CS631-103 Term Project

Phase 2

Group 26: Jairo A. Perez
Manavkumar Patel

Date: 11/18/2022

Table of Contents

PART 1 – Business Requirements:

The Following are the business requirements for the Car Rental system. These requirements will be grouped into sections to provide a better understanding from each component.

Customer requirements:

A. A customer must make a reservation prior to rent a car.
B. If customer decides to proceed with the rental, she/he will get a rental agreement.
C. The customer must provide name, address, and phone at reservation time.
D. Lastly, the customer must provide a valid driver's License and credit card if she/he gets a rental agreement.

Reservation requirements:

A. The reservation must specify a type of class car.
B. The reservation will be made for a specific pick-up and drop-off location.
C. Pick-up and drop-off location do not have to be the same but can.
D. The reservation will include a rental period. This period consists of the start date and time and end date and time.
E. Reservation will have a type. To know if customer reserved by phone, walk-in or online. *
F. The reservation will have a status. *
G. Status will be active if customer shows up at pick-up location on start date.
H. Reservation will be cancelled if customer does not appear on start date and/or calls to cancel.

Location requirements:

A. A location can hold several cars.
B. Not all locations will have the same number of cars or classes of cars.
C. Each location is associated with a reservation either in pick-up or drop-off form.

Car requirements:

A. A car must be either at a location or with a customer (under a rental agreement).
B. If a car is rented it must be associated with a rental agreement.
C. Once a car is returned it becomes available for a new rental agreement.
D. A car is associated with a specific model

Model Requirements:

A. A model has several cars associated to it.
B. A model is associated with a single car class.

Car Class Requirements:

A. A class car has several models associated to it.
B. A class car has several cars associated to it.
C. Each reservation has a car class specified.

Agreement Requirements:

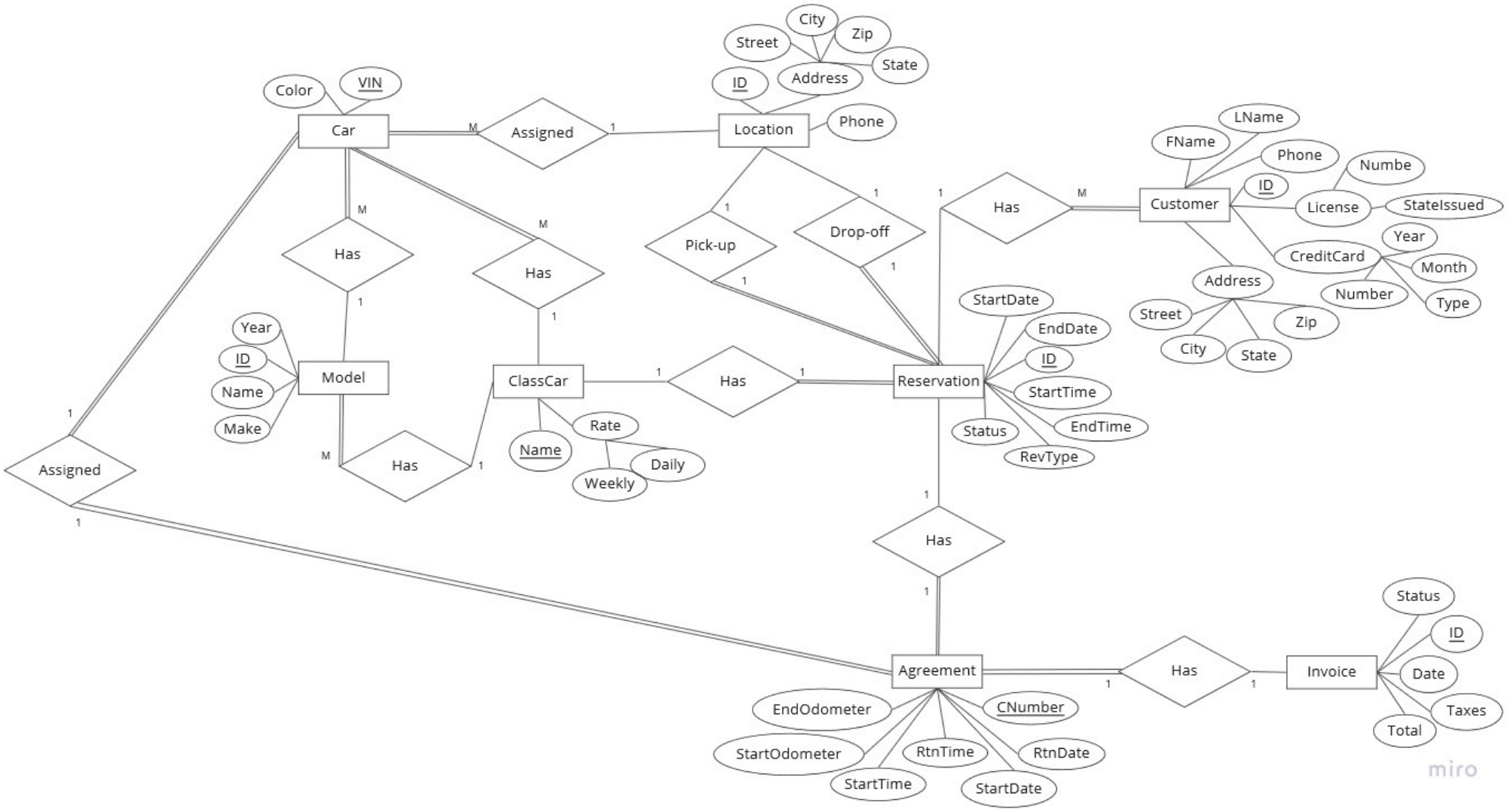
A. Getting a rental agreement is only possible if there is a reservation.
B. Agreements can only be made to one person.
C. Agreements can only be associated with one specific car.
D. An agreement's rental period is based on reservation rental period.
E. If a car is returned earlier, the Agreement will still base its rental period of the reservation.
F. If a car is returned late (after reservation rental period) the Agreement will save its new return date and time.
G. The agreement specifies the car's initial odometer reading.
H. Once the car is return the agreement will store the ending or returning odometer reading.

Invoice Requirements:

A. An invoice must be associated with a rental agreement.
B. Invoices will have a status. This indicate if it has been paid or not.
C. The invoice will calculate its amount due based on:
    a. Comparing Reservation's rental period and Agreement's return date.
        i. If return dates are equal no extra fees are applied
        ii. If Agreement's return date is less than Reservation's, amount is based on Reservation's period.
        iii. If Agreement's return date is older than Reservation's, amount is based on Reservation's period, extra days, and late fees.
    b. Extra days:
        i. Extra days are solely charge on daily rate
    c. Rates:
        i. Car rates are based on the car class selected
        ii. If a reservation period is a week period, the weekly rate is applied.
        iii. If a reservation period is less than a week period, the daily rate is applied.
        iv. If a reservation period is more than a week period:
            1. A 7-day period will have a weekly rate
            2. Days outside of a week period will be charge with a daily rate.
    d. Late Fee:
        i. A late fee is 15% of a daily rate for a type of class car.
        ii. A late fee will be applied to every single extra day.

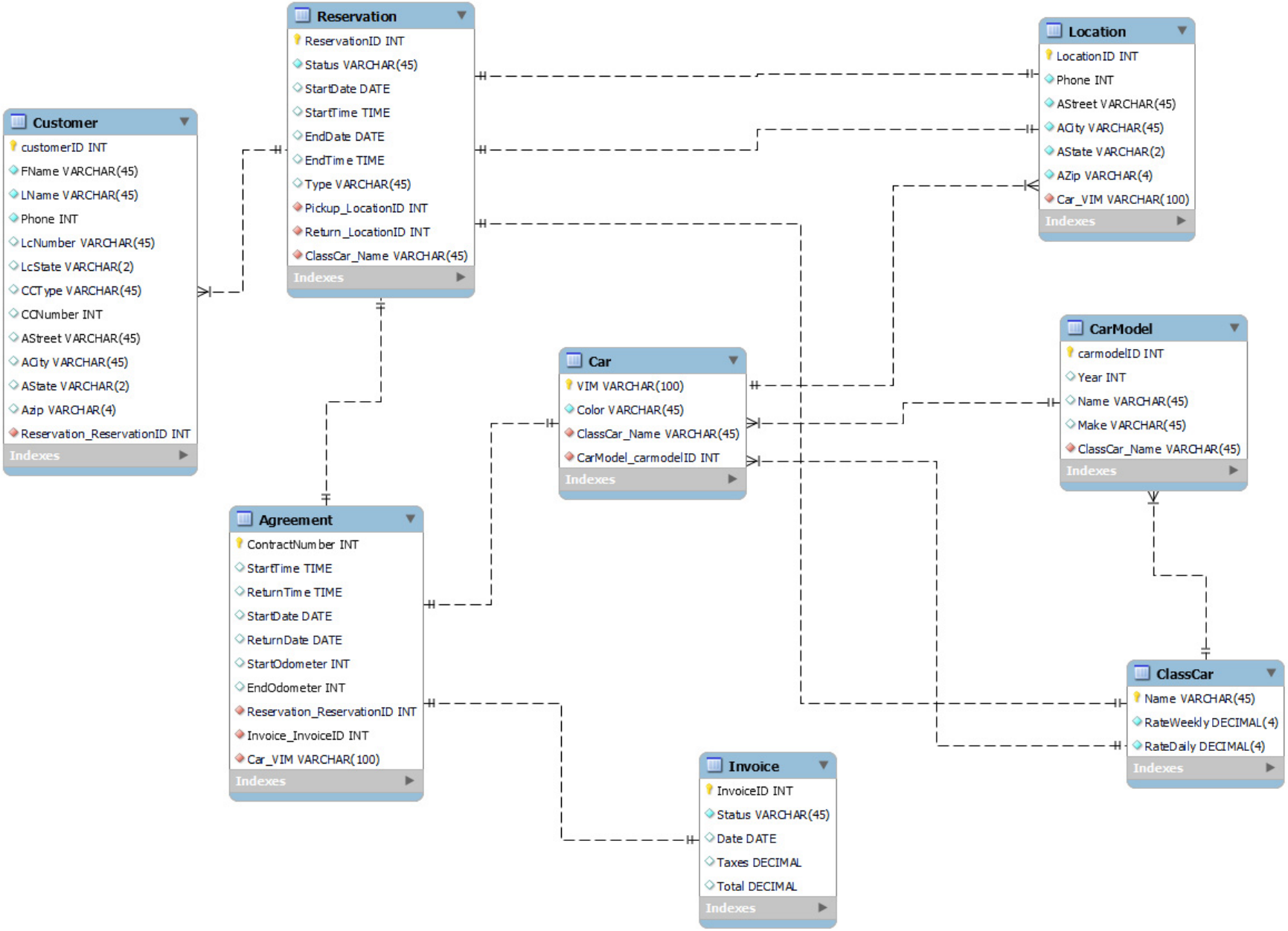
PART 2 – ER Diagram:

Below you can find the Entity Relationship diagram based on the business requirements.



PART 3 – Rational Schema Diagram:

Below you can find the Relational Schema based on the ER diagram.



PART 4 – Database and Table Creation

SQL Statements:

Below please find SQL statements used to create the database (NJIT_cs631103_carproject) and the appropriate tables for the database.

These statements were created based on the design from Phase 1.

```
-- -----------------------------------------------------
-- Schema NJIT_cs631103_carproject
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `NJIT_cs631103_carproject` DEFAULT CHARACTER SET utf8;
USE `NJIT_cs631103_carproject`;
-- -----------------------------------------------------
-- Table `NJIT_cs631103_carproject`.`ClassCar`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `NJIT_cs631103_carproject`.`ClassCar` (
  `Name` VARCHAR(45) NOT NULL,
  `RateWeekly` DECIMAL(4) NOT NULL,
  `RateDaily` DECIMAL(4) NOT NULL,
  PRIMARY KEY (`Name`))
UNIQUE INDEX `Name_UNIQUE` (`Name` ASC) VISIBLE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `NJIT_cs631103_carproject`.`CarModel`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `NJIT_cs631103_carproject`.`CarModel` (
  `carmodelID` INT NOT NULL AUTO_INCREMENT,
  `Year` INT NULL DEFAULT NULL,
  `Name` VARCHAR(45) NULL DEFAULT NULL,
  `Make` VARCHAR(45) NULL DEFAULT NULL,
  `ClassCar_Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`carmodelID`),
  UNIQUE INDEX `fk_CarModel_ClassCar1_idx` (`ClassCar_Name` ASC) VISIBLE,
  CONSTRAINT `fk_CarModel_ClassCar1`
    FOREIGN KEY (`ClassCar_Name`)
    REFERENCES `NJIT_cs631103_carproject`.`ClassCar` (`Name`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `NJIT_cs631103_carproject`.`Car`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `NJIT_cs631103_carproject`.`Car` (
  `VIM` VARCHAR(100) NOT NULL,
  `Color` VARCHAR(45) NOT NULL,
  `ClassCar_Name` VARCHAR(45) NOT NULL,
  `CarModel_carmodelID` INT NOT NULL,
  PRIMARY KEY (`VIM`),
  UNIQUE INDEX `VIM_UNIQUE` (`VIM` ASC) VISIBLE,
  INDEX `fk_Car_ClassCar1_idx` (`ClassCar_Name` ASC) VISIBLE,
  INDEX `fk_Car_CarModel1_idx` (`CarModel_carmodelID` ASC) VISIBLE,
  CONSTRAINT `fk_Car_ClassCar1`
    FOREIGN KEY (`ClassCar_Name`)
    REFERENCES `NJIT_cs631103_carproject`.`ClassCar` (`Name`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Car_CarModel1`
    FOREIGN KEY (`CarModel_carmodelID`)
    REFERENCES `NJIT_cs631103_carproject`.`CarModel` (`carmodelID`)
```

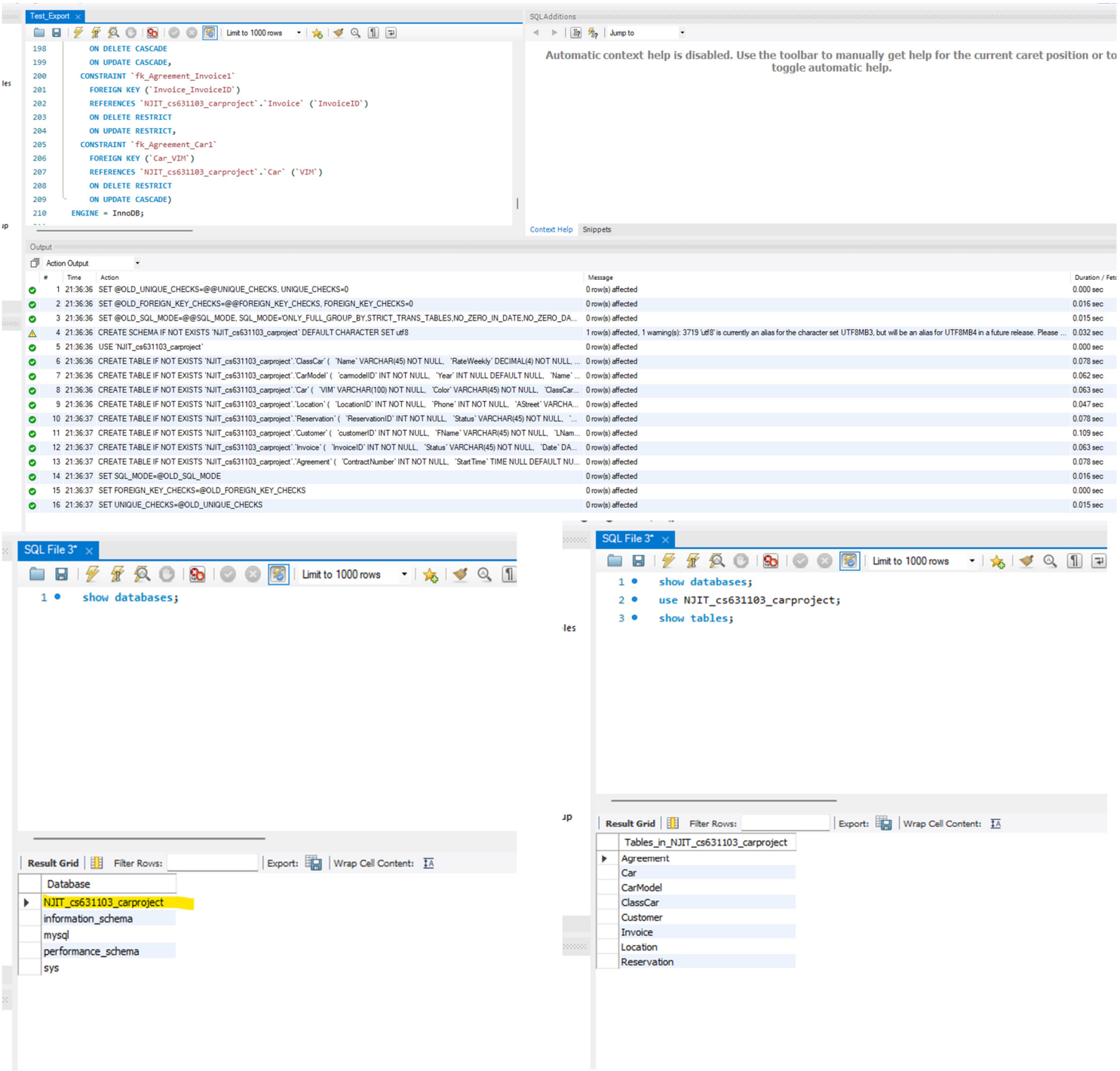
```
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `NJIT_cs631103_carproject`.`Location`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `NJIT_cs631103_carproject`.`Location` (
  `LocationID` INT NOT NULL AUTO_INCREMENT,
  `Phone` INT NOT NULL,
  `AStreet` VARCHAR(45) NOT NULL,
  `ACity` VARCHAR(45) NOT NULL,
  `AState` VARCHAR(2) NOT NULL,
  `AZip` VARCHAR(4) NOT NULL,
  `Car_VIM` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`LocationID`),
  UNIQUE INDEX `fk_Location_Car1_idx` (`Car_VIM` ASC) VISIBLE,
  CONSTRAINT `fk_Location_Car1`
    FOREIGN KEY (`Car_VIM`)
    REFERENCES `NJIT_cs631103_carproject`.`Car` (`VIM`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `NJIT_cs631103_carproject`.`Reservation`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `NJIT_cs631103_carproject`.`Reservation` (
  `ReservationID` INT NOT NULL AUTO_INCREMENT,
  `Status` VARCHAR(45) NOT NULL,
  `StartDate` DATE NULL DEFAULT NULL,
  `StartTime` TIME NULL DEFAULT NULL,
  `EndDate` DATE NULL DEFAULT NULL,
  `EndTime` TIME NULL DEFAULT NULL,
  `Type` VARCHAR(45) NULL DEFAULT NULL,
  `Pickup_LocationID` INT NOT NULL,
  `Return_LocationID` INT NOT NULL,
  `ClassCar_Name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ReservationID`),
  UNIQUE INDEX `ReservationID_UNIQUE` (`ReservationID` ASC) VISIBLE,
  INDEX `fk_Reservation_Location1_idx` (`Pickup_LocationID` ASC) VISIBLE,
  INDEX `fk_Reservation_Location2_idx` (`Return_LocationID` ASC) VISIBLE,
  INDEX `fk_Reservation_ClassCar1_idx` (`ClassCar_Name` ASC) VISIBLE,
  CONSTRAINT `fk_Reservation_Location1`
    FOREIGN KEY (`Pickup_LocationID`)
    REFERENCES `NJIT_cs631103_carproject`.`Location` (`LocationID`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Reservation_Location2`
    FOREIGN KEY (`Return_LocationID`)
    REFERENCES `NJIT_cs631103_carproject`.`Location` (`LocationID`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Reservation_ClassCar1`
    FOREIGN KEY (`ClassCar_Name`)
    REFERENCES `NJIT_cs631103_carproject`.`ClassCar` (`Name`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `NJIT_cs631103_carproject`.`Customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `NJIT_cs631103_carproject`.`Customer` (
```

```
  `customerID` INT NOT NULL AUTO_INCREMENT,
  `FName` VARCHAR(45) NOT NULL,
  `LName` VARCHAR(45) NOT NULL,
  `Phone` VARCHAR(45) NOT NULL,
  `LcNumber` VARCHAR(45) NULL DEFAULT NULL,
  `LcState` VARCHAR(2) NULL DEFAULT NULL,
  `CCType` VARCHAR(45) NULL DEFAULT NULL,
  `CCNumber` INT NULL DEFAULT NULL,
  `AStreet` VARCHAR(45) NULL DEFAULT NULL,
  `ACity` VARCHAR(45) NULL DEFAULT NULL,
  `AState` VARCHAR(2) NULL DEFAULT NULL,
  `Azip` VARCHAR(4) NULL DEFAULT NULL,
  `Reservation_ReservationID` INT NULL,
  PRIMARY KEY (`customerID`),
  UNIQUE INDEX `customerID_UNIQUE` (`customerID` ASC) VISIBLE,
  INDEX `fk_Customer_Reservation_idx` (`Reservation_ReservationID` ASC) VISIBLE,
  UNIQUE INDEX `LcNumber_UNIQUE` (`LcNumber` ASC) VISIBLE,
  CONSTRAINT `fk_Customer_Reservation`
    FOREIGN KEY (`Reservation_ReservationID`)
    REFERENCES `NJIT_cs631103_carproject`.`Reservation` (`ReservationID`)
    ON DELETE SET NULL
    ON UPDATE RESTRICT)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `NJIT_cs631103_carproject`.`Invoice`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `NJIT_cs631103_carproject`.`Invoice` (
  `InvoiceID` INT NOT NULL AUTO_INCREMENT,
  `Status` VARCHAR(45) NOT NULL,
  `Date` DATE NULL DEFAULT NULL,
  `Taxes` DECIMAL NULL DEFAULT NULL,
  `Total` DECIMAL NULL DEFAULT NULL,
  PRIMARY KEY (`InvoiceID`),
  UNIQUE INDEX `InvoiceID_UNIQUE` (`InvoiceID` ASC) VISIBLE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `NJIT_cs631103_carproject`.`Agreement`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `NJIT_cs631103_carproject`.`Agreement` (
  `ContractNumber` INT NOT NULL AUTO_INCREMENT,
  `StartTime` TIME NULL DEFAULT NULL,
  `ReturnTime` TIME NULL DEFAULT NULL,
  `StartDate` DATE NULL DEFAULT NULL,
  `ReturnDate` DATE NULL DEFAULT NULL,
  `StartOdometer` INT NULL DEFAULT NULL,
  `EndOdometer` INT NULL DEFAULT NULL,
  `Reservation_ReservationID` INT NOT NULL,
  `Invoice_InvoiceID` INT NULL,
  `Car_VIM` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`ContractNumber`),
  UNIQUE INDEX `ContractNumber_UNIQUE` (`ContractNumber` ASC) VISIBLE,
  INDEX `fk_Agreement_Reservation1_idx` (`Reservation_ReservationID` ASC) VISIBLE,
  INDEX `fk_Agreement_Invoice1_idx` (`Invoice_InvoiceID` ASC) VISIBLE,
  INDEX `fk_Agreement_Car1_idx` (`Car_VIM` ASC) VISIBLE,
  CONSTRAINT `fk_Agreement_Reservation1`
    FOREIGN KEY (`Reservation_ReservationID`)
    REFERENCES `NJIT_cs631103_carproject`.`Reservation` (`ReservationID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Agreement_Invoice1`
    FOREIGN KEY (`Invoice_InvoiceID`)
```

```
  REFERENCES `NJIT_cs631103_carproject`.`Invoice` (`InvoiceID`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT,
 CONSTRAINT `fk_Agreement_Car1`
  FOREIGN KEY (`Car_VIM`)
  REFERENCES `NJIT_cs631103_carproject`.`Car` (`VIM`)
  ON DELETE RESTRICT
  ON UPDATE CASCADE)
ENGINE = InnoDB;
```

SQL Execution Proof:

The images below shows the successful execution of the SQL statements.



PART 5 – Data Insertion

Requirements:

Insert 5 rows of data in each table

SQL Statements:

```
show databases;
use NJIT_cs631103_carproject;
show tables;

INSERT INTO ClassCar (Name, RateWeekly, RateDaily)
VALUES
("Van",100.00,20.00),
("Minivan",90.00,20.00),
("Pickup",700.00,100.00),
("Sedan",80.00,15.00),
("Sports",5000.00,1000.00);

INSERT INTO CarModel (Make,Name,Year,ClassCar_Name)
VALUES
("Toyota","Sienna",2023,"Minivan"),
("Dodge","Sprinter",2009,"Van"),
("Tesla","Model 3",2022,"Sedan"),
("GMC","Hummer ",2023,"Pickup"),
("Cadillac","CTS-V",2016,"Sports");

INSERT INTO Car(VIM, Color, ClassCar_Name, CarModel_carmodelID)
VALUES
("xyz0987abc1776","red","Sports", 5),
("abc1234xyz1492","purple","Pickup", 4),
("njitm02d09y1881","black","Sedan", 3),
("newark1666nj2022","yellow","Van", 2),
("idont000konw111vim","Green","Minivan", 1);

INSERT INTO Location(AStreet, ACity, AState, AZip, Phone, Car_VIM)
VALUE
("719 E 11th Ave", "Anchorage", "AK","99501", "123-456-7890", "idont000konw111vim"),
("8000 Gtwy Blvd E", "El Paso", "TX", "79907", "987-654-3210","abc1234xyz1492"),
("180 Washington St", "Newark", "NJ", "07102", "159-357-8520","newark1666nj2022"),
("500 Truman Ave", "Key West", "FL", "33040", "1-800-456-4562","njitm02d09y1881"),
("650 Airport Dr", "Presque Isle", "ME", "04769", "1-800-7416", "xyz0987abc1776");

INSERT INTO Reservation(Status, StartDate, StartTime, EndDate, EndTime, Type, Pickup_LocationID, Return_LocationID,
ClassCar_Name)
VALUES
("open", "2022-04-03", "06:20:15", "2022-04-13", "22:22:22", "walkin", 4, 2, "Minivan"),
("open", "2022-05-03", "06:20:15", "2022-05-13", "22:22:22", "phone", 3, 3, "Pickup"),
("open", "2022-06-03", "06:20:15", "2022-06-13", "22:22:22", "walkin", 2, 4,"Sedan"),
("open", "2022-06-03", "06:20:15", "2022-07-13", "22:22:22", "phone", 1, 5, "Sports");

INSERT INTO Customer(FName, LName, Phone, Reservation_ReservationID )
VALUES
("Jairo","Perez","1234567890", 11),
("Sebastian","Correa","9876543210", 12),
("Ana","Palacio","1597534862", 13),
("Camila","Salderriaga","9638527418", 14),
("Richy","Perez","1472583692", 15);

INSERT INTO Invoice (Status, Date,Taxes,Total)
VALUE
```

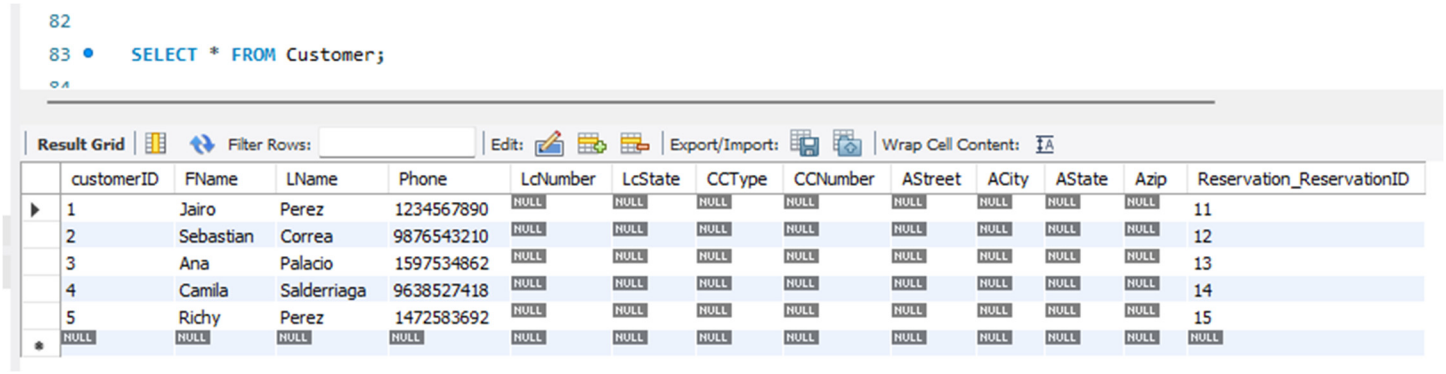
```
("open", "2022-03-03", 0.25, 0.00),
("open", "2022-04-03", 0.26, 0.00),
("open", "2022-05-03", 0.27, 0.00),
("open", "2022-06-03", 0.28, 0.00),
("open", "2022-06-03", 0.29, 0.00);

INSERT INTO Agreement (StartDate, StartTime, ReturnDate, ReturnTime, StartOdometer, Reservation_ReservationID, Car_VIM)
VALUE
("2022-03-03", "06:20:15", "2022-03-13", "22:22:22", 1001, 15,"xyz0987abc1776" ),
("2022-04-03", "06:20:15", "2022-04-13", "22:22:22", 2001, 14,"abc1234xyz1492"),
("2022-05-03", "06:20:15", "2022-05-13", "22:22:22", 3001, 13,"njitm02d09y1881"),
("2022-06-03", "06:20:15", "2022-06-13", "22:22:22", 4001, 12,"newark1666nj2022"),
("2022-06-03", "06:20:15", "2022-07-13", "22:22:22", 5001, 11,"idont000konw111vim");
```

SQL Execution Proof:

Write SQL statements with the "**SELECT ***" to proof data is stored in each corresponding table.

```
SELECT * FROM Customer;
```

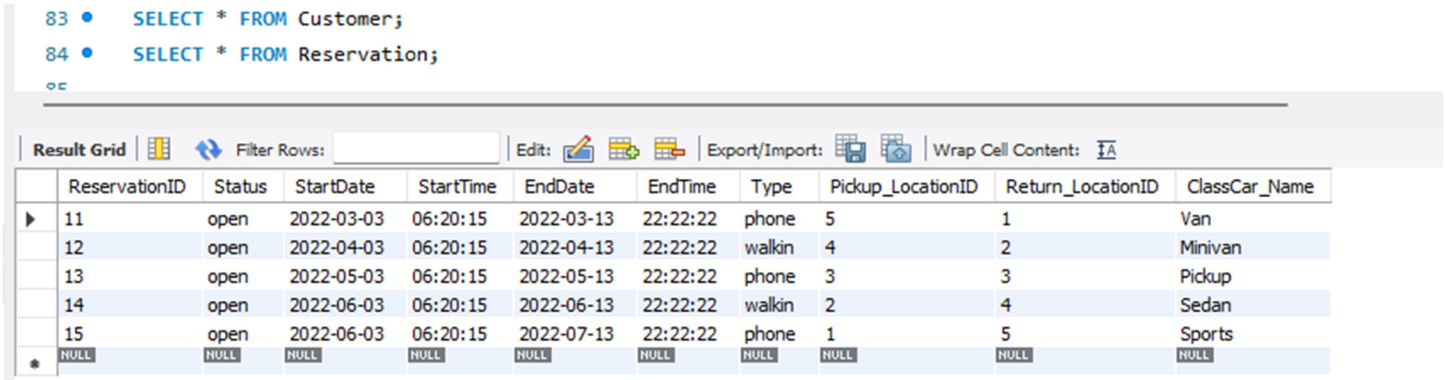


```
82
83 •  SELECT * FROM Customer;
84
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customerID	FName	LName	Phone	LcNumber	LcState	CCType	CCNumber	AStreet	ACity	AState	Azip	Reservation_ReservationID
1	Jairo	Perez	1234567890	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	11
2	Sebastian	Correa	9876543210	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	12
3	Ana	Palacio	1597534862	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	13
4	Camila	Salderriaga	9638527418	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	14
5	Richy	Perez	1472583692	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT * FROM Reservation;
```



```
83 •  SELECT * FROM Customer;
84 •  SELECT * FROM Reservation;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ReservationID	Status	StartDate	StartTime	EndDate	EndTime	Type	Pickup_LocationID	Return_LocationID	ClassCar_Name
11	open	2022-03-03	06:20:15	2022-03-13	22:22:22	phone	5	1	Van
12	open	2022-04-03	06:20:15	2022-04-13	22:22:22	walkin	4	2	Minivan
13	open	2022-05-03	06:20:15	2022-05-13	22:22:22	phone	3	3	Pickup
14	open	2022-06-03	06:20:15	2022-06-13	22:22:22	walkin	2	4	Sedan
15	open	2022-06-03	06:20:15	2022-07-13	22:22:22	phone	1	5	Sports
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM Location;

```
84 •  SELECT * FROM Reservation;
85 •  SELECT * FROM Location;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Cor

LocationID	Phone	AStreet	ACity	AState	AZip	Car_VIM
1	123-456-7890	719 E 11th Ave	Anchorage	AK	9950	idont000konw111vim
2	987-654-3210	8000 Gtwy Blvd E	El Paso	TX	7990	abc1234xyz1492
3	159-357-8520	180 Washington St	Newark	NJ	0710	newark1666nj2022
4	1-800-456-4562	500 Truman Ave	Key West	FL	3304	njitm02d09y1881
5	1-800-7416	650 Airport Dr	Presque Isle	ME	0476	xyz0987abc1776
NULL	NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM Car;

```
85 •  SELECT * FROM Location;
86 •  SELECT * FROM Car;
```

Result Grid | Filter Rows: | Edit: | Expo

VIM	Color	ClassCar_Name	CarModel_carmodelID
abc1234xyz1492	purple	Pickup	4
idont000konw111vim	Green	Minivan	1
newark1666nj2022	yellow	Van	2
njitm02d09y1881	black	Sedan	3
xyz0987abc1776	red	Sports	5
NULL	NULL	NULL	NULL

SELECT * FROM CarModel;

```
86 •  SELECT * FROM Car;
87 •  SELECT * FROM CarModel;
```

Result Grid | Filter Rows: | Edit:

carmodelID	Year	Name	Make	ClassCar_Name
1	2023	Sienna	Toyota	Minivan
2	2009	Sprinter	Dodge	Van
3	2022	Model 3	Tesla	Sedan
4	2023	Hummer	GMC	Pickup
5	2016	CTS-V	Cadillac	Sports
NULL	NULL	NULL	NULL	NULL

SELECT * FROM ClassCar;

```
87 •    SELECT * FROM CarModel;
88 •    SELECT * FROM ClassCar;
```

Result Grid | Filter Rows: | Edit:

Name	RateWeekly	RateDaily
Minivan	90	20
Pickup	700	100
Sedan	80	15
Sports	5000	1000
Van	100	20
NULL	NULL	NULL

SELECT * FROM Agreement;

```
88 •    SELECT * FROM ClassCar;
89 •    SELECT * FROM Agreement;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ContractNumber	StartTime	ReturnTime	StartDate	ReturnDate	StartOdometer	EndOdometer	Reservation_ReservationID	Invoice_InvoiceID	Car_VIM
7	06:20:15	22:22:22	2022-03-03	2022-03-13	1001	NULL	15	NULL	xyz0987abc1776
8	06:20:15	22:22:22	2022-04-03	2022-04-13	2001	NULL	14	NULL	abc1234xyz1492
9	06:20:15	22:22:22	2022-05-03	2022-05-13	3001	NULL	13	NULL	njitm02d09y1881
10	06:20:15	22:22:22	2022-06-03	2022-06-13	4001	NULL	12	NULL	newark1666nj2022
11	06:20:15	22:22:22	2022-06-03	2022-07-13	5001	NULL	11	NULL	idont000konw111vim
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM Invoice;

```
89 •    SELECT * FROM Agreement;
90 •    SELECT * FROM Invoice;
```

Result Grid | Filter Rows: | Edit: | Export/I

InvoiceID	Status	Date	Taxes	Total
1	open	2022-03-03	0.25	0.00
2	open	2022-04-03	0.26	0.00
3	open	2022-05-03	0.27	0.00
4	open	2022-06-03	0.28	0.00
5	open	2022-06-03	0.29	0.00
NULL	NULL	NULL	NULL	NULL

PART 6 – Data Update & Deletion

Requirements:

1) Update 1 column in each table.
2) Delete 1 row in each table.

SQL Statements:

```
UPDATE Customer SET FName=" Ana Maria" WHERE customerID=3;
UPDATE Reservation SET EndDate="2022-05-23" WHERE ReservationID=13;
UPDATE Location SET AStreet="38 Access Hwy", ACity="Caribou", AZip=0473 WHERE LocationID=5;
UPDATE Car SET Color="pink" WHERE VIM="xyz0987abc1776";
UPDATE CarModel SET Year=2020 WHERE carmodelID=3;
UPDATE ClassCar SET RateWeekly=5010, RateDaily=1010 WHERE Name="Sports";
UPDATE Agreement SET ReturnDate="2022-05-23" WHERE ContractNumber=9;
UPDATE Invoice SET Taxes=0.35 WHERE InvoiceID=3;

DELETE FROM Customer WHERE customerID=5;
DELETE FROM Reservation WHERE ReservationID=11;
DELETE FROM Location WHERE LocationID=5;
DELETE FROM Car WHERE VIM="xyz0987abc1776";
DELETE FROM CarModel WHERE CarmodelID=5;
DELETE FROM ClassCar WHERE Name="Sports";
DELETE FROM Invoice WHERE InvoiceID=5;
```

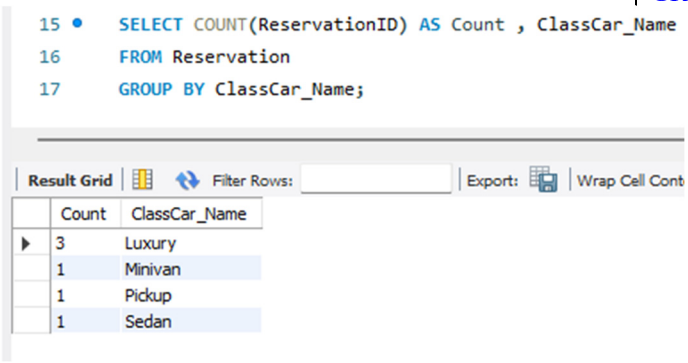
PART 7 – Data Update & Deletion

SQL Queries:

a. Group By

Get a count of the classes of cars being rented out.

```
SELECT COUNT(ReservationID) AS Count , ClassCar_Name
FROM Reservation
GROUP BY ClassCar_Name;
```

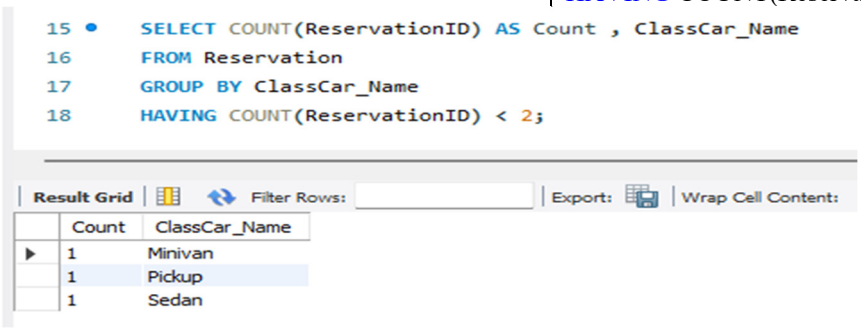


Count	ClassCar_Name
3	Luxury
1	Minivan
1	Pickup
1	Sedan

b. Group By & Having

Get all reservations that have only 1 class of car reserved.

```
SELECT COUNT(ReservationID) AS Count , ClassCar_Name
FROM Reservation
GROUP BY ClassCar_Name
HAVING COUNT(ReservationID) < 2;
```

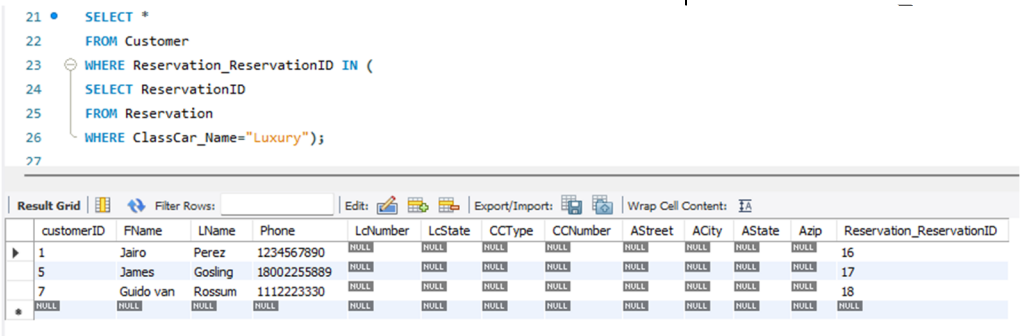


Count	ClassCar_Name
1	Minivan
1	Pickup
1	Sedan

c. In

Get all the customer who have a current reservation with a luxury car.

```
SELECT *
FROM Customer
WHERE Reservation_ReservationID IN (
SELECT ReservationID
FROM Reservation
WHERE ClassCar_Name="Luxury");
```

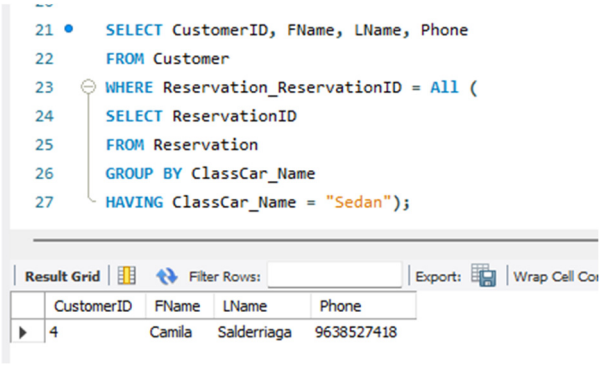


customerID	FName	LName	Phone	LcNumber	LcState	CCType	CCNumber	AStreet	ACity	AState	Azip	Reservation_ReservationID
1	Jairo	Perez	1234567890	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	16
5	James	Gosling	18002255889	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	17
7	Guido van	Rossum	1112223330	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	18
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

d. All

Get all the customer who have a current reservation with a luxury car.

```
SELECT CustomerID, FName, LName, Phone
FROM Customer
WHERE Reservation_ReservationID = All (
SELECT ReservationID
FROM Reservation
GROUP BY ClassCar_Name
HAVING ClassCar_Name = "Sedan");
```



PART 8 – Version Control

Version	Amendment Number	Date	Requestor	Comment	Results
1	1.1	11/2/2022	Prof. Canan	MODEL is a separate entity, not just an attribute. CAR has MODEL, MODEL is associated with CLASS. Please revise your diagram.	ER diagram has been modified to reflect changes. Entity MODEL has been added. Changes reflected on: Part 2, page 5.
	1.2	11/16/2022	Jairo A. Perez	Current "Relational Schema" diagram is difficult to read. Arrows are difficult to follow.	Switched from Microsoft Word to MySQL Workbench for relational schema diagram creation.