K-Town Car Share

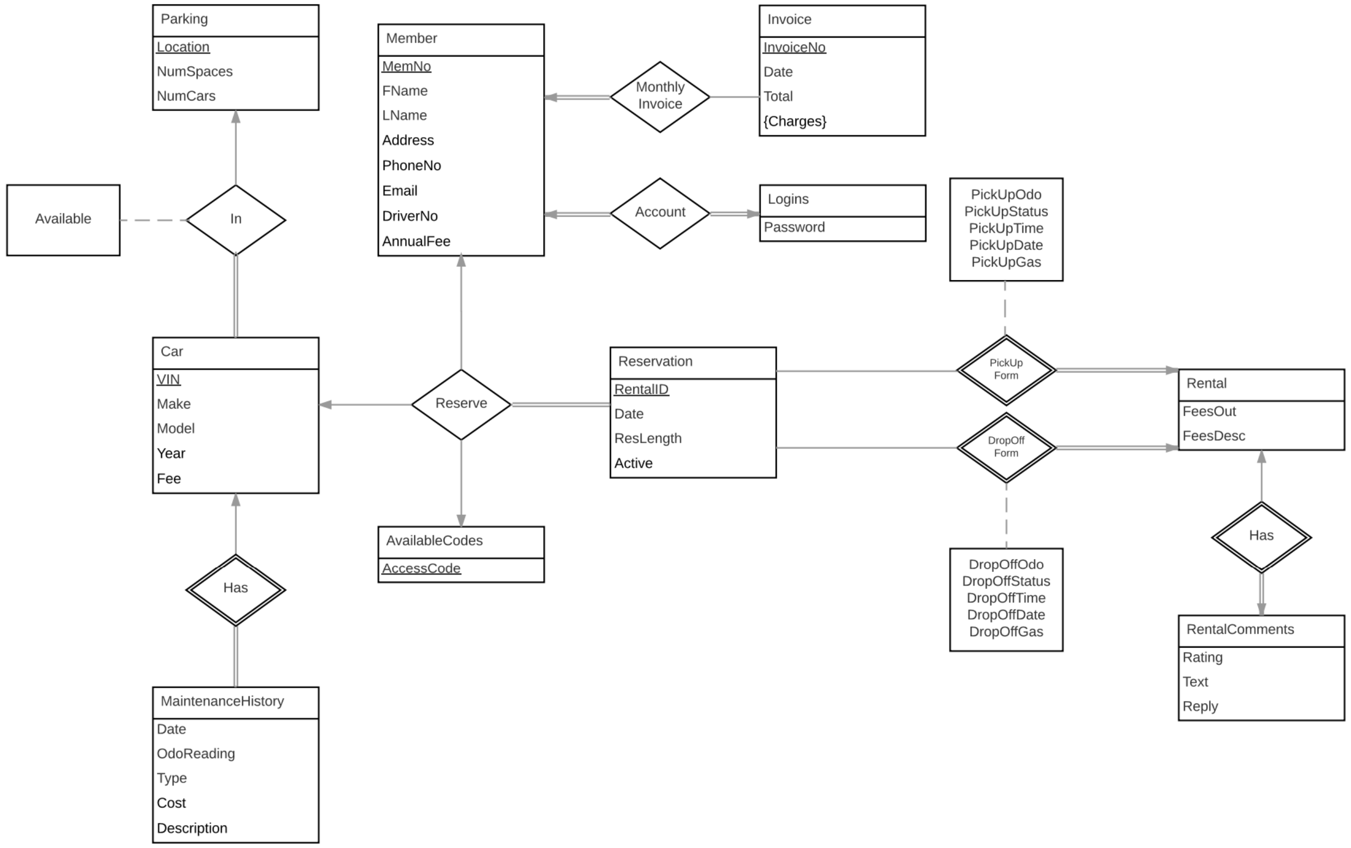
Technical Report

Kyle Lavorato

April 6th, 2017

Table of Contents

Database ER Diagram



Relational Schema

parking(Location, NumSpaces, NumCars)
member(MemNo, FName, LName, Address, PhoneNo, Email, DriverNo, MonthlyFee)
car(VIN, Make, Model, Year, Location, Fee, Available)
maintenanceHistory(VIN, Date, Cost, OdoReading, Type, Description)
reservation(RentalID, MemNo, VIN, Date, ResLength, Active)
availablecodes(AccessCode, RentalID)
invoice(InvoiceNo, Date, MemNo, Total)
invoicecharges(InvoiceNo, Description, Cost)
rental(RentalID, MemNo, VIN, PickUpStatus, PickUpOdo, PickUpGas, PickUpDate, PickUpTime, DropOffStatus, DropOffOdo, DropOffGas, DropOffDate, DropOffTime, FeesOut, FeesDesc)
rentalComments(MemNo, VIN, RentalID, Rating, Text, Reply)
logins(MemNo, Password)

Additional Design Notes

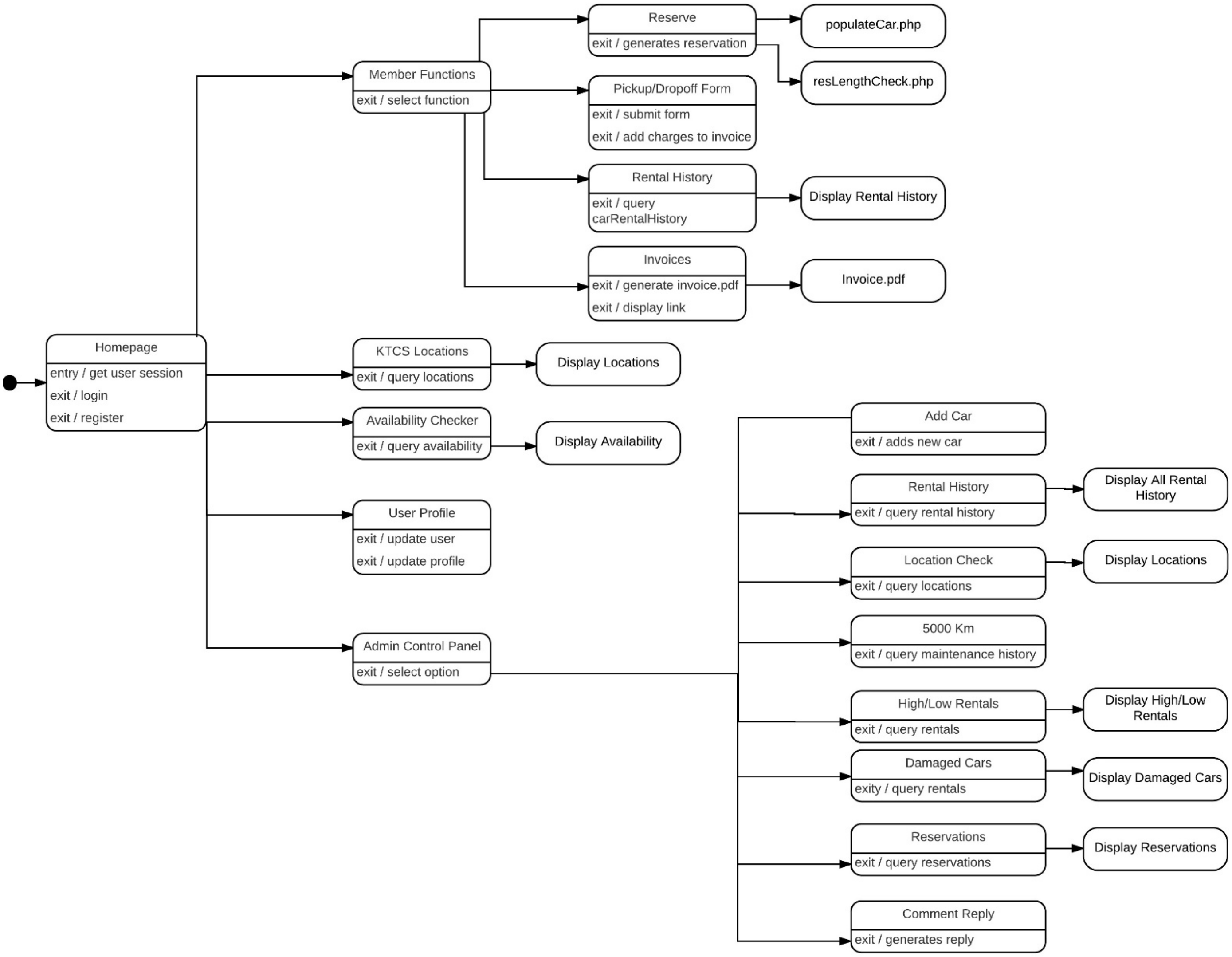
Reservation

The RentalID is created in the Reservation table for specific reasons. When the customer places a reservation on a car, they are entering a contract to rent that car. Since they are entering that contract, the rental will happen eventually, so the reservation is then assigned a unique identification number, so it can be tracked. Thus, since the unique ID is created in this table, the Rental table will have no primary key. It will use the RentalID from the reservation as a pseudo-primary key through a foreign key relationship. For all intents and purposes, it is a primary key for both relations. This also saves extra overhead of generating a second unique ID, one for each table. The reservations need a unique ID to be tracked, so there is no point of creating that second ID when a rental is parented to follow a reservation. Finally, this structure is required for the access code situation described below.

AvailableCodes

The available codes table has direct need for the RentalID in its table. This is necessary for multiple reasons. First, the table must exist to be able to ensure that only unique codes are generated as old codes must be somehow avoided. The current structure avoids this by generating a large number of unique codes and then removing them when used. Which codes are available for different cars is then kept track of through the RentalID being added. When a key is registered to a car, the RentalID is added to it. The access code cannot be stored on the car table, as many codes are generated in advance of pickup and the user should not be able to use the code to open the car before their scheduled time. With the current system, the active code for each car will be pushed to it daily by querying a natural join of availablecodes and reservation, looking for the current reservations. The codes need to be available then, as the user must be able to enter the car before they can fill out the pickup form, which is the only other time it could be generated.

State Machine Diagram



SQL Interactions

5000miles.php

Query

```
SELECT VIN, Distance, Make, Model
FROM (SELECT VIN, MAX(Date) as MaintDate, MAX(DropOffDate) as CurrentDate,
DropOffOdo-OdoReading as Distance FROM maintenancehistory NATURAL JOIN rental
GROUP BY VIN) as SUBQUERY NATURAL JOIN car
WHERE Distance > 5000
```

Output

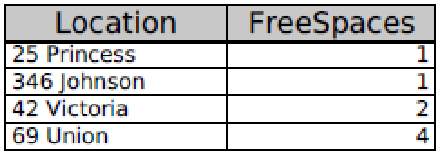
VIN	DISTANCE	Make	Model
1IYFW81OPEX701544	7400	Toyota	Corolla

Addcar.php

Query 1 – Find locations with available space

```
SELECT Location, NumSpaces-NumCars AS FreeSpaces
FROM parking
WHERE NumSpaces > NumCars
```

Output



Location	FreeSpaces
25 Princess	1
346 Johnson	1
42 Victoria	2
69 Union	4

Query 2 – Insert new car values

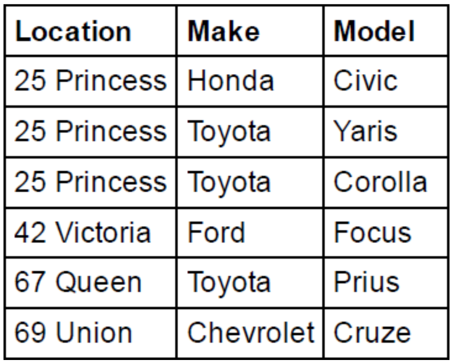
```
INSERT into car VALUES (?, ?, ?, ?, ?, ?, 'Y')
```

Alllocations.php

Query – Find all locations and their currently available cars

```
SELECT Location, Make, Model
FROM parking NATURAL JOIN car
WHERE Available='Y'
ORDER BY Location
```

Output



Location	Make	Model
25 Princess	Honda	Civic
25 Princess	Toyota	Yaris
25 Princess	Toyota	Corolla
42 Victoria	Ford	Focus
67 Queen	Toyota	Prius
69 Union	Chevrolet	Cruze

Availablecheck.php

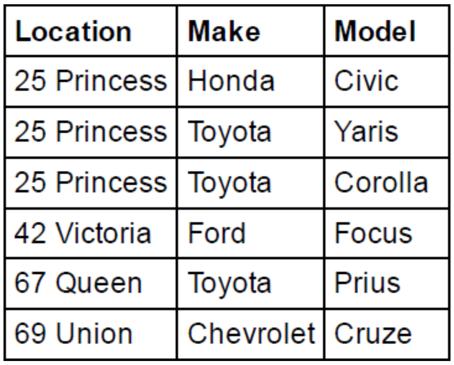
Query – Check if a car is available on a certain date

```
SELECT Location, Make, Model
FROM car NATURAL JOIN parking
WHERE VIN NOT IN (SELECT VIN FROM reservation WHERE ? >= Date AND ? <=
Date(Date + ResLength))
ORDER BY Location
```

Binding

The '?' value is the selected date in SQL date format (eg '2017-04-06)

Output



Location	Make	Model
25 Princess	Honda	Civic
25 Princess	Toyota	Yaris
25 Princess	Toyota	Corolla
42 Victoria	Ford	Focus
67 Queen	Toyota	Prius
69 Union	Chevrolet	Cruze

Commentreply.php

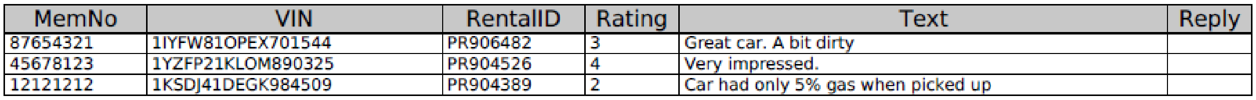
Query 1 – Update a comment

```
UPDATE rentalcomments SET Reply=? WHERE RentalID=?
```

Query 2 – Find all comments without a reply

```
SELECT *
FROM rentalcomments
WHERE Reply IS NULL AND Rating IS NOT NULL
```

Output



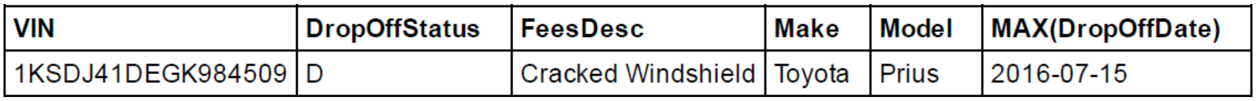
MemNo	VIN	RentalID	Rating	Text	Reply
87654321	1IYFW81OPEX701544	PR906482	3	Great car. A bit dirty	
45678123	1YZFP21KLOM890325	PR904526	4	Very impressed.	
12121212	1KSDJ41DEGK984509	PR904389	2	Car had only 5% gas when picked up	

Damagedcars.php

Query

```
SELECT VIN, DropOffStatus, FeesDesc, Make, Model, MAX(DropOffDate)
FROM rental NATURAL JOIN car
WHERE DropOffStatus='D' OR DropOffStatus='NR'
GROUP BY VIN
```

Output



VIN	DropOffStatus	FeesDesc	Make	Model	MAX(DropOffDate)
1KSDJ41DEGK984509	D	Cracked Windshield	Toyota	Prius	2016-07-15

Form.php

Car Pickup Form

Query 1 – Create rental entry; Where ? binding is form information

```
INSERT into rental VALUES (?, ?, ?, ?, ?, ?, ?, ?, NULL, NULL, NULL, NULL,
NULL, NULL, NULL)
```

Query 2 – Create comments entry for rental; Where ? binding is rental info

```
INSERT into rentalcomments VALUES (?, ?, ?, NULL, NULL, NULL)
```

Query 3 – Update the car to not be available; Where ? binding is the VIN

```
UPDATE car
SET Available='N'
WHERE VIN=?
```

Car Dropoff Form

Query 1 – Find the initial data; Where '?' binding is the rentalID

```
SELECT PickUpStatus, PickUpGas
FROM rental
WHERE RentalID=?
```

Query 2 – Update rental information; Where '?' binding is the form info

```
UPDATE rental
SET DropOffStatus=?, DropOffOdo=?, DropOffGas=?, DropOffDate=?,
DropOffTime=?, FeesOut=?
WHERE RentalID=?
```

Query 3 – Update the car to be available; Where '?' binding is the VIN

```
UPDATE reservation
SET Active='N'
WHERE RentalID=?
```

Query 4 – Find the fee information for the rental and member monthly fee

```
SELECT Fee, ResLength, MonthlyFee
FROM car NATURAL JOIN reservation NATURAL JOIN member
WHERE MemNo=? AND VIN=?"
```

Query 5 – Check if a current invoice exists for the user

```
SELECT InvoiceNo
FROM invoice
WHERE MemNo=? AND Date=?
```

Query 5 (if 'No') – Create a new invoice for the user and add monthly fees

```
INSERT into invoice VALUES (?, ?, ?, NULL)
INSERT into invoicecharges VALUES (?, ?, ?)
```

Query 6 – Add rental fees to invoice; Where '?' binding is the rental fees

```
INSERT into invoicecharges VALUES (?, ?, ?)
```

Query 7 – Add damage fees if they exist

```
INSERT into invoicecharges VALUES (?, ?, ?)
```

Main Page

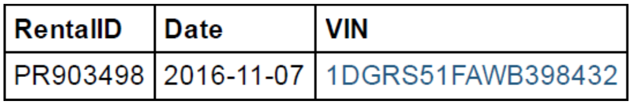
Query 1 – Find all current rentals for the user; Where '?' binding is the memberID

```
SELECT RentalID, Date, VIN FROM reservation WHERE Active='Y' AND MemNo=?
```

Binding

'23232323'

Output



RentalID	Date	VIN
PR903498	2016-11-07	1DGRS51FAWB398432

Index.php

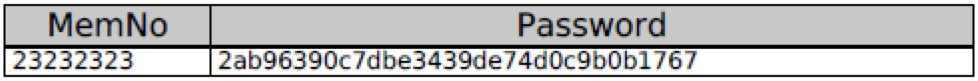
Query – Check if username and password hash exist in the database

```
SELECT MemNo, Password FROM logins WHERE MemNo=? AND Password=?
```

Binding

The '?' binding is the MemNo and Password, eg '23232323' and '2ab96390c7dbe3439de74d0c9b0b1767'

Output



MemNo	Password
23232323	2ab96390c7dbe3439de74d0c9b0b1767

Invoice.php

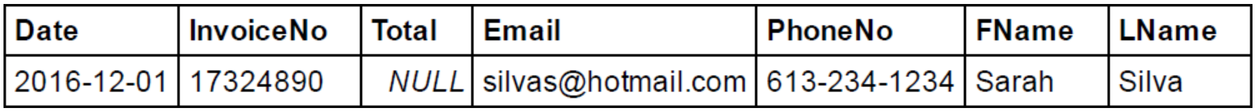
Query 1 – Find invoice and member information

```
SELECT Date, InvoiceNo, Total, Email, PhoneNo, FName, LName
FROM invoice NATURAL JOIN member
WHERE MemNo=? AND Date=?
```

Binding

‘23232323’ and ‘2016-12-01’

Output



Date	InvoiceNo	Total	Email	PhoneNo	FName	LName
2016-12-01	17324890	*NULL*	silvas@hotmail.com	613-234-1234	Sarah	Silva

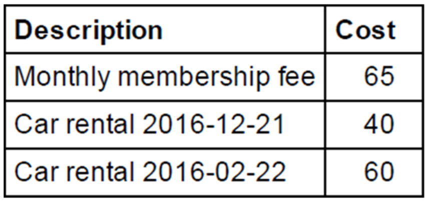
Query 2 – Find all charges for the selected invoice

```
SELECT Description, Cost
FROM invoicecharges
WHERE InvoiceNo=?
```

Binding

‘17324890’

Output



Description	Cost
Monthly membership fee	65
Car rental 2016-12-21	40
Car rental 2016-02-22	60

Query 3 – Set the total of the invoice once its ben calculated

```
UPDATE invoice
SET Total=?
WHERE MemNo=?
```

Locationcheck.php

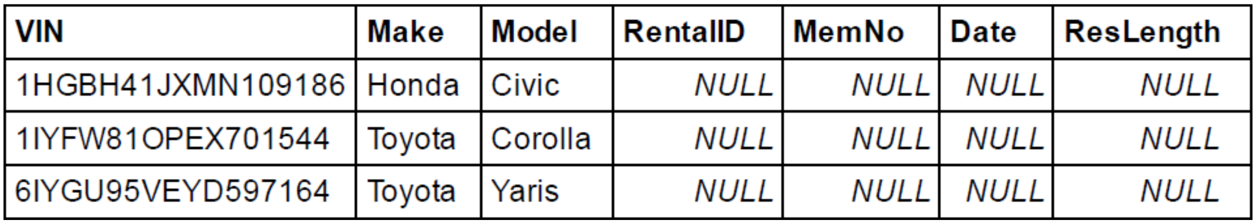
Query – Find all cars at each location and all their active reservations

```
SELECT VIN, Make, Model, RentalID, MemNo, Date, ResLength
FROM car LEFT OUTER JOIN (SELECT VIN, RentalID, MemNo, Date, ResLength FROM
reservation WHERE Date > Date(?)) AS subquery USING (VIN)
WHERE Location=?
ORDER BY Make
```

Binding

‘2016-06-05’ and ’25 Princess’

Output



VIN	Make	Model	RentalID	MemNo	Date	ResLength
1HGBH41JXMN109186	Honda	Civic	*NULL*	*NULL*	*NULL*	*NULL*
1IYFW81OPEX701544	Toyota	Corolla	*NULL*	*NULL*	*NULL*	*NULL*
6IYGU95VEYD597164	Toyota	Yaris	*NULL*	*NULL*	*NULL*	*NULL*

Memrentalhist.php

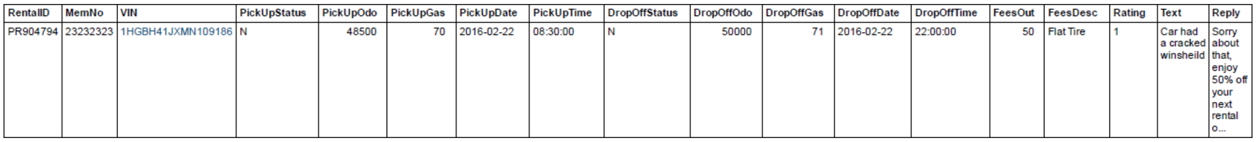
Query 1 – Find all history and comment info for the user

```
SELECT *
FROM rental NATURAL JOIN rentalcomments
WHERE MemNo=?
```

Binding

‘23232323’

Output



RentalID	MemNo	VIN	PickUpStatus	PickUpOdo	PickUpGas	PickUpDate	PickUpTime	DropOffStatus	DropOffOdo	DropOffGas	DropOffDate	DropOffTime	FeesOut	FeesDesc	Rating	Text	Reply
PR904794	23232323	1HGBH41JXMN109186	N	48500	70	2016-02-22	08:30:00	N	50000	71	2016-02-22	22:00:00	50	Flat Tire	1	Car had a cracked winsheild	Sorry about that, enjoy 50% off your next rental o...

Query 2.1 – Update a rating with a comment entered

```
UPDATE rentalcomments
SET Rating=?, Text=?
WHERE RentalID=?
```

Query 2.2 – Update a rating without a comment entered

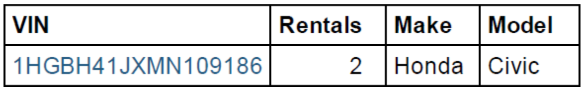
```
UPDATE rentalcomments
SET Rating=?
WHERE RentalID=?
```

Mostrentals.php

Query 1.1 – Find the most rented car

```
SELECT VIN, Rentals, Make, Model
FROM (SELECT VIN, COUNT(VIN) as Rentals FROM rental GROUP BY VIN) as SUBQUERY
NATURAL JOIN car
ORDER BY Rentals
DESC LIMIT 1
```

Output

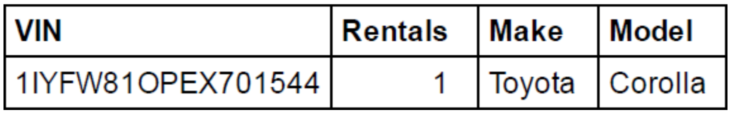


VIN	Rentals	Make	Model
1HGBH41JXMN109186	2	Honda	Civic

Query 1.2 – Find the least rented car

```
SELECT VIN, Rentals, Make, Model
FROM (SELECT VIN, COUNT(VIN) as Rentals FROM rental GROUP BY VIN) as SUBQUERY
NATURAL JOIN car
ORDER BY Rentals
ASC LIMIT 1
```

Output



VIN	Rentals	Make	Model
1IYFW81OPEX701544	1	Toyota	Corolla

Populatecar.php

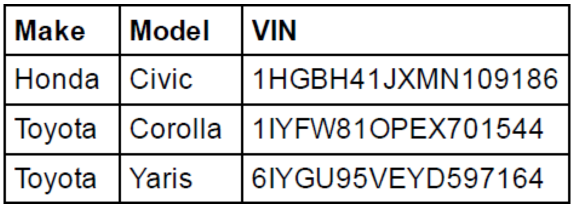
Query – Find all available cars for the location; Triggered through Javascript

```
SELECT Make, Model, VIN
FROM car
WHERE Location='".mysql_real_escape_string($_GET["parent"])."'
```

Binding

’25 Princess’

Output



Make	Model	VIN
Honda	Civic	1HGBH41JXMN109186
Toyota	Corolla	1IYFW81OPEX701544
Toyota	Yaris	6IYGU95VEYD597164

Profile.php

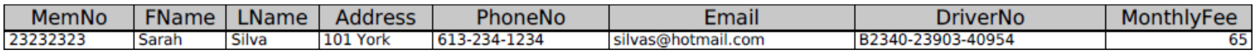
Query 1 – Find all the current member information to display

```
SELECT MemNo, FName, LName, Address, PhoneNo, Email, DriverNo, MonthlyFee
FROM member
WHERE MemNo=?
```

Binding

‘23232323’

Output



MemNo	FName	LName	Address	PhoneNo	Email	DriverNo	MonthlyFee
23232323	Sarah	Silva	101 York	613-234-1234	silvas@hotmail.com	B2340-23903-40954	65

Query 2 – Update member information; Where ‘?’ binding is the user form info

```
UPDATE member
SET Address=?, PhoneNo=?, Email=?
WHERE MemNo=?
```

Query 3 – Update user password; Where ‘?’ binding is the new and old password info

```
UPDATE logins
SET Password=?
WHERE MemNo=? AND Password=?
```

Register.php

Query 1 – Add the new user to the user table; Where ? binding is the register form info

```
INSERT into member VALUES (?, ?, ?, ?, ?, ?, ?, ?)
```

Query 2 – Add the username and password to the logins; Where ? binding is the memNo and hashed password

```
INSERT into logins VALUES (?, ?)
```

Rentalhistory.php

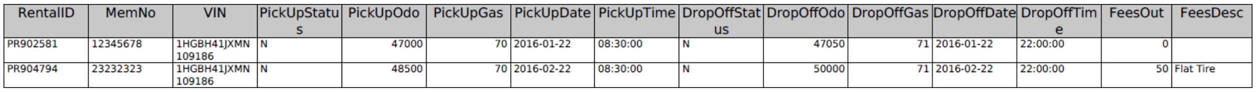
Query – Find all rental history for a car

```
SELECT *
FROM rental
WHERE VIN=?
```

Binding

‘1HGBH41JXMN109186’

Output



RentalID	MemNo	VIN	PickUpStatus	PickUpOdo	PickUpGas	PickUpDate	PickUpTime	DropOffStatus	DropOffOdo	DropOffGas	DropOffDate	DropOffTime	FeesOut	FeesDesc
PR902581	12345678	1HGBH41JXMN109186	N	47000	70	2016-01-22	08:30:00	N	47050	71	2016-01-22	22:00:00	0	
PR904794	23232323	1HGBH41JXMN109186	N	48500	70	2016-02-22	08:30:00	N	50000	71	2016-02-22	22:00:00	50	Flat Tire

Reservations.php

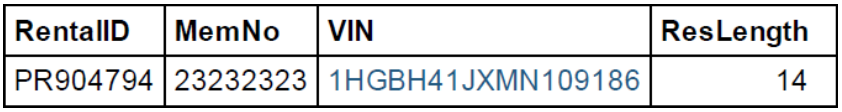
Query – Find all reservations for a specific date

```
SELECT RentalID, MemNo, VIN, ResLength
FROM reservation
WHERE Date=?
```

Binding

‘2016-06-06’

Output



RentalID	MemNo	VIN	ResLength
PR904794	23232323	1HGBH41JXMN109186	14

Reservecar.php

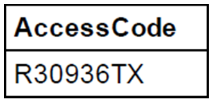
Query 1 – Add the active reservation for the car; Where ‘?’ binding is the form info

```
INSERT into reservation VALUES (?, ?, ?, ?, ?, 'Y')
```

Query 2 – Randomly select a single access code for this rental

```
SELECT AccessCode
FROM availablecodes
WHERE RentalID IS NULL
ORDER BY RAND()
LIMIT 1
```

Output



AccessCode
R30936TX

Query 3 – Add the rentalID to the selected access code for this rental

```
UPDATE availablecodes
SET RentalID=?
WHERE AccessCode=?
```

Reslengthcheck.php

Query – Check if the car is reserved during the requested time frame; Triggered through Javascript

```
SELECT VIN
FROM reservation
WHERE (Date <= ADDDATE(?, INTERVAL ? DAY) AND ADDDATE(Date, INTERVAL
ResLength DAY) >= ?) AND VIN=?
```

Binding

‘2017-04-06’ (Current Date) and ‘7’ (Reservation length) and ‘2017-04-06’ (Current Date) and ‘1HGBH41JXMN109186’ (VIN)

Output

Blank as the car is free at this time.