickup_ID char(5) Not null,
        Dropoff_ID char(5) Not null,
        Vehicle_ID char(5) Not null,

        Pickup_time time,
        Dropoff_time time,
        First_Rental_Day date,
        Last_Rental_Day date,

        CC_Address varchar(30),  --CC = credit card
        CC_City varchar(30),
        CC_State varchar(30),
        CC_Zip char(6),       -- since we know zip code always 6 characters
        Pay_method varchar(30),
        Card_Num varchar(30),
        Exp_Date date,
        CCV char(3),    --ccv always 3 digits

        Primary key (Reservation_ID),
        Foreign key (Customer_ID) references Customer,
        Foreign key (Pickup_ID) references location,
        Foreign key (Dropoff_ID) references location,
        Foreign Key (Vehicle_ID) references Vehicle
);
```

Relational Database



Integrity control

```
Create table Vehicle (
    Vehicle_ID char(5) not null,
    Supplier_ID char(5),
    Make varchar(20),
    Model varchar(20),
    Daily_Rate varchar(20),
    Seats_qty varchar(2),
    Large_Bag_qty varchar(2),
    Small_Bag_qty varchar(2),
    Primary Key (Vehicle_ID),
    Foreign Key (Supplier_ID) references Supplier
        on delete cascade
        on update cascade
);
```

```
Create table Rental_Car_Inventory (
    Location_ID char(5),
    Vehicle_ID char(5),
    Rental_Date date,
    Available bit,
    Foreign Key (Location_ID) references Location
        On delete set null
        On update cascade,
    Foreign key (vehicle_ID) references Vehicle
        On delete set null
        On update cascade
);
```

Integrity Control Continued

```
    Primary key (Reservation_ID),
    Foreign key (Customer_ID) references Customer
        On delete No action
        On update No action,
    Foreign key (Pickup_ID) references location
        On delete No action
        On update No action,
    Foreign key (Dropoff_ID) references location
        On delete No action
        On update No action,
    Foreign Key (Vehicle_ID) references Vehicle
        On delete No action
        On update No action
);
```