**Project team #6 - Car Dealership**

**Team Members:**

**Sharat Vyas**

**Roshanak Ashrafi**

**Divya Ayila**

**Sai Nishanth Dilly**

## ***SPRINT 0***

## ***PROJECT PROPOSAL***

**Content, Scope and Objectives**

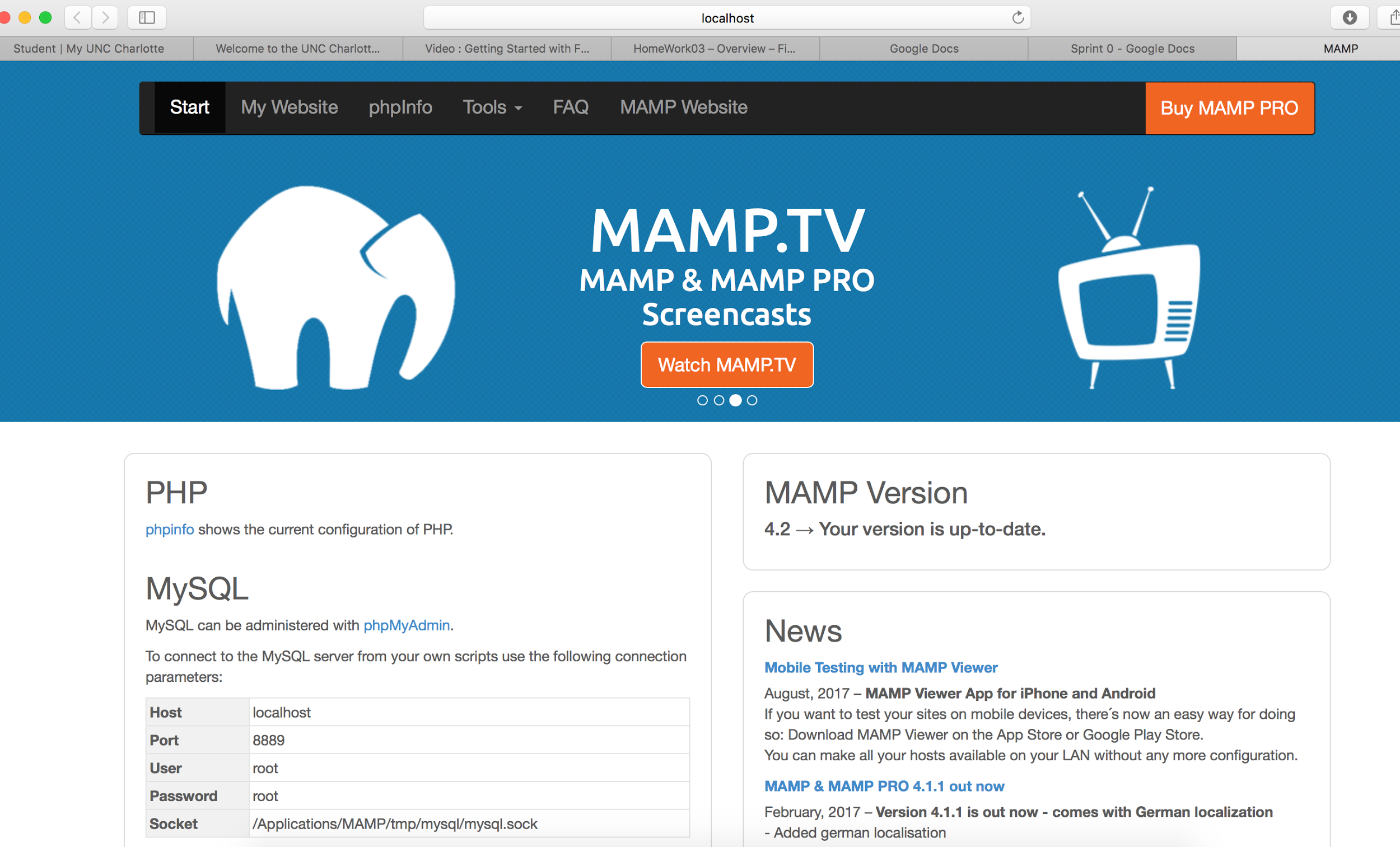
Our group would like to create a car dealership management system. Normally car dealerships have 2 components, the service center and the sales center. We plan to follow a similar approach. Our system will have the 2 basic components of a sales center and a service center. The sales center will be responsible for keeping track of dealership’s new car inventory, used car inventory, purchase history, customer information, etc. The service center will focus primarily on the servicing records, customer information, car information, etc. The system will also keep track of the dealership’s employees and manage their permissions and abilities.

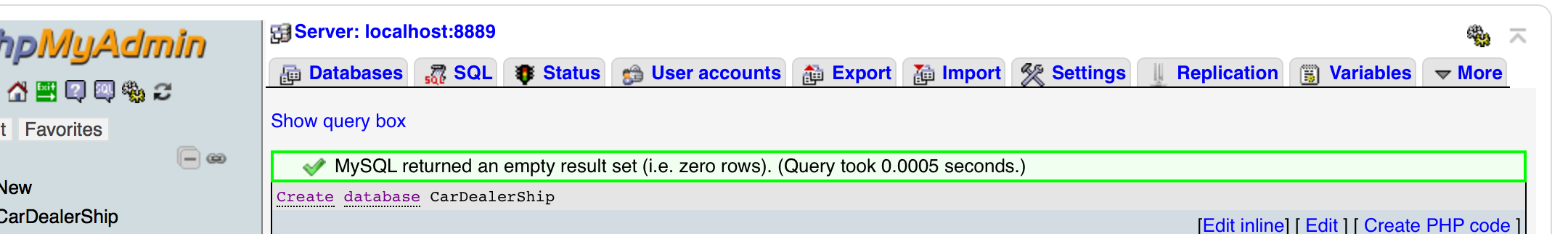
## ***PROJECT ENVIRONMENT***

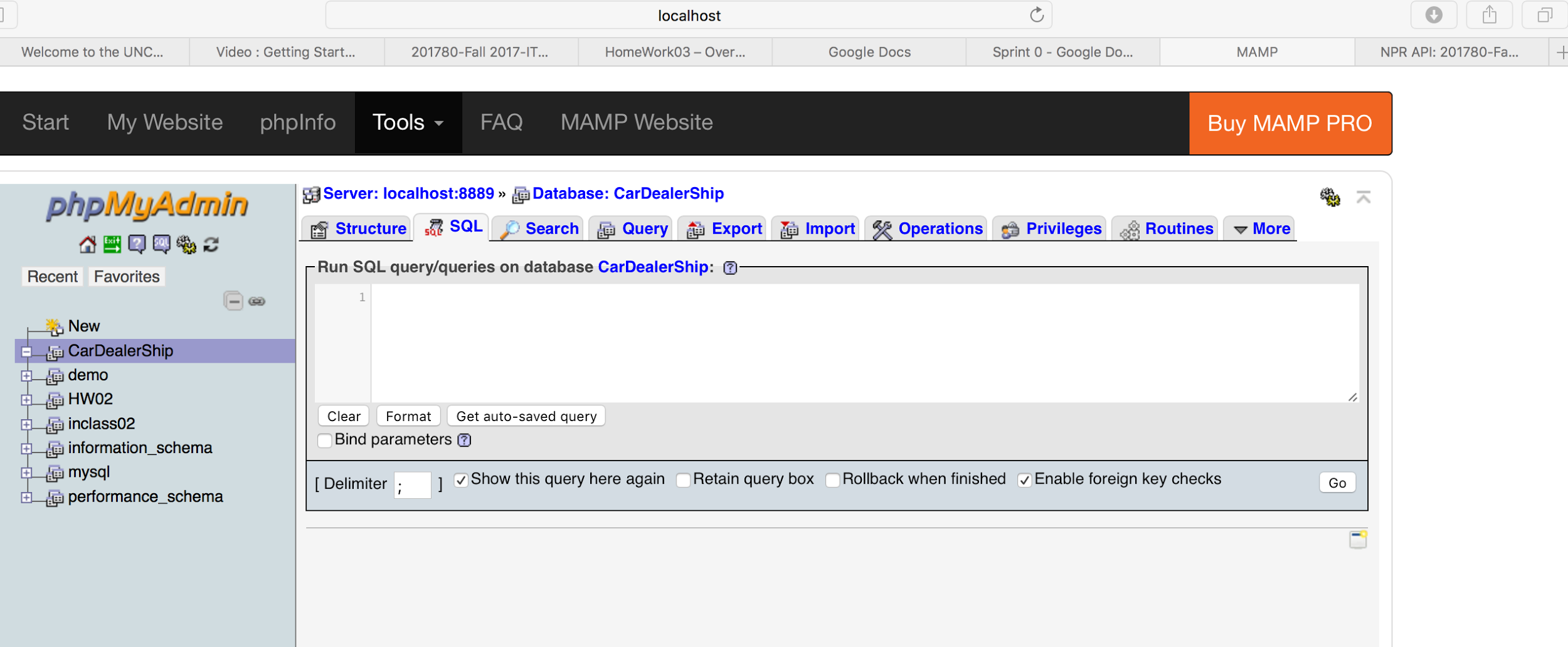
For project environment setup we have used MAMP/WAMP server which contains MYSQL, APACHE and PHPMYADMIN.

MAMP version - 4.2 [Mac Os]

WAMP version - 3.0.6 [Windows]



Create Database Car Dealership;



## ***HIGH LEVEL REQUIREMENTS***

### **Initial user roles**

|  |  |
| --- | --- |
| **User Role** | **Description** |
| Inventory Manager | The inventory manager is responsible for keep track of the current inventory of the dealership. He/She adds new cars to the inventory list when they are acquired and removes them when they are sold. He/She can also view the current inventory and manage their working hours. He/She reports to the Dealership manager. |
| Salesman | The salesman works with the potential customers to sell them a car. He/She can view all the current new/used car inventory the dealership has to offer. He/She can also retrieve information about a specific car and manage their working hours. He/She reports to the Inventory manager. |
| Serviceman | The serviceman works with the potential car to repair them. He/She can view the all the current car service history and update them. He/She can also retrieve information about a specific car and manage their working hours. He/She reports to the Service manager. |
| Service Manager | The service manager is in charge of the service center of the dealership. He/She can look at the history of the cars serviced, see the specific service history of a car, and can also update the service history if a car comes in for servicing. The service manager can also view and update the list of services provided. In addition to this, he/she can also manage their working hours. He/She reports to the Dealership manager. |
| Dealership Manager | Dealership Manager is the Admin of the system. The dealership manager’s job is to make sure that the dealership and it’s logistics are running smoothly. He/She can view the list of employees and customers, retrieve information on a specific employee, and manage his/her own working hours. |
| Registered Customer | Customers can see the service history of their car and can browse the current inventory the dealership has to offer. He/She can also look at information about a specific car and see what services the dealership offers. |
| Guest | Guest is an unregistered user. He/She can see information about the available new/used cars and available services. The guest can register with the dealership and become a Registered Customer if he/she would like to. |

### **Initial user story descriptions**

|  |  |
| --- | --- |
| **Story ID** | **Story Description** |
| US1 | As a Dealership manager, I want to add users(Serviceman, Inventory Manager, Service Manager, Salesman) |
| US2 | As a Guest, I want to register with the dealership and become a Registered Customer. |
| US3 | As a Dealership manager, I want to remove Users(Serviceman, Salesman, Service Manager, Dealership Manager, Salesman,Registered Customer) who are terminated. |
| US4 | As an Inventory Manager, I want to add new cars to the new car inventory |
| US5 | As an Inventory Manager, I want to add used cars to the used car inventory |
| US6 | As an Inventory Manager, I want to remove new/used cars from the inventory |
| US7 | As an Inventory Manager, I want to see all new/used car inventory |
| US8 | As an Inventory Manager, I want to see the information about a specific new/used car |
| US9 | As a Salesman, I want to see all new/used car inventory |
| US10 | As a Salesman, I want to see information about a specific new/used car |
| US11 | As a Customer(guest)/Registered Customer, I want to see brief information about the available new/used cars |
| US12 | As a Service Manager, I want to see history of cars serviced |
| US13 | As a Service Manager, I want to see the service history of a specific car |
| US14 | As a Service Manager, I want to add/update a car service history |
| US15 | As a Serviceman, I want to update a car’s service history |
| US16 | As a Serviceman, I want to view the service history of a car |
| US17 | As a Registered Customer I want to see the history of my car servicing process |
| US18 | As a Dealership manager, I want to see a list of all employees/customers |
| US19 | As a Dealership manager, I want to see information about a specific employee/customer |
| US20 | As a Guest/Registered Customer, I want to browse information about the available services |
| US21 | As a Registered Customer, I want to review my purchase history. |
| US22 | As a Registered Customer, I should be able to buy cars from catalog. |
| US23 | As an Inventory Manager, I want to add my working hours to my daily schedule sheet |
| US24 | As an Inventory Manager, I want to see my working and earning history |
| US25 | As a Salesman, I want to add my working hours to my daily schedule sheet |
| US26 | As a Salesman, I want to see my working and earning history |
| US27 | As a Service Manager, I want to add my working hours to my daily schedule sheet |
| US28 | As a Service Manager, I want to see my working and earning history |
| US29 | As a Serviceman, I want to add my working hours to my daily schedule sheet |
| US30 | As a Serviceman, I want to see my working and earning history |
| US31 | As a Dealership manager, I want to add my working hours to my daily schedule sheet |
| US32 | As a Dealership Manager, I want to see my working and earning history |
| US33 | As a Service Manager, I want to access to Serviceman working history |
| US34 | As a Sales Manager, I want to access Salesman working history. |
| US35 | As a Dealership Manager, I want to access all employee working history. |

*High Level Conceptual Design*

### Entities:

* Employee
* Dealership Manager
* Inventory Manager
* Service Manager
* ServiceMan
* Salesman
* Registered Customer
* Service History
* Purchase History
* Car Inventory
* Offered Services
* Working History

### Relationships:

* Dealership Manager registers Employees(Serviceman, Salesman, Service Manager, Dealership Manager, Salesman)
* Dealership Manager removes Employee(Serviceman, Salesman, Service Manager, Dealership Manager, Salesman)
* Inventory Manager adds the new/used cars to Car Inventory
* Inventory Manager removes the new/used cars from Car Inventory
* Serviceman views Car Inventory
* Salesman views Car Inventory
* Service Manager views Car Inventory
* Dealership Manager views Car Inventory
* Salesman views Car Inventory
* Inventory Manager views Car Inventory
* Registered Customer views Car Inventory
* Registered Customer buys car from Car Inventory
* Registered Customer views his/her own Service History
* Registered Customer view/add his/her own Purchase History
* Service Manager add/update/view Service History
* Salesman update/view Service History
* Salesman update/view Purchase History
* Inventory Manager adds his/her own Working hours to Working History.
* Service Manager adds his/her own Working hours to Working History.
* Salesman adds his/her own Working hours to Working History.
* Serviceman adds his/her own Working hours to Working History.
* Dealership Manager adds his/her own Working hours to Working History.
* Inventory Manager views his/her own Working History.
* Service Manager views his/her own Working History.
* Salesman views Working his/her own Working History.
* Serviceman views his/her own Working History.
* Dealership Manager views his/her own Working History.
* Serviceman reports to Service Manager
* Salesman reports to Inventory Manager
* Inventory Manager reports to Dealership Manager
* Service Manager reports to Dealership Manager

## ***SPRINT 1***

## ***REQUIREMENTS***

### **Refined user story descriptions**

|  |  |
| --- | --- |
| **Story ID** | **Story Description** |
| US1 | As a Dealership manager, I want to add/remove Inventory Manager |
| US2 | As a Dealership manager, I want to add/remove Service Manager |
| US3 | As a Dealership manager, I want to add/remove Serviceman |
| US4 | As a Dealership manager, I want to add/remove Salesman |
| US5 | As a Guest, I want to register with the dealership and become a Registered Customer. |
| US6 | As an Inventory Manager, I want to add new/used cars to the new car inventory |
| US7 | As an Inventory Manager, I want to remove new/used cars from the inventory |
| US8 | As an Inventory Manager, I want to see all new/used car inventory |
| US9 | As an Inventory Manager, I want to see the information about a specific new/used car |
| US10 | As a Salesman, I want to see all new/used car inventory |
| US11 | As a Salesman, I want to see information about a specific new/used car |
| US12 | As a Customer(guest)/Registered Customer, I want to see brief information about the available new/used cars |
| US13 | As a Service Manager, I want to see history of cars serviced |
| US14 | As a Service Manager, I want to see the service history of a specific car |
| US15 | As a Service Manager, I want to add/update a car service history |
| US16 | As a Serviceman, I want to update a car’s service history |
| US17 | As a Serviceman, I want to view the service history of a car |
| US18 | As a Registered Customer I want to see the history of my car servicing process |
| US19 | As a Dealership manager, I want to see a list of all employees/customers |
| US20 | As a Dealership manager, I want to see information about a specific employee/customer |
| US21 | As a Guest/Registered Customer, I want to browse information about the available services |
| US22 | As a Registered Customer, I want to review my purchase history. |
| US23 | As a Registered Customer, I should be able to buy cars from catalog. |
| US24 | As an Inventory Manager, I want to add my working hours to my daily schedule sheet |
| US25 | As an Inventory Manager, I want to see my working and earning history |
| US26 | As a Salesman, I want to add my working hours to my daily schedule sheet |
| US27 | As a Salesman, I want to see my working and earning history |
| US28 | As a Service Manager, I want to add my working hours to my daily schedule sheet |
| US29 | As a Service Manager, I want to see my working and earning history |
| US30 | As a Serviceman, I want to add my working hours to my daily schedule sheet |
| US31 | As a Serviceman, I want to see my working and earning history |
| US32 | As a Dealership manager, I want to add my working hours to my daily schedule sheet |
| US33 | As a Dealership Manager, I want to see my working and earning history |
| US34 | As a Service Manager, I want to access to Serviceman working history |
| US35 | As a Inventory Manager, I want to access Salesman working history. |
| US36 | As a Dealership Manager, I want to access all employee working history. |

## ***SPRINT 1***

## ***REQUIREMENTS***

|  |  |
| --- | --- |
| **Story ID** | **Story Description** |
| US1 | As a Dealership manager, I want to add/remove Inventory Manager so that new employed Inventory manager can add/remove car inventory and terminated Inventory managers can be removed from organization. |
| US2 | As a Dealership manager, I want to add/remove Service Manager so that service manager can service cars, review history of serviced cars and terminated Service Manager can be removed from organization. |
| US6 | As an Inventory Manager, I want to add new/used cars to the new car inventory so that new cars can be viewed by customers. |
| US7 | As an Inventory Manager, I want to remove new/used cars from the inventory so that sold out cars are no longer visible to customers. |
| US8 | As an Inventory Manager, I want to see all new/used car inventory so that inventory manager can manage list of cars added by him. |
| US9 | As an Inventory Manager, I want to see the information about a specific new/used car so that inventory manager can review detailed information about the selected car. |
| US15 | As a Service Manager, I want to add/update a car service history so that we can see if that car is serviced and the type of service done to the car. |
| US13 | As a Service Manager, I want to see history of cars serviced so that service manager can see all the cars serviced under his supervision. |
| US14 | As a Service Manager, I want to see the service history of a specific car so that service manager get history of particular car he wants to  review. |

## Conceptual Design

Entity: **Car\_Detail**

Attributes:

* Vehicle Identification Number
* Year
* Make
* Model
* Mileage
* Car\_Condition

Entity: **Inventory**

Attributes:

* Price
* Description

Entity: **Cars\_Sold**

Attributes:

* Sold\_on
* price\_sold

Entity: **Car\_Features**

Attributes:

* feature\_name

Entity: **Service\_History**

Attributes:

* Invoice\_id
* Service\_date
* Description
* Price

Entity: **Dealership\_Manager**

Attributes:

* Name [composite]
  + First\_name
  + Middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Hours\_per\_week

Entity: **Inventory\_Manager**

Attributes:

* Name [composite]
  + First\_name
  + Middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Commission\_percentage
* Hours\_per\_week

Entity: **Service\_Manager**

Attributes:

* Name [composite]
  + First\_name
  + Middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Hours\_per\_week

Relationships:

* Dealership\_Manager **registers** Inventory\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory\_Manager has total participation
* Dealership\_Manager **removes** Inventory\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory\_Manager has total participation
* Dealership\_Manager **registers** Service\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Service\_Manager has total participation
* Dealership\_Manager **removes** Service\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Service\_Manager has total participation
* Inventory\_Manager **adds** new/used Inventory
  + Cardinality : one to many
  + Participation:
    - Inventory\_Manager has partial participation
    - Inventory has total participation
* Inventory\_Manager **removes** new/used Inventory
  + Cardinality : one to many
  + Participation:
    - Inventory\_Manager has partial participation
    - Inventory has total participation
* Inventory\_Manager **views** Inventory
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory has partial participation
* Service\_Manager **adds** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Service\_Manager **updates** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Service\_Manager **views** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Car\_Sold **has** Service\_History
  + Cardinality : one to many
  + Participation:
    - Car has total participation
    - Service\_Histoy has total participation
* Inventory **IS-A** Car\_Detail
  + Cardinality : one to one
  + Participation:
    - Inventory has total participation
    - Car\_Detail is Partial participation
* Car\_Sold **IS-A** Car\_Detail
  + Cardinality : one to one
  + Participation:
    - Car\_Detail has partial participation
    - Car\_Sold Detail is Total participation

## Logical Design

Table: **Car\_Detail**

Columns:

* Vehicle Identification Number [primary\_key]
* Year
* Make
* Model
* Mileage
* Car\_Condition
* Added\_by[foreign key; references **Email** from **Inventory\_Manager**]

Table: **Inventory**

Columns:

* Vehicle Identification Number [foreign key; references Vehicle Identification Number from **Car\_Detail**] [primary key]
* Description
* Price

Table: **Car\_Sold**

Columns:

* Vehicle Identification Number [foreign key; references Vehicle Identification Number from **Car\_Detail**] [primary key]
* Sold\_on
* Price\_sold

Table: **Car\_Features**

Columns:

* VIN [foreign key; references **VIN** from **Inventory**]
* feature\_name

Table:**Service\_History**

Columns:

* Invoice\_id
* Vehicle\_Identification\_Number [foreign key; references Vehicle Identification Number from **Car\_sold**]
* Service\_date
* Description
* Price
* Serviced\_by[foreign key; references **Email** from **Service\_Manager**]

Table: **Dealership\_Manager**

Columns:

* Name [composite]
  + First\_name
  + middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Hours\_per\_week

Table: **Inventory\_Manager**

Columns:

* Name [composite]
  + First\_name
  + middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Commission\_percentage
* Hours\_per\_week
* Reports\_to[foreign key; references **Email** from **Dealership\_Manager**]

Table: **Service\_Manager**

Columns:

* Name [composite]
  + First\_name
  + middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Hours\_per\_week
* Reports\_to[foreign key; references **Email** from **Dealership\_Manager**]

**Part 5: Identify and write key SQL queries**

### 

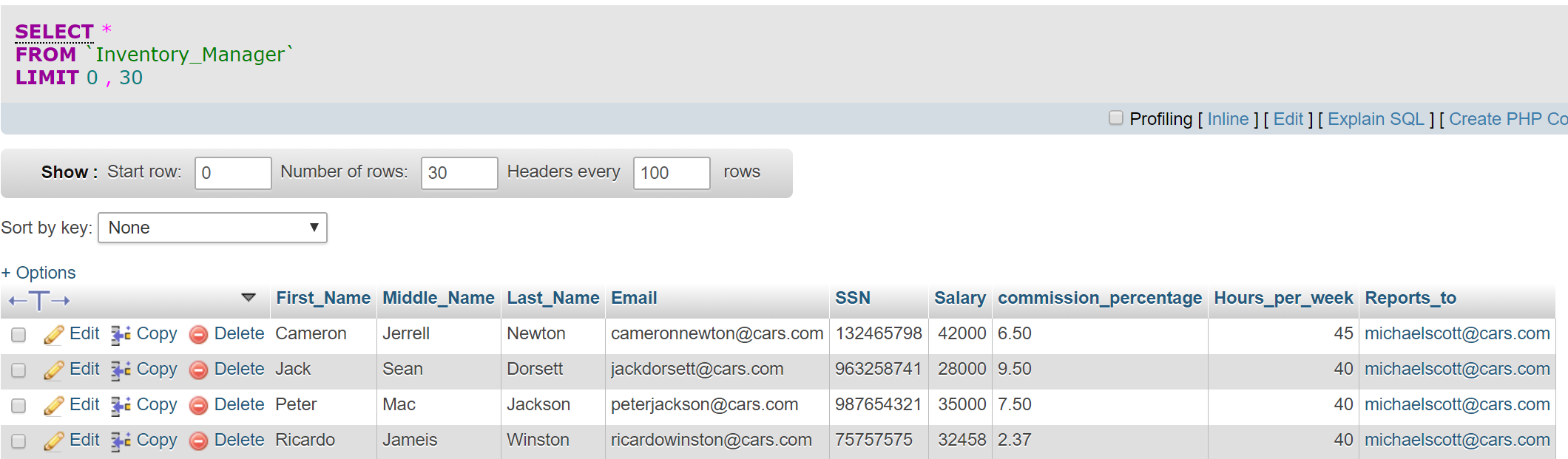
|  |  |
| --- | --- |
| US1 | As a Dealership manager, I want to add/remove Inventory Manager |

A query to add an inventory manager:

INSERT INTO `Inventory\_Manager`(`First\_Name`, `Middle\_Name`, `Last\_Name`, `Email`, `SSN`, `Salary`, `commission\_percentage`, `Hours\_per\_week`, `Reports\_to`)

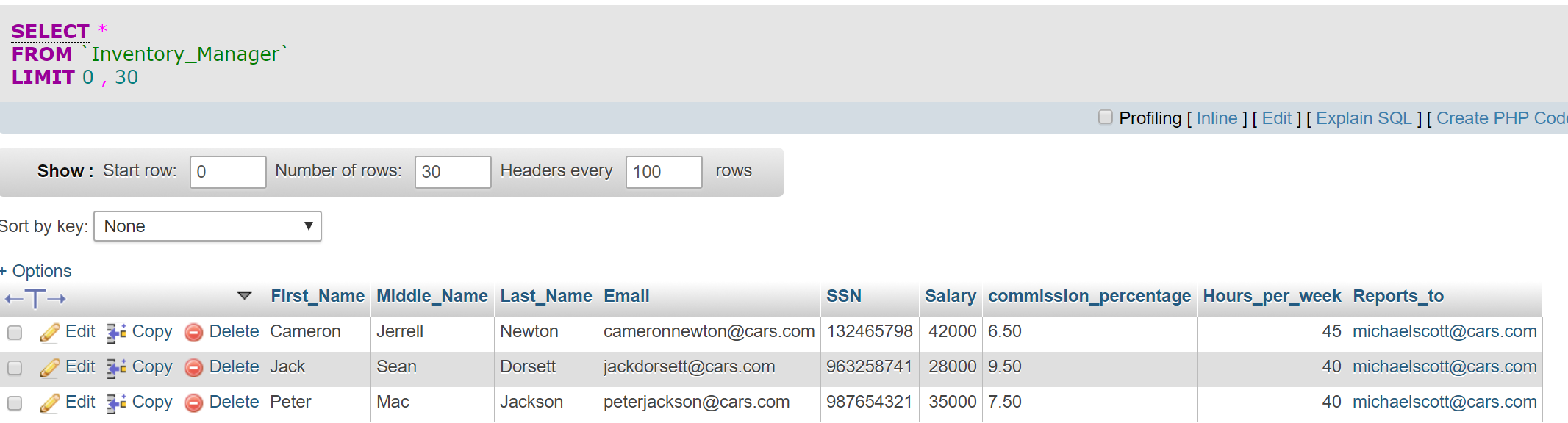
VALUES

('Ricardo','Jameis','Winston','ricardowinston@cars.com',75757575,32458,2.37,40,'michaelscott@cars.com')



A query to remove an inventory manager:

DELETE FROM `Inventory\_Manager` WHERE first\_name = 'Ricardo'



### 

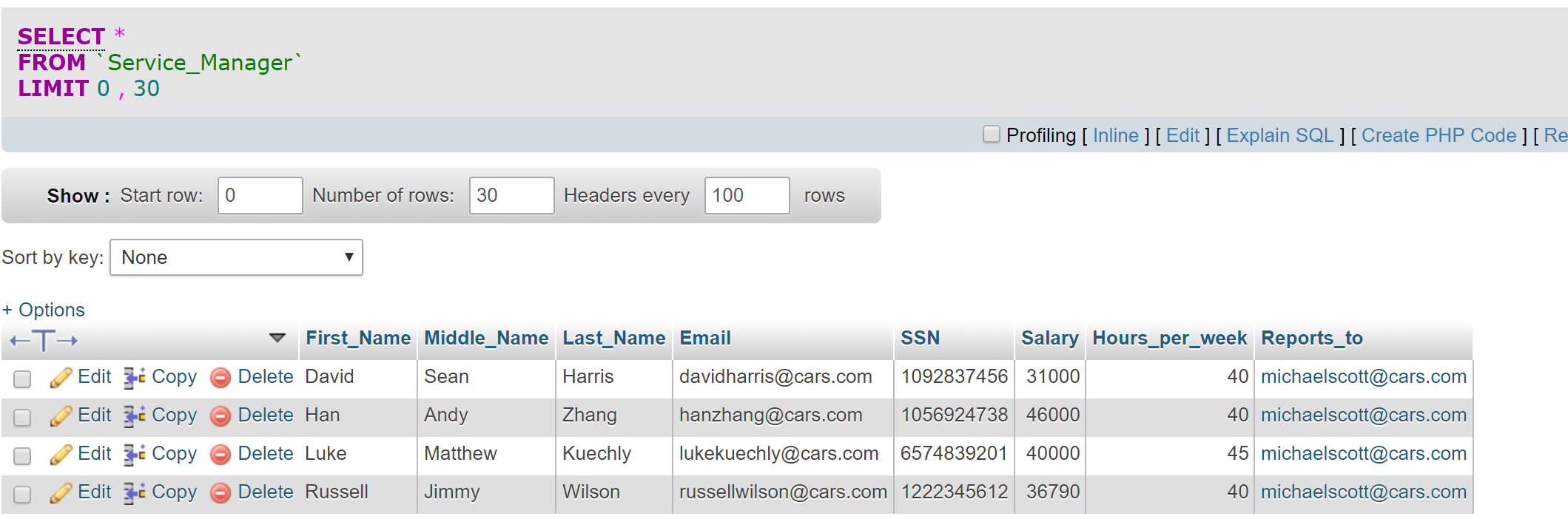
|  |  |
| --- | --- |
| US2 | As a Dealership manager, I want to add/remove Service Manager |

A query to add a service manager:

INSERT INTO `Service\_Manager`(`First\_Name`, `Middle\_Name`, `Last\_Name`, `Email`, `SSN`, `Salary`, `Hours\_per\_week`, `Reports\_to`)

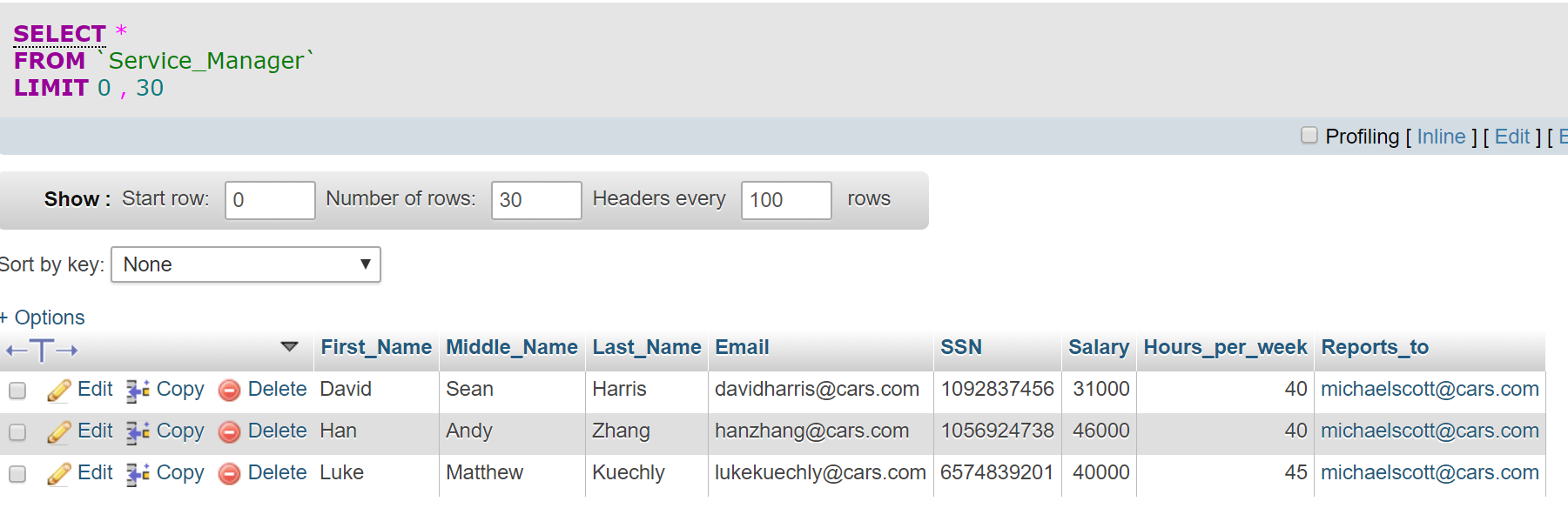
VALUES

(['Russell','Jimmy','Wilson','russellwilson@cars.com](about:blank)',1222345612,36790,40,'michaelscott@cars.com')



A query to remove a service manager:

DELETE FROM `Service\_Manager` WHERE email = 'russellwilson@cars.com'



|  |  |
| --- | --- |
| US8 | As an Inventory Manager, I want to see all new/used car inventory |

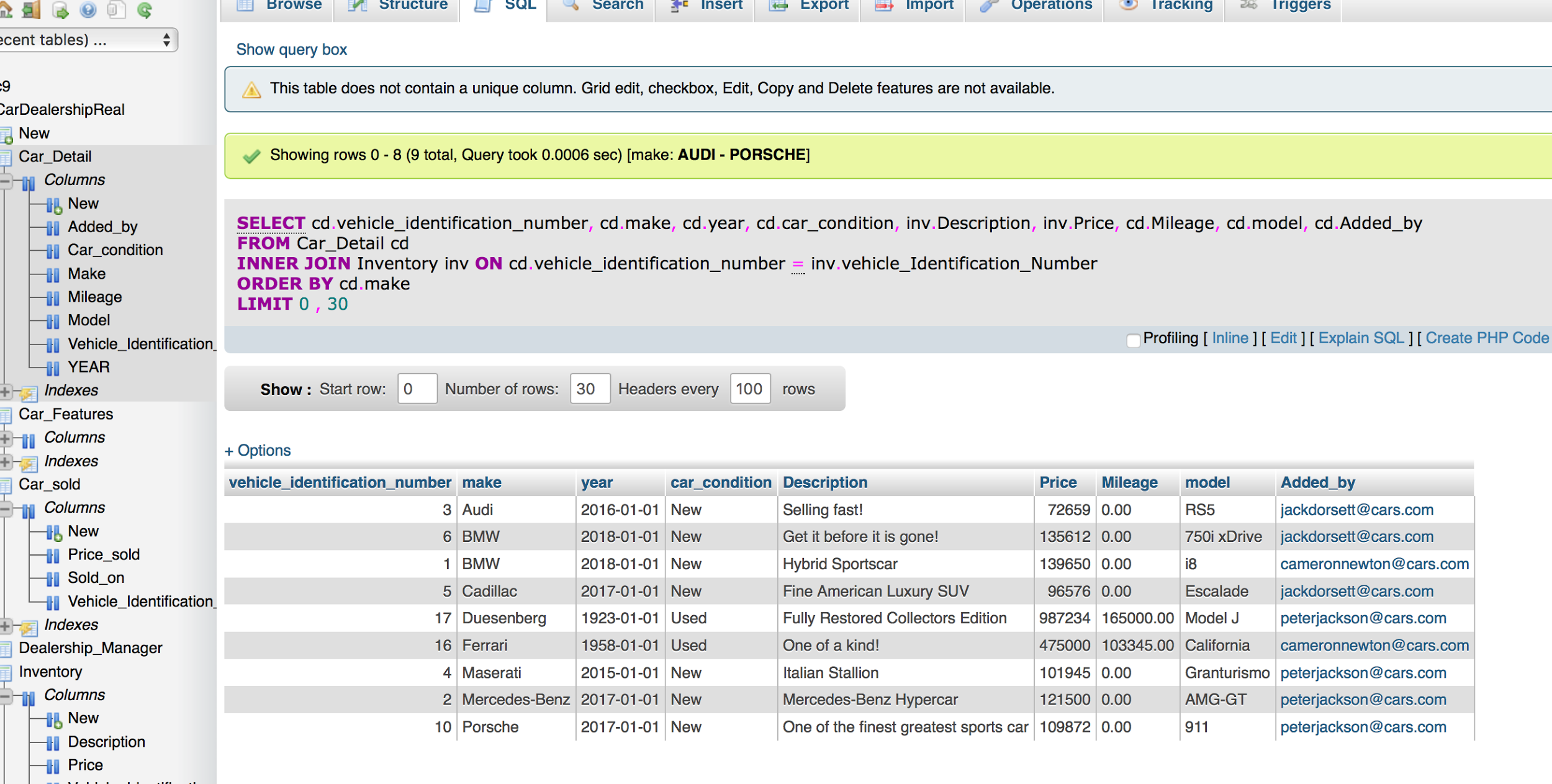
A query that gives us the information about all the cars in the inventory :

[**SELECT**](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=7fd8a2fe2bee4b14ba3db3b631473dc5) cd.vehicle\_identification\_number, cd.make, cd.year, cd.car\_condition, inv.Description, inv.Price, cd.Mileage, cd.model, cd.Added\_by

**FROM** Car\_Detail cd

**INNER** **JOIN** Inventory inv **ON** cd.vehicle\_identification\_number [=](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=7fd8a2fe2bee4b14ba3db3b631473dc5) inv.vehicle\_Identification\_Number

**ORDER** **BY** cd.make



### 

|  |  |
| --- | --- |
| US9 | As an Inventory Manager, I want to see the information about a specific new/used car |

A query that gives us the Factory,model and year of the used cars:

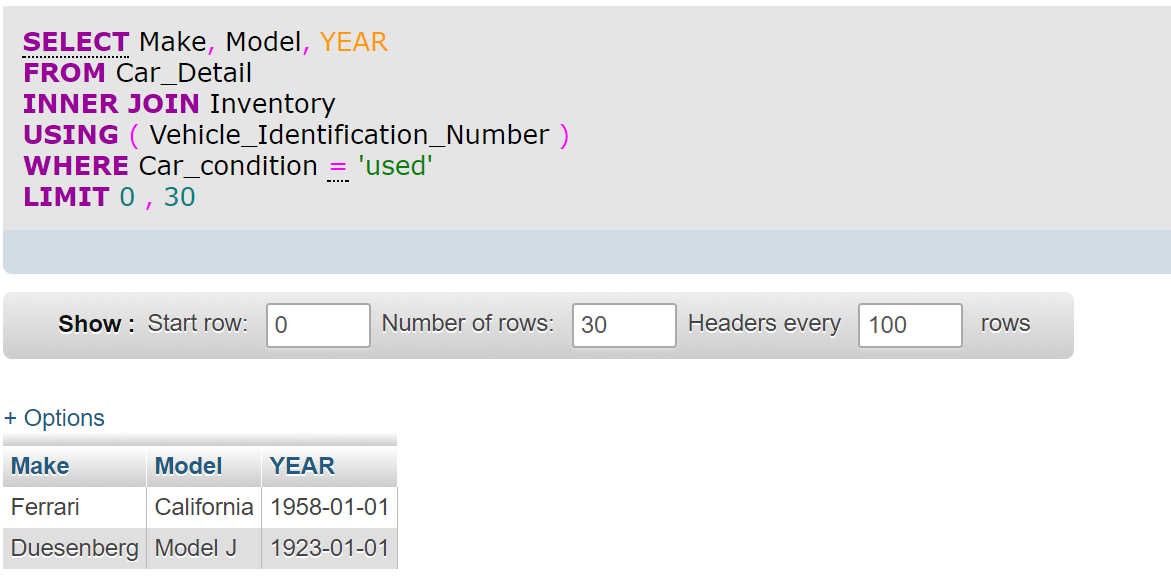
[**SELECT**](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=7c46a848129486a581bf0b3a76518b98) Make, Model, YEAR

**FROM** Car\_Detail

**INNER** **JOIN** Inventory

**USING** ( Vehicle\_Identification\_Number )

**WHERE** Car\_condition [=](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=7c46a848129486a581bf0b3a76518b98) 'used'



|  |  |
| --- | --- |
| US9 | As an Inventory Manager, I want to see the information about a specific new/used car |

A query that gives us the Factory,year,mileage,price,description and all the features of the cars,which are cheaper than 140000$:

[**SELECT**](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=7c46a848129486a581bf0b3a76518b98) Make, YEAR, Mileage, Price, Description, [GROUP\_CONCAT](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fgroup-by-functions.html%23function_group_concat&token=7c46a848129486a581bf0b3a76518b98)( feature\_name )

**FROM** Car\_Detail D

**INNER** **JOIN** Inventory

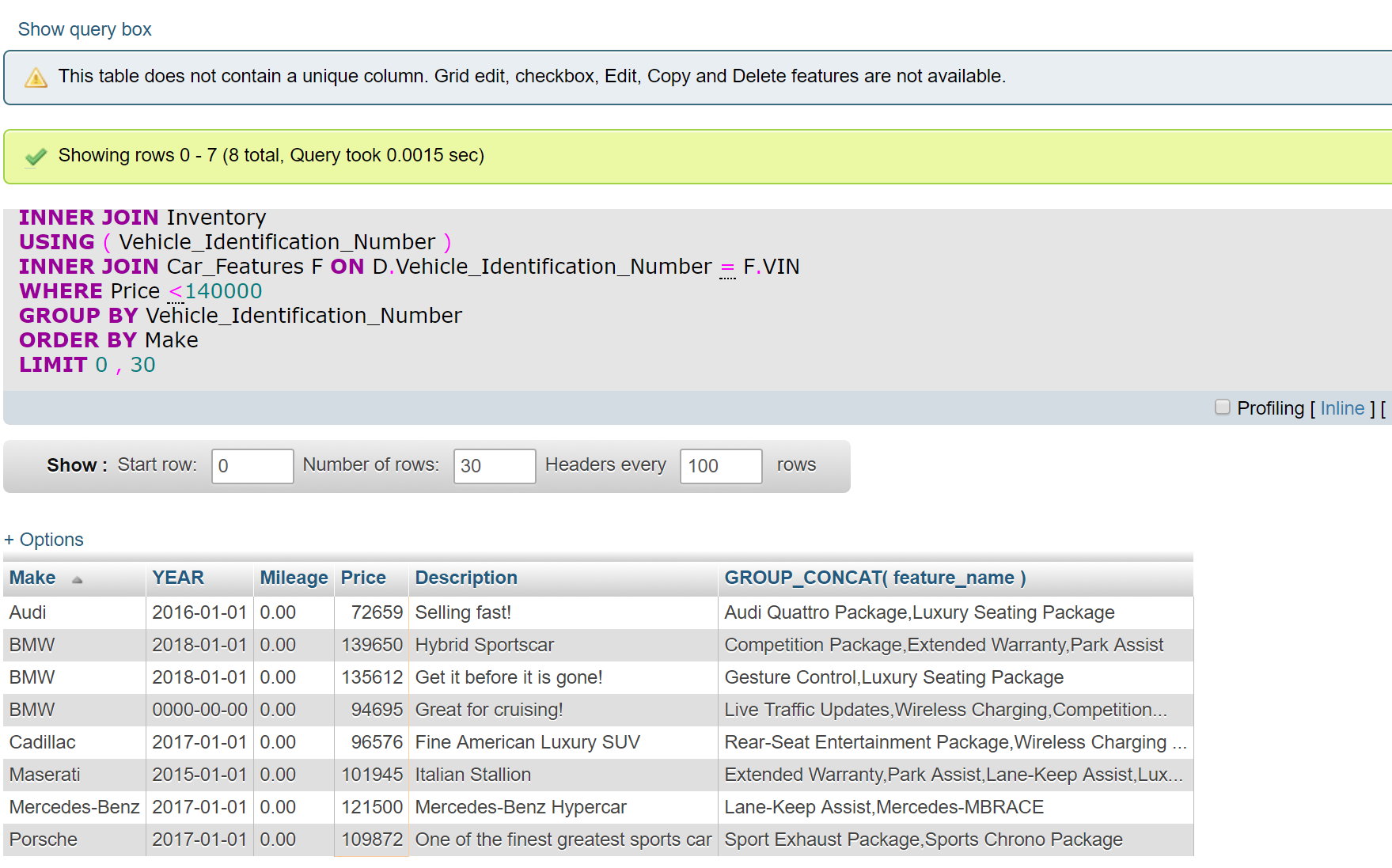
**USING** ( Vehicle\_Identification\_Number )

**INNER** **JOIN** Car\_Features F **ON** D.Vehicle\_Identification\_Number [=](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=7c46a848129486a581bf0b3a76518b98) F.VIN

**WHERE** Price [<](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_less-than&token=7c46a848129486a581bf0b3a76518b98)140000

**GROUP** **BY** Vehicle\_Identification\_Number

**ORDER** **BY** Make



### 

|  |  |
| --- | --- |
| US13 | As a Service Manager, I want to see history of cars serviced |

A query that gives us the Identification number,model,date of the service and the service manager of all the cars that their tires were changed:

[**SELECT**](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=7fd8a2fe2bee4b14ba3db3b631473dc5) Vehicle\_Identification\_Number, Make, Model, Service\_Date, Serviced\_by

**FROM** Car\_Detail

**INNER** **JOIN** Service\_History

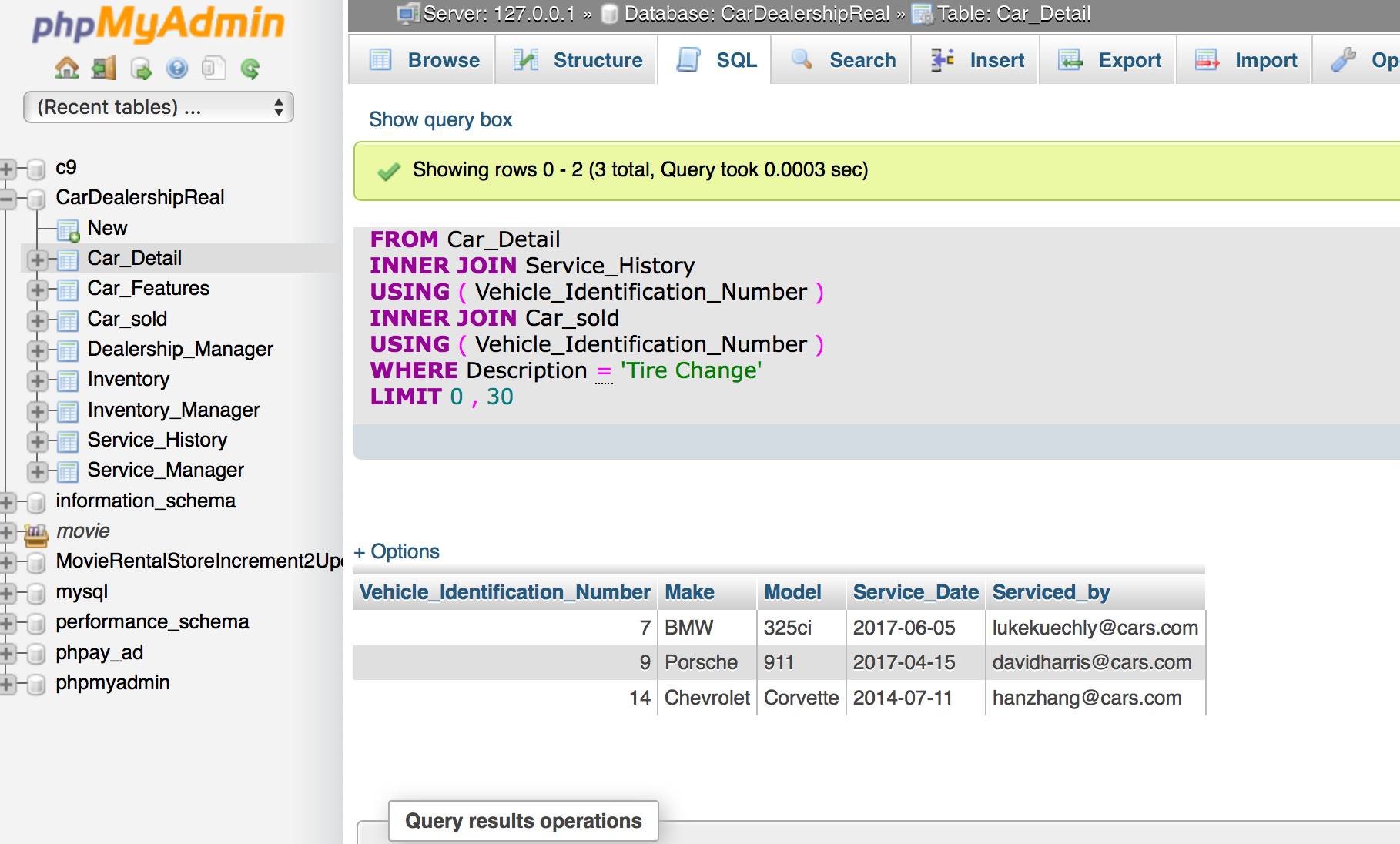
**USING** ( Vehicle\_Identification\_Number )

**INNER** **JOIN** Car\_sold

**USING** ( Vehicle\_Identification\_Number )

**WHERE** Description [=](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=7fd8a2fe2bee4b14ba3db3b631473dc5) 'Tire Change'

**LIMIT** 0 , 30



## 

|  |  |
| --- | --- |
| US14 | As a Service Manager, I want to see the service history of a specific car so that service manager get history of particular car he wants to  review. |

a.)A query that shows all the services provided along with selling and car details and to a particular sold out car for a Service Manager

[**SELECT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=114e8b26efb0e192d98e4898f36697be) D.Make, D.YEAR, Mileage, CS.Price\_sold, CS.sold\_on, [GROUP\_CONCAT](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fgroup-by-functions.html%23function_group_concat&token=114e8b26efb0e192d98e4898f36697be)( SH.Description ) **AS** services

**FROM** Car\_Detail D

**INNER** **JOIN** Service\_History SH

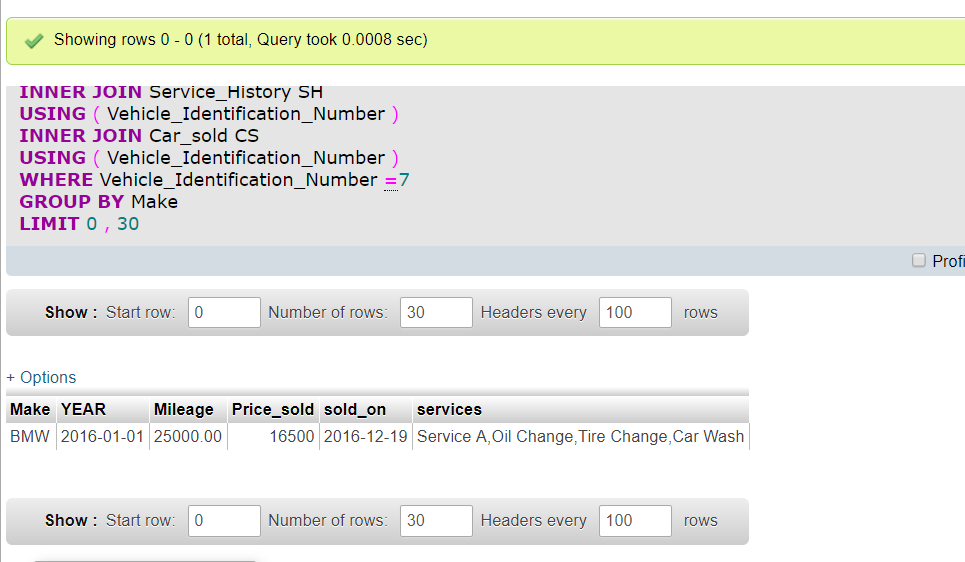
**USING** ( Vehicle\_Identification\_Number )

**INNER** **JOIN** Car\_sold CS

**USING** ( Vehicle\_Identification\_Number )

**WHERE** Vehicle\_Identification\_Number [=](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=114e8b26efb0e192d98e4898f36697be)7

**GROUP** **BY** Make



b.) A query to fetch Details for all cars if the car has services display list of services as a group or display ‘-’ and firstname of managers who serviced particular or not serviced if the car has not been serviced.

[**SELECT**](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fselect.html&token=7fd8a2fe2bee4b14ba3db3b631473dc5) cs.Vehicle\_Identification\_Number, cs.Make, cs.Model, [IFNULL](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcontrol-flow-functions.html%23function_ifnull&token=7fd8a2fe2bee4b14ba3db3b631473dc5)( [GROUP\_CONCAT](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fgroup-by-functions.html%23function_group_concat&token=7fd8a2fe2bee4b14ba3db3b631473dc5)( sm.First\_Name

**SEPARATOR** ';' ) , 'Not Serviced ' ) **AS** `Service Managers who serviced the car` , cs.car\_condition, [IFNULL](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcontrol-flow-functions.html%23function_ifnull&token=7fd8a2fe2bee4b14ba3db3b631473dc5)( [GROUP\_CONCAT](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fgroup-by-functions.html%23function_group_concat&token=7fd8a2fe2bee4b14ba3db3b631473dc5)( **DISTINCT** sh.Description

**ORDER** **BY** sh.Description **ASC**

**SEPARATOR** ',' ) , '-' ) **AS** `List Of Services Offered`

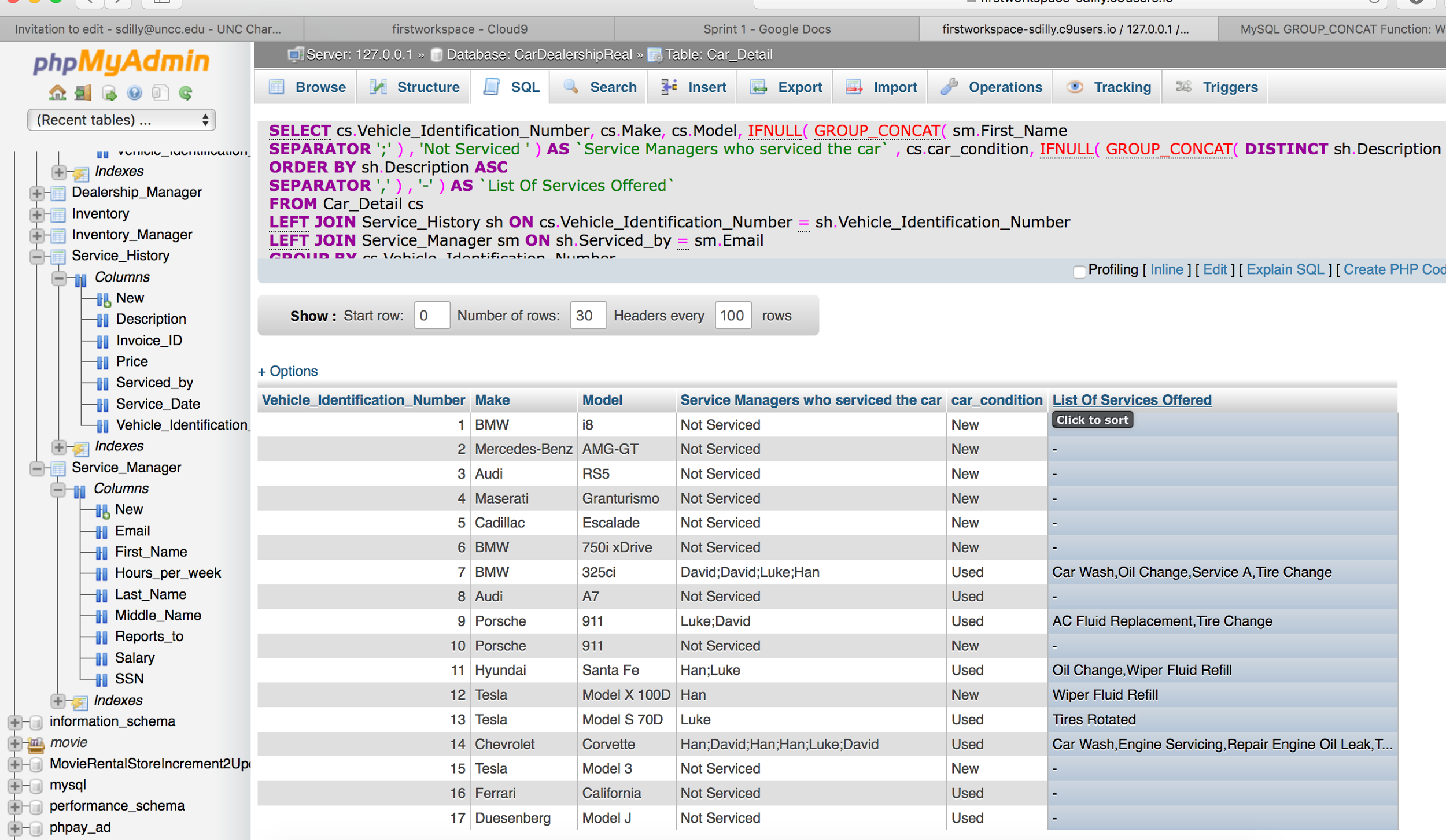
**FROM** Car\_Detail cs

[**LEFT**](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fstring-functions.html%23function_left&token=7fd8a2fe2bee4b14ba3db3b631473dc5) **JOIN** Service\_History sh **ON** cs.Vehicle\_Identification\_Number [=](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=7fd8a2fe2bee4b14ba3db3b631473dc5) sh.Vehicle\_Identification\_Number

[**LEFT**](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fstring-functions.html%23function_left&token=7fd8a2fe2bee4b14ba3db3b631473dc5) **JOIN** Service\_Manager sm **ON** sh.Serviced\_by [=](https://firstworkspace-sdilly.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcomparison-operators.html%23operator_equal&token=7fd8a2fe2bee4b14ba3db3b631473dc5) sm.Email

**GROUP** **BY** cs.Vehicle\_Identification\_Number

**LIMIT** 0 , 30



### 

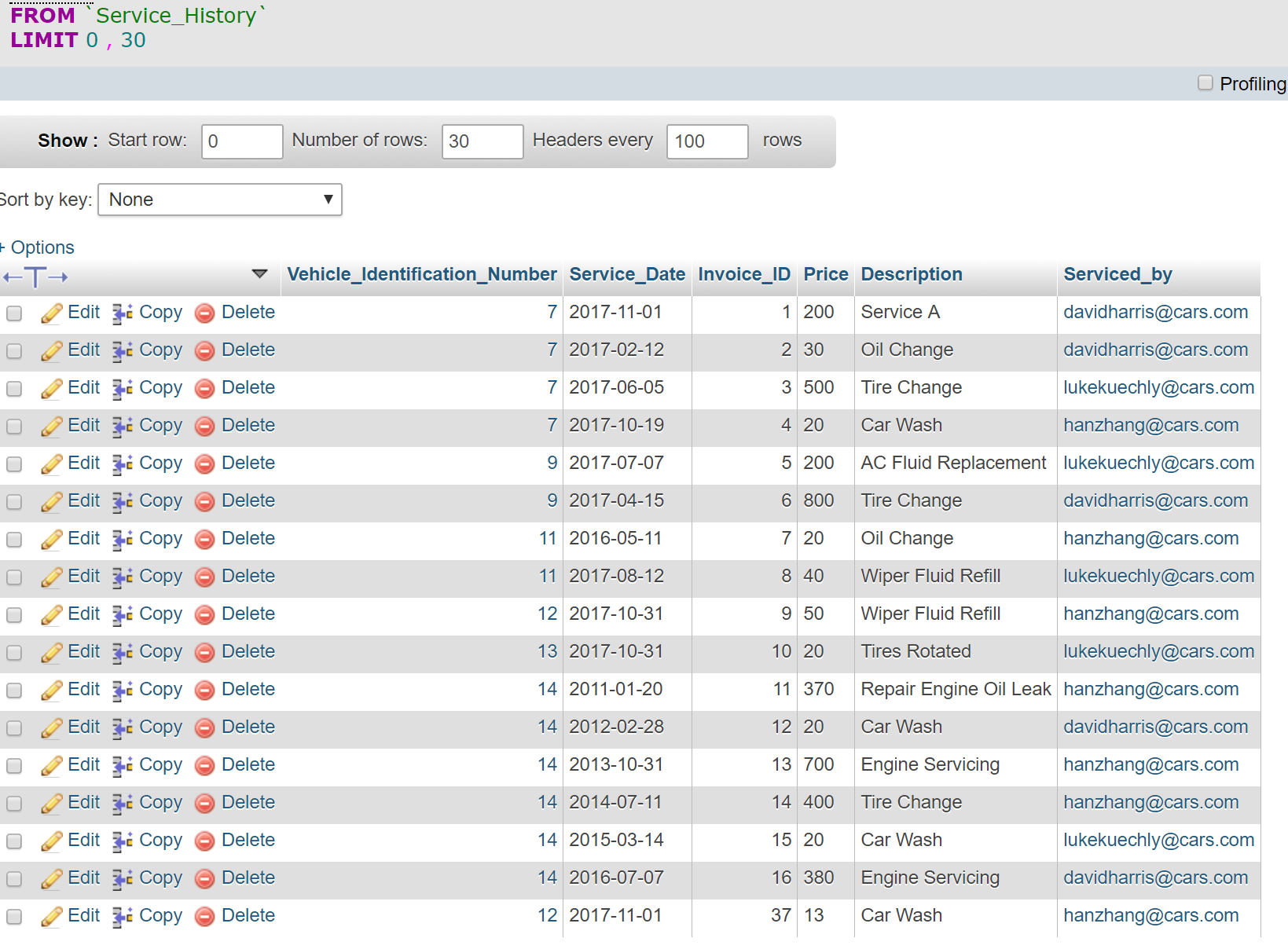
|  |  |
| --- | --- |
| US15 | As a Service Manager, I want to add/update a car service history |

Query to add service\_history to a specific car:

INSERT INTO `Service\_History`(`Vehicle\_Identification\_Number`, `Service\_Date`, `Invoice\_ID`, `Price`, `Description`, `Serviced\_by`)

VALUES

(12,STR\_TO\_DATE('1-11-2017', '%d-%m-%Y'),37,12.99,'Car Wash','hanzhang@cars.com')



**Additional SQL Queries Utilized:**

**Create Table:**

**CREATE** **TABLE** Dealership\_Manager(

First\_Name VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Middle\_Name VARCHAR( 50 ) DEFAULT **NULL** ,

Last\_Name VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Email VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

SSN VARCHAR( 20 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Salary INT( 20 ) ,

Hours\_per\_week INT( 10 ) ,

**PRIMARY** **KEY** ( Email )

);

**CREATE** **TABLE** Inventory\_Manager(

First\_Name VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Middle\_Name VARCHAR( 50 ) DEFAULT **NULL** ,

Last\_Name VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Email VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

SSN VARCHAR( 20 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Salary INT( 20 ) ,

commission\_percentage DECIMAL( 4, 2 ) DEFAULT 0,

Hours\_per\_week INT( 10 ) ,

Reports\_to VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

**PRIMARY** **KEY** ( Email ) ,

**FOREIGN** **KEY** fk\_cat( Reports\_to ) **REFERENCES** Dealership\_Manager( Email )

)

**CREATE** **TABLE** Service\_Manager(

First\_Name VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Middle\_Name VARCHAR( 50 ) DEFAULT **NULL** ,

Last\_Name VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Email VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

SSN VARCHAR( 20 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Salary INT( 20 ) ,

Hours\_per\_week INT( 10 ) ,

Reports\_to VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

**PRIMARY** **KEY** ( Email ) ,

**FOREIGN** **KEY** fk\_cat( Reports\_to ) **REFERENCES** Dealership\_Manager( Email )

)

CREATE TABLE Car\_Detail(

Vehicle\_Identification\_Number INT( 20 ) NOT NULL ,

YEAR DATE NOT NULL ,

Make VARCHAR( 200 ) DEFAULT NULL ,

Model VARCHAR( 20 ) DEFAULT NULL ,

Mileage DECIMAL( 9, 2 ) ,

Car\_condition ENUM( 'New', 'Used' ) NOT NULL ,

Added\_by VARCHAR( 50 ) NOT NULL ,

PRIMARY KEY (Vehicle\_Identification\_Number),

FOREIGN KEY fk\_car( Added\_by ) REFERENCES Inventory\_Manager( Email )

)

CREATE TABLE Inventory(

Vehicle\_Identification\_Number INT( 20 ) NOT NULL ,

Price INT( 20 ) DEFAULT NULL ,

Description VARCHAR( 200 ) DEFAULT NULL ,

PRIMARY KEY ( Vehicle\_Identification\_Number ) ,

FOREIGN KEY fk\_car( Vehicle\_Identification\_Number ) REFERENCES Car\_Detail( Vehicle\_Identification\_Number )

)

CREATE TABLE Car\_sold(

Vehicle\_Identification\_Number INT( 20 ) NOT NULL ,

Price\_sold INT( 20 ) NOT NULL ,

Sold\_on DATE NOT NULL ,

PRIMARY KEY ( Vehicle\_Identification\_Number ) ,

FOREIGN KEY fk\_car( Vehicle\_Identification\_Number ) REFERENCES Car\_Detail( Vehicle\_Identification\_Number )

)

**CREATE** **TABLE** Car\_Features(

VIN INT( 20 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

feature\_name VARCHAR( 200 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

**PRIMARY** **KEY** ( VIN, feature\_name ) ,

**FOREIGN** **KEY** fk\_fr( VIN ) **REFERENCES** Inventory( Vehicle\_Identification\_Number )

)

[**CREATE** **TABLE**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fcreate-table.html&token=114e8b26efb0e192d98e4898f36697be) Service\_History(

Vehicle\_Identification\_Number INT( 20 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Service\_Date DATE [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Invoice\_ID INT( 20 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

Price INT( 20 ) DEFAULT **NULL** ,

Description VARCHAR( 200 ) DEFAULT **NULL** ,

Serviced\_by VARCHAR( 50 ) [**NOT**](https://sql-practice-dayila.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Flogical-operators.html%23operator_not&token=114e8b26efb0e192d98e4898f36697be) **NULL** ,

**PRIMARY** **KEY** ( Invoice\_ID ) ,

**FOREIGN** **KEY** fk\_num( Vehicle\_Identification\_Number ) **REFERENCES** Car\_sold( Vehicle\_Identification\_Number ) ,

**FOREIGN** **KEY** fk\_email( Serviced\_by ) **REFERENCES** Service\_Manager( Email )

)

[**INSERT**](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Finsert.html&token=51e2121ea6a4b027deb6f6dbbe9dd43c) **INTO** Dealership\_Manager( First\_Name, Middle\_Name, Last\_Name, SSN, Hours\_per\_week, Salary, Email )

**VALUES** (

'Michael', 'Duke', 'Scott', 123456789, 40, 50000, 'michaelscott@cars.com'

)

[**INSERT**](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Finsert.html&token=51e2121ea6a4b027deb6f6dbbe9dd43c) **INTO** Inventory\_Manager( First\_Name, Middle\_Name, Last\_Name, SSN, Hours\_per\_week, Salary, commission\_percentage, Email, Reports\_to )

**VALUES** (

'Peter', 'Mac', 'Jackson', 987654321, 40, 35000, 7.5, 'peterjackson@cars.com', 'michaelscott@cars.com'

)

[**INSERT**](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Finsert.html&token=51e2121ea6a4b027deb6f6dbbe9dd43c) **INTO** Inventory\_Manager( First\_Name, Middle\_Name, Last\_Name, SSN, Hours\_per\_week, Salary, commission\_percentage, Email, Reports\_to )

**VALUES** (

'Jack', 'Sean', 'Dorsett', 963258741, 40, 28000, 9.5, 'jackdorsett@cars.com', 'michaelscott@cars.com'

), (

'Cameron', 'Jerrell', 'Newton', 132465798, 45, 42000, 6.5, 'cameronnewton@cars.com', 'michaelscott@cars.com'

)

[**INSERT**](https://car-dealership-svyas7.c9users.io/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Finsert.html&token=18fe84fcfcafcb70cdbeb1ff47ad6356) **INTO** Service\_Manager( First\_Name, Middle\_Name, Last\_Name, SSN, Hours\_per\_week, Salary, Email, Reports\_to )

**VALUES** (

'David', 'Sean', 'Harris', 1092837456, 40, 31000, 'davidharris@cars.com', 'michaelscott@cars.com'

), (

'Luke', 'Matthew', 'Kuechly', 6574839201, 45, 40000, 'lukekuechly@cars.com', 'michaelscott@cars.com'

), (

'Han', 'Andy', 'Zhang', 1056924738, 40, 46000, 'hanzhang@cars.com', 'michaelscott@cars.com'

)

Insert into Car(Vehicle\_Identification\_Number, YEAR, Make, Model, Car\_condition, Mileage, Description, Price, Added\_by)

Values

(197346825,STR\_TO\_DATE( '1-01-2018', '%d-%m-%Y' ),'BMW', 'i8','New',0,'Excellent Condition!!!',125000,'cameronnewton@cars.com'),

(512397468,STR\_TO\_DATE( '1-01-2016', '%d-%m-%Y' ),'Audi', 'A5','New',0,'Fully Loaded!',58000,'jackdorsett@cars.com'),

(999001785,STR\_TO\_DATE( '1-01-2015', '%d-%m-%Y' ),'Mercedes-Benz', 'E400','New',0,'Reduced Price!',56256,'peterjackson@cars.com'),

(777755521,STR\_TO\_DATE( '1-01-2018', '%d-%m-%Y' ),'Maserati', 'Quattroporte','New',0,'Italian Stallion',91253,'jackdorsett@cars.com'),

(976431258,STR\_TO\_DATE( '1-01-2017', '%d-%m-%Y' ),'Cadillac', 'Escalade','New',0,'American Luxury SUV',91690,'peterjackson@cars.com'),

(134679258,STR\_TO\_DATE( '1-01-2012', '%d-%m-%Y' ),'Hyundai', 'Santa Fe','Used',20000,'Only One Previous Owner',20900,'jackdorsett@cars.com'),

(963135790,STR\_TO\_DATE( '1-01-1997', '%d-%m-%Y' ),'Chevrolet', 'Tahoe','Used',150985,'Great Financing!',6985,'peterjackson@cars.com'),

(108335017,STR\_TO\_DATE( '1-01-1958', '%d-%m-%Y' ),'Ferrari', 'California Spider','Used',40000,'Collectible Classic Car!',150000,'cameronnewton@cars.com'),

(100025897,STR\_TO\_DATE( '1-01-1925', '%d-%m-%Y' ),'Duesenberg', 'Model J','Used',98562,'One-of-a-kind classic!',600000,'cameronnewton@cars.com'),

(336699885,STR\_TO\_DATE( '1-01-2006', '%d-%m-%Y' ),'BMW', '325i','Used',89652,'Great condition',12500,'cameronnewton@cars.com')

INSERT INTO `Car\_sold`(`Vehicle\_Identification\_Number`, `YEAR`, `Make`, `Model`, `Price\_sold`, `Mileage`, `Car\_condition`, `Sold\_on`)

VALUES

(00000001,STR\_TO\_DATE('1-01-2016', '%d-%m-%Y'),'BMW','M4',89250,0,'New',STR\_TO\_DATE('15-04-2016', '%d-%m-%Y' )),

(00000002,STR\_TO\_DATE('1-01-2015', '%d-%m-%Y'),'Mercedes-Benz','C63 AMG',91950,0,'New',STR\_TO\_DATE('02-12-2015','%d-%m-%Y' )),

(00000013,STR\_TO\_DATE('1-01-2012', '%d-%m-%Y'),'Audi','RS5',72250,0,'New',STR\_TO\_DATE('25-06-2013', '%d-%m-%Y' )),

(00000231,STR\_TO\_DATE('1-01-2010', '%d-%m-%Y'),'Toyota','Corolla',13260,5600,'Used',STR\_TO\_DATE('19-12-2014', '%d-%m-%Y' )),

(00005123,STR\_TO\_DATE('1-01-2011', '%d-%m-%Y'),'Chevrolet','Corvette',69100,9267,'Used',STR\_TO\_DATE('01-03-2013', '%d-%m-%Y' )),

(00001489,STR\_TO\_DATE('1-01-2017', '%d-%m-%Y'),'Tesla','Model S 100D',109250,0,'New',STR\_TO\_DATE('06-12-2017', '%d-%m-%Y' )),

(00000003,STR\_TO\_DATE('1-01-2013', '%d-%m-%Y'),'BMW','740i',41250,72500,'Used',STR\_TO\_DATE('19-09-2015', '%d-%m-%Y' )),

(00000004,STR\_TO\_DATE('1-01-2014', '%d-%m-%Y'),'Tesla','Model X',45250,25000,'Used',STR\_TO\_DATE('11-1-2017', '%d-%m-%Y' )),

(00000005,STR\_TO\_DATE('1-01-2016', '%d-%m-%Y'),'Mercedes-Benz','AMG GT',109250,1950,'Used',STR\_TO\_DATE('05-11-2017', '%d-%m-%Y' )),

(00000006,STR\_TO\_DATE('1-01-2018', '%d-%m-%Y'),'BMW','i8',149100,0,'New',STR\_TO\_DATE('15-10-2017', '%d-%m-%Y' ))

------

CREATE TABLE Car\_Detail(

Vehicle\_Identification\_Number INT( 20 ) NOT NULL ,

YEAR DATE NOT NULL ,

Make VARCHAR( 200 ) DEFAULT NULL ,

Model VARCHAR( 20 ) DEFAULT NULL ,

Mileage DECIMAL( 9, 2 ) ,

Car\_condition ENUM( 'New', 'Used' ) NOT NULL ,

Added\_by VARCHAR( 50 ) NOT NULL ,

PRIMARY KEY (Vehicle\_Identification\_Number),

FOREIGN KEY fk\_car( Added\_by ) REFERENCES Inventory\_Manager( Email )

)

CREATE TABLE Inventory(

Vehicle\_Identification\_Number INT( 20 ) NOT NULL ,

Price INT( 20 ) DEFAULT NULL ,

Description VARCHAR( 200 ) DEFAULT NULL ,

PRIMARY KEY ( Vehicle\_Identification\_Number ) ,

FOREIGN KEY fk\_car( Vehicle\_Identification\_Number ) REFERENCES Car\_Detail( Vehicle\_Identification\_Number )

)

CREATE TABLE Car\_sold(

Vehicle\_Identification\_Number INT( 20 ) NOT NULL ,

Price\_sold INT( 20 ) NOT NULL ,

Sold\_on DATE NOT NULL ,

PRIMARY KEY ( Vehicle\_Identification\_Number ) ,

FOREIGN KEY fk\_car( Vehicle\_Identification\_Number ) REFERENCES Car\_Detail( Vehicle\_Identification\_Number )

)

INSERT INTO `Car\_Detail`(`Vehicle\_Identification\_Number`, `YEAR`, `Make`, `Model`, `Mileage`, `Car\_condition`, `Added\_by`)

VALUES (1,STR\_TO\_DATE('1-01-2018','%d-%m-%Y'),'BMW','i8',0,'New','cameronnewton@cars.com')

INSERT INTO `Car\_Detail`(`Vehicle\_Identification\_Number`, `YEAR`, `Make`, `Model`, `Mileage`, `Car\_condition`, `Added\_by`)

VALUES

(2,STR\_TO\_DATE('1-01-2017','%d-%m-%Y'),'Mercedes-Benz','AMG-GT',0,'New','peterjackson@cars.com'),

(3,STR\_TO\_DATE('1-01-2016','%d-%m-%Y'),'Audi','RS5',0,'New','jackdorsett@cars.com'),

(4,STR\_TO\_DATE('1-01-2015','%d-%m-%Y'),'Maserati','Granturismo',0,'New','peterjackson@cars.com'),

(5,STR\_TO\_DATE('1-01-2017','%d-%m-%Y'),'Cadillac','Escalade',0,'New','jackdorsett@cars.com'),

(6,STR\_TO\_DATE('1-01-2018','%d-%m-%Y'),'BMW','750i xDrive',0,'New','jackdorsett@cars.com'),

(10,STR\_TO\_DATE('1-01-2017','%d-%m-%Y'),'Porsche','911',0,'New','peterjackson@cars.com'),

(7,STR\_TO\_DATE('1-01-2016','%d-%m-%Y'),'BMW','325ci',25000,'Used','jackdorsett@cars.com'),

(8,STR\_TO\_DATE('1-01-2012','%d-%m-%Y'),'Audi','A7',125000,'Used','cameronnewton@cars.com'),

(9,STR\_TO\_DATE('1-01-2010','%d-%m-%Y'),'Porsche','911',103345,'Used','cameronnewton@cars.com'),

(11,STR\_TO\_DATE('1-01-2013','%d-%m-%Y'),'Hyundai','Santa Fe',45567,'Used','cameronnewton@cars.com'),

(12,STR\_TO\_DATE('1-01-2017','%d-%m-%Y'),'Tesla','Model X 100D',0,'New','cameronnewton@cars.com'),

(13,STR\_TO\_DATE('1-01-2015','%d-%m-%Y'),'Tesla','Model S 70D',41938,'Used','peterjackson@cars.com'),

(14,STR\_TO\_DATE('1-01-1997','%d-%m-%Y'),'Chevrolet','Corvette',76812,'Used','peterjackson@cars.com'),

(15,STR\_TO\_DATE('1-01-2017','%d-%m-%Y'),'Tesla','Model 3',0,'New','cameronnewton@cars.com'),

(16,STR\_TO\_DATE('1-01-1958','%d-%m-%Y'),'Ferrari','California',103345,'Used','cameronnewton@cars.com'),

(17,STR\_TO\_DATE('1-01-1923','%d-%m-%Y'),'Duesenberg','Model J',165000,'Used','peterjackson@cars.com')

INSERT INTO `Inventory`(`Vehicle\_Identification\_Number`, `Price`, `Description`)

VALUES

(1,139650,'Hybrid Sportscar'),

(2,121500,'Mercedes-Benz Hypercar'),

(3,72659,'Selling fast!'),

(4,101945,'Italian Stallion'),

(5,96576, 'Fine American Luxury SUV'),

(6,135612, 'Get it before it is gone!'),

(10,109872, 'One of the finest greatest sports car'),

(16,475000,'One of a kind!'),

(17,987234,'Fully Restored Collectors Edition')

INSERT INTO `Car\_sold`(`Vehicle\_Identification\_Number`, `Price\_sold`, `Sold\_on`)

VALUES

(7,16500,STR\_TO\_DATE('19-12-2016','%d-%m-%Y')),

(8,23405,STR\_TO\_DATE('1-11-2017','%d-%m-%Y')),

(9,21304,STR\_TO\_DATE('12-2-2016','%d-%m-%Y')),

(11,9999,STR\_TO\_DATE('18-9-2015','%d-%m-%Y')),

(12,114000,STR\_TO\_DATE('1-04-2017','%d-%m-%Y')),

(13,38952,STR\_TO\_DATE('20-10-2017','%d-%m-%Y')),

(14,12999,STR\_TO\_DATE('11-08-2010','%d-%m-%Y')),

(15,37875,STR\_TO\_DATE('01-11-2017','%d-%m-%Y'))

INSERT INTO `Car\_Features`(`VIN`, `feature\_name`)

VALUES

(1,'Park Assist'),

(1,'Competition Package'),

(1,'Extended Warranty'),

(2,'Lane-Keep Assist'),

(2,'Mercedes-MBRACE'),

(3,'Luxury Seating Package'),

(3,'Audi Quattro Package'),

(4,'Park Assist'),

(4,'Lane-Keep Assist'),

(4,'Luxury Seating Package'),

(4,'Extended Warranty'),

(5,'Rear-Seat Entertainment Package'),

(5,'Luxury Seating Package'),

(5,'Wireless Charging Pad'),

(6,'Gesture Control'),

(6,'Luxury Seating Package'),

(10,'Sports Chrono Package'),

(10,'Sport Exhaust Package'),

(16,'Fully Restored'),

(16,'Rossa Corsa Paint'),

(17,'Fully Restored'),

(17,'Refinished Chrome')

## ***SPRINT 2***

## ***Refined User Stories***

|  |  |
| --- | --- |
| **Story ID** | **Story Description** |
| US1 | As a Dealership manager, I want to add/remove Inventory Manager |
| US2 | As a Dealership manager, I want to add/remove Service Manager |
| US3 | As a Dealership manager, I want to add/remove Serviceman |
| US4 | As a Dealership manager, I want to add/remove Salesman |
| US5 | As a Guest, I want to register with the dealership and become a Registered Customer. |
| US6 | As an Inventory Manager, I want to add new/used cars to the new car inventory |
| US7 | As an Inventory Manager, I want to remove new/used cars from the inventory |
| US8 | As an Inventory Manager, I want to see all new/used car inventory |
| US9 | As an Inventory Manager, I want to see the information about a specific new/used car |
| US10 | As a Salesman, I want to see all new/used car inventory |
| US11 | As a Salesman, I want to see information about a specific new/used car |
| US12 | As a Customer(guest)/Registered Customer, I want to see brief information about the available new/used cars |
| US13 | As a Service Manager, I want to see history of cars serviced |
| US14 | As a Service Manager, I want to see the service history of a specific car |
| US15 | As a Service Manager, I want to add/update a car service history |
| US16 | As a Serviceman, I want to update a car’s service history |
| US17 | As a Serviceman, I want to view the service history of a car |
| US18 | As a Registered Customer I want to see the history of my car servicing process |
| US19 | As a Dealership manager, I want to see a list of all employees/customers |
| US20 | As a Dealership manager, I want to see information about a specific employee/customer |
| US21 | As a Guest/Registered Customer, I want to browse information about the available services |
| US22 | As a Registered Customer, I want to review my purchase history. |
| US23 | As a Registered Customer, I should be able to buy cars from catalog. |
| US24 | As an Inventory Manager, I want to add my working hours to my daily schedule sheet |
| US25 | As an Inventory Manager, I want to see my working and earning history |
| US26 | As a Salesman, I want to add my working hours to my daily schedule sheet |
| US27 | As a Salesman, I want to see my working and earning history |
| US28 | As a Service Manager, I want to add my working hours to my daily schedule sheet |
| US29 | As a Service Manager, I want to see my working and earning history |
| US30 | As a Serviceman, I want to add my working hours to my daily schedule sheet |
| US31 | As a Serviceman, I want to see my working and earning history |
| US32 | As a Dealership manager, I want to add my working hours to my daily schedule sheet |
| US33 | As a Dealership Manager, I want to see my working and earning history |
| US34 | As a Service Manager, I want to access to Serviceman working history |
| US35 | As a Inventory Manager, I want to access Salesman working history. |
| US36 | As a Dealership Manager, I want to access all employee working history. |

## ***REQUIREMENTS***

|  |  |
| --- | --- |
| US3 | As a Dealership manager, I want to add/remove Serviceman so that new serviceman employees can be added and terminated serviceman can be removed |
| US4 | As a Dealership manager, I want to add/remove Salesman so that new salesman can be added in the system and terminated salesman can be removed from the system |
| US5 | As a Guest, I want to register with the dealership and become a Registered Customer so that I can buy cars and view my car’s service history. |
| US16 | As a Serviceman, I want to update a car’s service history so that it reflects the latest services performed on the customer’s car. |
| US17 | As a Serviceman, I want to view the service history of a car so that I can see what services have been performed on a car. |
| US19 | As a Dealership manager, I want to see a list of all employees/customers |
| US20 | As a Dealership manager, I want to see information about a specific employee/customer |
| US21 | As a Guest/Registered Customer, I want to browse information about the available services so that I can see if the service I want to get done is offered at the dealership. |
| US23 | As a Registered Customer, I should be able to buy cars from the dealership’s inventory. |
| US37 | As a salesman, I want to see all the features available on cars so that I can understand all the ways a car can be configured. |

## Conceptual Design

Entity: **Car\_Detail**

Attributes:

* Vehicle Identification Number
* Year
* Make
* Model
* Mileage
* Car\_Condition

Entity: **Inventory**

Attributes:

* Price
* Description

Entity: **Cars\_Sold**

Attributes:

* Sold\_on
* price\_sold

Entity: **Service\_History**

Attributes:

* Invoice\_id
* Service\_date
* Description
* Price

Entity: **Employee**

Attributes:

* Name [composite]
  + First\_name
  + Middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Hours\_per\_week

Entity: **Dealership\_Manager**

Attributes:

* start\_date

Entity: **Inventory\_Manager**

Attributes:

* Commission\_percentage
* Reports\_to

Entity: **Service\_Manager**

Attributes:

* Start\_date
* Reports\_to

Entity: **Serviceman**

Attributes:

* Reports\_to

Entity: **Salesman**

Attributes:

* Reports\_to
* Commission\_percentage

Entity: **Features\_List**

Attributes:

* Feature\_name

Entity: **Service\_List**

Attributes:

* Service\_name

Entity: **Customer**

Attributes:

* Name [composite]
  + First\_name
  + Middle\_name
  + Last\_name
* Email
* Address

Relationships:

* Dealership\_Manager **registers** Inventory\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory\_Manager has total participation
* Dealership\_Manager **IS-A** Employee
  + Cardinality : one to one
  + Participation:
    - Employee has partial participation
    - Dealarship\_Manager has total participation
* Service\_Manager **IS-A** Employee
  + Cardinality : one to one
  + Participation:
    - Employee has partial participation
    - Service\_Manager has total participation
* Inventory\_Manager **IS-A** Employee
  + Cardinality : one to one
  + Participation:
    - Employee has partial participation
    - Inventory\_Manager has total participation
* ServiceMan **IS-A** Employee
  + Cardinality : one to one
  + Participation:
    - Employee has partial participation
    - ServiceMan has total participation
* Salesman **IS-A** Employee
  + Cardinality : one to one
  + Participation:
    - Employee has partial participation
    - Salesman has total participation
* Dealership\_Manager **removes** Inventory\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory\_Manager has total participation
* Dealership\_Manager **registers** Service\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Service\_Manager has total participation
* Dealership\_Manager **removes** Service\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Service\_Manager has total participation
* Inventory\_Manager **adds** new/used Inventory
  + Cardinality : one to many
  + Participation:
    - Inventory\_Manager has partial participation
    - Inventory has total participation
* Inventory\_Manager **removes** new/used Inventory
  + Cardinality : one to many
  + Participation:
    - Inventory\_Manager has partial participation
    - Inventory has total participation
* Inventory\_Manager **views** Inventory
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory has partial participation
* Serviceman **adds** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Serviceman **updates** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Serviceman **views** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Serviceman **reports to** Service\_Manager
  + Cardinality: many to one
  + Participation:
    - Serviceman has total participation
    - Service\_Manager has partial participation
* ServiceMan **adds** Service\_History
  + Cardinality : one to many
  + Participation:
    - ServiceMan has partial participation
    - Service\_History has total participation
* ServiceMan **updates** Service\_History
  + Cardinality : one to many
  + Participation:
    - ServiceMan has partial participation
    - Service\_History has total participation
* ServiceMan **views** Service\_History
  + Cardinality : one to many
  + Participation:
    - ServiceMan has partial participation
    - Service\_History has total participation
* Service\_Manager **views** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Car\_Sold **has** Service\_History
  + Cardinality : one to many
  + Participation:
    - Car has total participation
    - Service\_Histoy has total participation
* Inventory **IS-A** Car\_Detail
  + Cardinality : one to one
  + Participation:
    - Inventory has total participation
    - Car\_Detail is Partial participation
* Car\_Sold **IS-A** Car\_Detail
  + Cardinality : one to one
  + Participation:
    - Car\_Detail has partial participation
    - Car\_Sold Detail is Total participation
* Customer **bought** Car\_sold
  + Cardinality: one to many
  + Participation:
    - Customer has partial participation
    - Car\_sold has total participation
* Salesman **reports to** Inventory\_Manager
  + Cardinality: many to one
  + Participation:
    - Salesman has total participation
    - Inventory\_Manager has partial participation

## Logical Design

Table: **Car\_Detail**

Columns:

* Vehicle Identification Number [primary\_key]
* Year
* Make
* Model
* Mileage
* Car\_Condition
* Added\_by[foreign key; references **inventory\_manager\_id** from **Inventory\_Manager**]

Table: **Inventory**

Columns:

* Vehicle Identification Number [foreign key; references Vehicle Identification Number from **Car\_Detail**] [primary key]
* Description
* Price

Table: **Car\_Sold**

Columns:

* Vehicle Identification Number [foreign key; references Vehicle Identification Number from **Car\_Detail**] [primary key]
* Sold\_on
* Price\_sold
* Sold\_by[foreign key; references **salesman\_id** from **Salesman**]
* Customer\_id[foreign key; references **customer\_id** from Customer]

Table: **Car\_Features**

Columns:

* VIN [foreign key; references **VIN** from **Inventory**]
* Feature\_Id [foreign key; references **Feature\_id** from **Features\_List**]

Table: **Features\_List**

Columns:

* Feature\_id
* Feature\_name

Table: **Services\_List**

Columns

* Service\_id
* Service\_name

Table:**Service\_History**

Columns:

* Invoice\_id
* Vehicle\_Identification\_Number [foreign key; references Vehicle Identification Number from **Car\_sold**]
* Service\_date
* Service\_id[foreign key; references **Service\_id** from **Services\_List**]
* Price
* Serviced\_by[foreign key; references **serviceman\_id** from **Serviceman**]
* Customer\_id[foreign key; references **customer\_id** from **Customer**]

Table: **Employee**

Columns:

* Emp\_id
* Name [composite]
  + First\_name
  + middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Hours\_per\_week

Table: **Dealership\_Manager**

Columns:

* start\_date
* dealership\_manager\_id [foreign key; references **Emp\_id** from **Employee**]

Table: **Inventory\_Manager**

Columns:

* Commission\_percentage
* Reports\_to[foreign key; references **dealership\_manager\_id** from **Dealership\_Manager**]
* inventory\_manager\_id[foreign key; references **Emp\_id** from **Employee**]

Table: **Service\_Manager**

Columns:

* Reports\_to[foreign key; references **dealership\_manager\_id** from **Dealership\_Manager**]
* service\_manager\_id[foreign key; references **Emp\_id** from **Employee**]

Table: **Serviceman**

Columns:

* serviceman\_id[foreign key; references **Emp\_id** from **Employee**]
* Reports\_to[foreign key; references **service\_manager\_id** from **Service\_Manager**]

Table: **Salesman**

Columns:

* salesman\_id[foreign key; references **Emp\_id** from **Employee**]]
* Commission\_percentage
* Reports\_to[foreign key; references **inventory\_manager\_id** from **Inventory\_manager**]

Table: **Customer**

Columns:

* Customer\_id
* Name [composite]
  + First\_name
  + middle\_name
  + Last\_name
* Email
* Address
  + First\_line
  + Second\_line
  + City
  + State
  + Zip\_code

## ***VIEWS AND STORED PROGRAMSVIEWS AND STORED PROGRAMS***

**View**: viewCustList

Goal: This view contains a list of all the dealership’s customers as well as some of their personal information. It can be useful for the Dealership manager to see information about his/her customers.

**View**: viewEmployeeList

Goal: This view contains a list of all the dealership’s employees as well as some of their personal information. It will primarily be used by the Dealership manager to keep track of his employees.

**View**: viewFeatures

Goal: This view contains a list of all features that can be added to a car. It is useful for the customer to see what options are available on a car and useful for the salesman in helping a customer configure his/her car.

**View**: viewInventory

Goal: This view will list information about all the current cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**View**: viewNewInventory

Goal: This view will list information about all the current new cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**View**: viewUsedInventory

Goal: This view will list information about all the current used cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**View**: viewServices

Goal: This view contains a list of all the services offered by the service center of the dealership. This will be helpful to the customer as they will be able to see if the dealership can service his/her car.

**Stored procedure**: GetCarSoldInfo

Parameters: In car\_vin

Goal: This procedure will return the information about a specific car purchase given the car’s VIN. It will be useful for the customer and salesman to get information about a past transacation.

**Stored procedure**: GetCustomerPurchaseHistory

Parameters: In cust\_id

Goal: This procedure will return a customer’s purchase history given his/her customer\_id. It will be useful to the salesman in identifying candidates that may be interested in buying another car.

**Stored procedure**: GetCarServiceHistory

Parameters: In car\_vin

Goal: This procedure will get the service history of a specific car. It will be used by the customer and service managers so that they can see if a car may need servicing in the future.

**Stored procedure**: GetCustomerServiceHistory

Parameters: In cust\_id

Goal: This procedure will return the service history of a customer’s car. This will be helpful to the customer in keeping track of when his car may need servicing.

**Stored procedure**: GetSpecificInventory

Parameters: In car\_vin

Goal: This procedure will provide information about a specific car for sale given a VIN. This will be useful to the salesman and customers because it will allow them to focus on a specific car in the inventory.

**View**: viewCustList

Goal: This view contains a list of all the dealership’s customers as well as some of their personal information. It can be useful for the Dealership manager to see information about his/her customers.

**View**: viewEmployeeList

Goal: This view contains a list of all the dealership’s employees as well as some of their personal information. It will primarily be used by the Dealership manager to keep track of his employees.

**View**: viewFeatures

Goal: This view contains a list of all features that can be added to a car. It is useful for the customer to see what options are available on a car and useful for the salesman in helping a customer configure his/her car.

**View**: viewInventory

Goal: This view will list information about all the current cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**View**: viewNewInventory

Goal: This view will list information about all the current new cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**View**: viewUsedInventory

Goal: This view will list information about all the current used cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**View**: viewServices

Goal: This view contains a list of all the services offered by the service center of the dealership. This will be helpful to the customer as they will be able to see if the dealership can service his/her car.

**Stored procedure**: GetCarSoldInfo

Parameters: In car\_vin

Goal: This procedure will return the information about a specific car purchase given the car’s VIN. It will be useful for the customer and salesman to get information about a past transacation.

**Stored procedure**: GetCustomerPurchaseHistory

Parameters: In cust\_id

Goal: This procedure will return a customer’s purchase history given his/her customer\_id. It will be useful to the salesman in identifying candidates that may be interested in buying another car.

**Stored procedure**: GetCarServiceHistory

Parameters: In car\_vin

Goal: This procedure will get the service history of a specific car. It will be used by the customer and service managers so that they can see if a car may need servicing in the future.

**Stored procedure**: GetCustomerServiceHistory

Parameters: In cust\_id

Goal: This procedure will return the service history of a customer’s car. This will be helpful to the customer in keeping track of when his car may need servicing.

**Stored procedure**: GetSpecificInventory

Parameters: In car\_vin

Goal: This procedure will provide information about a specific car for sale given a VIN. This will be useful to the salesman and customers because it will allow them to focus on a specific car in the inventory.

**SQL DUMP :**

-- phpMyAdmin SQL Dump

-- version 4.0.10deb1

-- http://www.phpmyadmin.net

--

-- Host: 127.0.0.1

-- Generation Time: Nov 17, 2017 at 12:22 AM

-- Server version: 5.5.57-0ubuntu0.14.04.1

-- PHP Version: 5.5.9-1ubuntu4.22

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

--

-- Database: `Sprint22`

--

DELIMITER $$

--

-- Procedures

--

CREATE PROCEDURE `GetCarServiceHistory`(IN car\_vin int(100))

BEGIN

Select Invoice\_ID as InvoiceNumber, concat(Year(YEAR),' ',cd.Make,' ',cd.Model) as Car, s.service\_name as Service, Price, Service\_Date, Concat(First\_Name,' ',Last\_Name) as Serviceman from Service\_History

Inner Join Employee e on e.Emp\_id=Service\_History.Serviced\_by

Inner Join Service\_List s on s.service\_id=Service\_History.service\_id

Inner Join Car\_Detail cd on cd.Vehicle\_Identification\_Number= Service\_History.Vehicle\_Identification\_Number

Where Service\_History.Vehicle\_Identification\_Number = car\_vin;

END$$

CREATE PROCEDURE `GetCarSoldInfo`(IN `car\_vin` INT(20))

SQL SECURITY INVOKER

BEGIN

SELECT Make,Model,Price\_sold,Concat(e.First\_name, ' ',e.Last\_name) as Salesman,Sold\_on,Concat(c.First\_Name,' ',c.Last\_Name) as Customer FROM Car\_sold

Inner join Car\_Detail on Car\_Detail.Vehicle\_Identification\_Number=Car\_sold.Vehicle\_Identification\_Number

Inner join Employee e on e.Emp\_id=Car\_sold.Sold\_by

Inner join Customer c on c.customer\_id=Car\_sold.Customer\_id

Where Car\_sold.Vehicle\_Identification\_Number = car\_vin;

END$$

CREATE PROCEDURE `GetCustomerPurchaseHistory`(IN `cust\_id` INT(20))

SQL SECURITY INVOKER

BEGIN

SELECT Make,Model,Price\_sold,Concat(e.First\_name, ' ',e.Last\_name) as Salesman,Sold\_on,Concat(c.First\_Name,' ',c.Last\_Name) as Customer FROM Car\_sold

Inner join Car\_Detail on Car\_Detail.Vehicle\_Identification\_Number=Car\_sold.Vehicle\_Identification\_Number

Inner join Employee e on e.Emp\_id=Car\_sold.Sold\_by

Inner join Customer c on c.customer\_id=Car\_sold.Customer\_id

Where Car\_sold.Customer\_id = cust\_id;

END$$

CREATE PROCEDURE `GetCustomerServiceHistory`(IN `cust\_id` INT(10))

SQL SECURITY INVOKER

BEGIN

Select Invoice\_ID as InvoiceNumber, concat(Year(YEAR),' ',cd.Make,' ',cd.Model) as Car, s.service\_name as Service, Price, Service\_Date, Concat(First\_Name,' ',Last\_Name) as Serviceman from Service\_History

Inner Join Employee e on e.Emp\_id=Service\_History.Serviced\_by

Inner Join Service\_List s on s.service\_id=Service\_History.service\_id

Inner Join Car\_Detail cd on cd.Vehicle\_Identification\_Number= Service\_History.Vehicle\_Identification\_Number

Where Customer\_id = cust\_id;

END$$

CREATE PROCEDURE `GetSpecificInventory`(IN `car\_vin` INT(20))

SQL SECURITY INVOKER

BEGIN

SELECT YEAR,Make,Model,Description,Price,Added\_by FROM Inventory

Inner join Car\_Detail on Car\_Detail.Vehicle\_Identification\_Number=Inventory.Vehicle\_Identification\_Number

Where Inventory.Vehicle\_Identification\_Number = car\_vin;

END$$

DELIMITER ;

-- --------------------------------------------------------

--

-- Table structure for table `Car\_Detail`

--

CREATE TABLE IF NOT EXISTS `Car\_Detail` (

`Vehicle\_Identification\_Number` int(20) NOT NULL,

`YEAR` date NOT NULL,

`Make` varchar(200) DEFAULT NULL,

`Model` varchar(20) DEFAULT NULL,

`Mileage` decimal(9,2) DEFAULT NULL,

`Car\_condition` enum('New','Used') NOT NULL,

`Added\_by` int(100) NOT NULL,

PRIMARY KEY (`Vehicle\_Identification\_Number`),

KEY `fk\_car` (`Added\_by`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Car\_Detail`

--

INSERT INTO `Car\_Detail` (`Vehicle\_Identification\_Number`, `YEAR`, `Make`, `Model`, `Mileage`, `Car\_condition`, `Added\_by`) VALUES

(1, '2018-01-01', 'BMW', 'i8', '0.00', 'New', 2),

(2, '2017-01-01', 'Mercedes-Benz', 'AMG-GT', '0.00', 'New', 4),

(3, '2016-01-01', 'Audi', 'RS5', '0.00', 'New', 3),

(4, '2015-01-01', 'Maserati', 'Granturismo', '0.00', 'New', 4),

(5, '2017-01-01', 'Cadillac', 'Escalade', '0.00', 'New', 3),

(6, '2018-01-01', 'BMW', '750i xDrive', '0.00', 'New', 3),

(7, '2016-01-01', 'BMW', '325ci', '25000.00', 'Used', 3),

(8, '2012-01-01', 'Audi', 'A7', '125000.00', 'Used', 3),

(9, '2010-01-01', 'Porsche', '911', '103345.00', 'Used', 2),

(10, '2017-01-01', 'Porsche', '911', '0.00', 'New', 4),

(11, '2013-01-01', 'Hyundai', 'Santa Fe', '45567.00', 'Used', 2),

(12, '2017-01-01', 'Tesla', 'Model X 100D', '0.00', 'New', 2),

(13, '2015-01-01', 'Tesla', 'Model S 70D', '41938.00', 'Used', 4),

(14, '1997-01-01', 'Chevrolet', 'Corvette', '76812.00', 'Used', 4),

(15, '2017-01-01', 'Tesla', 'Model 3', '0.00', 'New', 2),

(16, '1958-01-01', 'Ferrari', 'California', '103345.00', 'Used', 2),

(17, '1923-01-01', 'Duesenberg', 'Model J', '165000.00', 'Used', 4);

-- --------------------------------------------------------

--

-- Table structure for table `Car\_Features`

--

CREATE TABLE IF NOT EXISTS `Car\_Features` (

`VIN` int(20) NOT NULL,

`feature\_id` int(100) NOT NULL,

PRIMARY KEY (`VIN`,`feature\_id`),

KEY `carf\_f2\_fid` (`feature\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Car\_Features`

--

INSERT INTO `Car\_Features` (`VIN`, `feature\_id`) VALUES

(2, 1),

(10, 1),

(2, 2),

(6, 2),

(3, 3),

(4, 4),

(3, 5),

(4, 6),

(4, 7),

(5, 8),

(5, 9),

(5, 10),

(17, 10),

(6, 11),

(16, 11),

(10, 12),

(4, 13),

(16, 13),

(17, 14);

-- --------------------------------------------------------

--

-- Table structure for table `Car\_sold`

--

CREATE TABLE IF NOT EXISTS `Car\_sold` (

`Vehicle\_Identification\_Number` int(100) NOT NULL,

`Price\_sold` int(20) NOT NULL,

`Sold\_on` date NOT NULL,

`Sold\_by` int(100) NOT NULL,

`Customer\_id` int(10) DEFAULT NULL,

PRIMARY KEY (`Vehicle\_Identification\_Number`),

KEY `fk\_cust\_num` (`Customer\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Car\_sold`

--

INSERT INTO `Car\_sold` (`Vehicle\_Identification\_Number`, `Price\_sold`, `Sold\_on`, `Sold\_by`, `Customer\_id`) VALUES

(1, 199999, '2017-11-17', 15, 9),

(7, 16500, '2016-12-19', 14, 1),

(8, 23405, '2017-11-01', 14, 7),

(9, 21304, '2016-02-12', 15, 6),

(11, 9999, '2015-09-18', 15, 2),

(12, 114000, '2017-04-01', 16, 3),

(13, 38952, '2017-10-20', 16, 4),

(14, 12999, '2010-08-11', 17, 5),

(15, 37875, '2017-11-01', 18, 7);

-- --------------------------------------------------------

--

-- Table structure for table `Customer`

--

CREATE TABLE IF NOT EXISTS `Customer` (

`customer\_id` int(100) NOT NULL AUTO\_INCREMENT,

`First\_Name` varchar(50) NOT NULL,

`Middle\_Name` varchar(50) DEFAULT NULL,

`Last\_Name` varchar(50) NOT NULL,

`Email` varchar(50) NOT NULL,

`Addr\_Line1` varchar(100) NOT NULL,

`Addr\_Line2` varchar(100) DEFAULT NULL,

`City` varchar(50) NOT NULL,

`State` varchar(50) NOT NULL,

`ZIP` varchar(20) NOT NULL,

PRIMARY KEY (`customer\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=10 ;

--

-- Dumping data for table `Customer`

--

INSERT INTO `Customer` (`customer\_id`, `First\_Name`, `Middle\_Name`, `Last\_Name`, `Email`, `Addr\_Line1`, `Addr\_Line2`, `City`, `State`, `ZIP`) VALUES

(1, 'Jon', NULL, 'Beason', 'jb@gmail.com', '5017 Darby Chase', NULL, 'Charlotte', 'NC', '28277'),

(2, 'Mike', NULL, 'Adams', 'mike@gmail.com', '12135 Darby Chase', NULL, 'Charlotte', 'NC', '28277'),

(3, 'Sean', NULL, 'Tupp', 'sup@yahoo.com', '501 Petunia Dr', NULL, 'Charlotte', 'NC', '28210'),

(4, 'Jason', NULL, 'Marx', 'jason@yahoo.com', '301 N Tryon St', NULL, 'Charlotte', 'NC', '28110'),

(5, 'Alex', 'Richard', 'Pink', 'ap@yahoo.com', '110 Orchard Dr', NULL, 'Waxhaw', 'NC', '28203'),

(6, 'Michael', NULL, 'Norse', 'michaelnorse@yahoo.com', '5017 Munich Dr', NULL, 'Greenville', 'SC', '28216'),

(7, 'Rich', NULL, 'Door', 'rd@yahoo.com', '10234 Outpost Dr', NULL, 'Matthews', 'NC', '28211'),

(9, 'Sharat', 'C', 'Vyas', 'svyas7@uncc.edu', '10837 Oxford St', NULL, 'Charlotte', 'NC', '28194');

-- --------------------------------------------------------

--

-- Table structure for table `Dealership\_Manager`

--

CREATE TABLE IF NOT EXISTS `Dealership\_Manager` (

`dealership\_manager\_id` int(100) NOT NULL,

`start\_date` date NOT NULL,

PRIMARY KEY (`dealership\_manager\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Dealership\_Manager`

--

INSERT INTO `Dealership\_Manager` (`dealership\_manager\_id`, `start\_date`) VALUES

(1, '2017-11-14');

-- --------------------------------------------------------

--

-- Table structure for table `Employee`

--

CREATE TABLE IF NOT EXISTS `Employee` (

`Emp\_id` int(100) NOT NULL AUTO\_INCREMENT,

`First\_Name` varchar(50) NOT NULL,

`Middle\_Name` varchar(50) DEFAULT NULL,

`Last\_Name` varchar(50) NOT NULL,

`Email` varchar(50) NOT NULL,

`SSN` varchar(20) NOT NULL,

`Salary` int(20) DEFAULT NULL,

`Hours\_per\_week` int(10) DEFAULT NULL,

PRIMARY KEY (`Emp\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=19 ;

--

-- Dumping data for table `Employee`

--

INSERT INTO `Employee` (`Emp\_id`, `First\_Name`, `Middle\_Name`, `Last\_Name`, `Email`, `SSN`, `Salary`, `Hours\_per\_week`) VALUES

(1, 'Michael', 'Duke', 'Scott', 'michaelscott@cars.com', '123456789', 50000, 40),

(2, 'Cameron', 'Jerrell', 'Newton', 'cameronnewton@cars.com', '132465798', 42000, 45),

(3, 'Jack', 'Sean', 'Dorsett', 'jackdorsett@cars.com', '963258741', 28000, 40),

(4, 'Peter', 'Mac', 'Jackson', 'peterjackson@cars.com', '987654321', 35000, 40),

(5, 'David', 'Sean', 'Harris', 'davidharris@cars.com', '1092837456', 31000, 40),

(6, 'Han', 'Andy', 'Zhang', 'hanzhang@cars.com', '1056924738', 46000, 40),

(7, 'Luke', 'Matthew', 'Kuechly', 'lukekuechly@cars.com', '6574839201', 40000, 45),

(8, 'Michael', 'Jim', 'Smith', 'michaelsmith@cars.com', '84769876', 36700, 40),

(9, 'Todd', 'Dob', 'Vez', 'toddvez@cars.com', '11119956', 28960, 35),

(10, 'Ben', 'Sean', 'Stone', 'benstone@cars.com', '54250172', 33000, 40),

(11, 'Sean', 'Nick', 'Toon', 'seantoon@cars.com', '12135012', 35000, 40),

(12, 'Mike', 'Dean', 'Ditka', 'mikeditka@cars.com', '19736874', 38000, 45),

(13, 'Marty', 'McFly', 'Williams', 'martywilliams@cars.com', '19687201', 40000, 45),

(14, 'Thomas', 'Peter', 'Davis', 'thomasdavis@cars.com', '48979875', 40000, 40),

(15, 'Kawann', 'Warwick', 'Short', 'kawannshort@cars.com', '48960986', 38000, 40),

(16, 'Kurt', 'Don', 'Von', 'kurtvon@cars.com', '11112119', 39000, 40),

(17, 'Josh', 'Vick', 'Dunn', 'joshdunn@cars.com', '10220111', 39000, 40),

(18, 'Mike', 'Peter', 'Griffin', 'mikegriffin@cars.com', '20330245', 40000, 40);

-- --------------------------------------------------------

--

-- Table structure for table `Feature\_List`

--

CREATE TABLE IF NOT EXISTS `Feature\_List` (

`feature\_id` int(100) NOT NULL AUTO\_INCREMENT,

`feature\_name` varchar(200) NOT NULL,

PRIMARY KEY (`feature\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=15 ;

--

-- Dumping data for table `Feature\_List`

--

INSERT INTO `Feature\_List` (`feature\_id`, `feature\_name`) VALUES

(1, 'Lane-Keep Assist'),

(2, 'Mercedes-MBRACE'),

(3, 'Audi Quattro Package'),

(4, 'Luxury Seating Package'),

(5, 'Extended Warranty'),

(6, 'Park Assist'),

(7, 'Rear-Seat Entertainment Package'),

(8, 'Wireless Charging Pad'),

(9, 'Gesture Control'),

(10, 'Sport Exhaust Package'),

(11, 'Sports Chrono Package'),

(12, 'Fully Restored'),

(13, 'Rossa Corsa Paint'),

(14, 'Refinished Chrome');

-- --------------------------------------------------------

--

-- Table structure for table `Inventory`

--

CREATE TABLE IF NOT EXISTS `Inventory` (

`Vehicle\_Identification\_Number` int(100) NOT NULL,

`Price` int(20) DEFAULT NULL,

`Description` varchar(200) DEFAULT NULL,

PRIMARY KEY (`Vehicle\_Identification\_Number`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Inventory`

--

INSERT INTO `Inventory` (`Vehicle\_Identification\_Number`, `Price`, `Description`) VALUES

(2, 121500, 'Mercedes-Benz Hypercar'),

(3, 72659, 'Selling fast!'),

(4, 101945, 'Italian Stallion'),

(5, 96576, 'Fine American Luxury SUV'),

(6, 135612, 'Get it before it is gone!'),

(10, 109872, 'One of the finest greatest sports car'),

(16, 475000, 'One of a kind!'),

(17, 987234, 'Fully Restored Collectors Edition');

-- --------------------------------------------------------

--

-- Table structure for table `Inventory\_Manager`

--

CREATE TABLE IF NOT EXISTS `Inventory\_Manager` (

`inventory\_manager\_id` int(100) NOT NULL,

`commission\_percentage` decimal(4,2) DEFAULT '0.00',

`Reports\_to` int(100) NOT NULL,

PRIMARY KEY (`inventory\_manager\_id`),

KEY `Inventory\_Manager\_ibfk\_2` (`Reports\_to`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Inventory\_Manager`

--

INSERT INTO `Inventory\_Manager` (`inventory\_manager\_id`, `commission\_percentage`, `Reports\_to`) VALUES

(2, '6.50', 1),

(3, '9.50', 1),

(4, '7.50', 1);

-- --------------------------------------------------------

--

-- Table structure for table `Salesman`

--

CREATE TABLE IF NOT EXISTS `Salesman` (

`salesman\_id` int(100) NOT NULL,

`commission\_percentage` decimal(4,2) DEFAULT '0.00',

`Reports\_to` int(100) NOT NULL,

PRIMARY KEY (`salesman\_id`),

KEY `Salesman\_ibfk\_2` (`Reports\_to`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Salesman`

--

INSERT INTO `Salesman` (`salesman\_id`, `commission\_percentage`, `Reports\_to`) VALUES

(14, '5.10', 2),

(15, '5.00', 2),

(16, '4.80', 3),

(17, '4.00', 3),

(18, '5.50', 4);

-- --------------------------------------------------------

--

-- Table structure for table `Serviceman`

--

CREATE TABLE IF NOT EXISTS `Serviceman` (

`serviceman\_id` int(100) NOT NULL,

`Reports\_to` int(100) NOT NULL,

PRIMARY KEY (`serviceman\_id`),

KEY `Servicemn\_Manager\_ibfk\_2` (`Reports\_to`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Serviceman`

--

INSERT INTO `Serviceman` (`serviceman\_id`, `Reports\_to`) VALUES

(8, 5),

(9, 5),

(10, 6),

(11, 6),

(12, 7),

(13, 7);

-- --------------------------------------------------------

--

-- Table structure for table `Service\_History`

--

CREATE TABLE IF NOT EXISTS `Service\_History` (

`Vehicle\_Identification\_Number` int(100) NOT NULL,

`Service\_Date` date NOT NULL,

`Invoice\_ID` int(100) NOT NULL AUTO\_INCREMENT,

`Price` decimal(20,0) DEFAULT NULL,

`Serviced\_by` int(100) NOT NULL,

`service\_id` int(100) NOT NULL,

`Customer\_id` int(10) DEFAULT NULL,

PRIMARY KEY (`Invoice\_ID`),

KEY `fk\_sh\_vin` (`Vehicle\_Identification\_Number`),

KEY `fk\_sh\_sb` (`Serviced\_by`),

KEY `Service\_History\_fk\_4` (`service\_id`),

KEY `fk\_cust\_id` (`Customer\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=19 ;

--

-- Dumping data for table `Service\_History`

--

INSERT INTO `Service\_History` (`Vehicle\_Identification\_Number`, `Service\_Date`, `Invoice\_ID`, `Price`, `Serviced\_by`, `service\_id`, `Customer\_id`) VALUES

(7, '2017-11-01', 1, '200', 8, 1, 1),

(7, '2017-02-12', 2, '30', 9, 2, 1),

(7, '2017-06-05', 3, '500', 8, 3, 1),

(7, '2017-10-19', 4, '20', 9, 4, 1),

(9, '2017-07-07', 5, '200', 10, 5, 6),

(9, '2017-04-15', 6, '800', 11, 3, 6),

(11, '2016-05-11', 7, '20', 12, 2, 2),

(11, '2017-08-12', 8, '40', 13, 6, 2),

(12, '2017-10-31', 9, '50', 13, 6, 3),

(13, '2017-10-31', 10, '20', 11, 7, 4),

(14, '2011-01-20', 11, '370', 9, 8, 5),

(14, '2012-02-28', 12, '20', 10, 4, 5),

(14, '2013-10-31', 13, '700', 13, 9, 5),

(14, '2014-07-11', 14, '400', 12, 3, 5),

(14, '2015-03-14', 15, '20', 11, 4, 5),

(14, '2016-07-07', 16, '380', 13, 9, 5),

(15, '2017-11-04', 17, '20', 12, 4, 7),

(8, '2017-11-05', 18, '15', 11, 4, 7);

-- --------------------------------------------------------

--

-- Table structure for table `Service\_List`

--

CREATE TABLE IF NOT EXISTS `Service\_List` (

`service\_id` int(100) NOT NULL AUTO\_INCREMENT,

`service\_name` varchar(200) DEFAULT NULL,

PRIMARY KEY (`service\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=10 ;

--

-- Dumping data for table `Service\_List`

--

INSERT INTO `Service\_List` (`service\_id`, `service\_name`) VALUES

(1, 'Service A'),

(2, 'Oil Change'),

(3, 'Tire Change'),

(4, 'Car Wash'),

(5, 'AC Fluid Replacement'),

(6, 'Wiper Fluid Refill'),

(7, 'Tires Rotated'),

(8, 'Repair Engine Oil Leak'),

(9, 'Engine Servicing');

-- --------------------------------------------------------

--

-- Table structure for table `Service\_Manager`

--

CREATE TABLE IF NOT EXISTS `Service\_Manager` (

`service\_manager\_id` int(100) NOT NULL,

`Reports\_to` int(100) NOT NULL,

PRIMARY KEY (`service\_manager\_id`),

KEY `Service\_Manager\_ibfk\_2` (`Reports\_to`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `Service\_Manager`

--

INSERT INTO `Service\_Manager` (`service\_manager\_id`, `Reports\_to`) VALUES

(5, 1),

(6, 1),

(7, 1);

-- --------------------------------------------------------

--

-- Stand-in structure for view `viewCustList`

--

CREATE TABLE IF NOT EXISTS `viewCustList` (

`Name` varchar(101)

,`Email` varchar(50)

,`Address` varchar(224)

);

-- --------------------------------------------------------

--

-- Stand-in structure for view `viewEmployeeList`

--

CREATE TABLE IF NOT EXISTS `viewEmployeeList` (

`EmployeeID` int(100)

,`Name` varchar(101)

,`Email` varchar(50)

,`SSN` varchar(20)

,`Salary` int(20)

,`Hours\_per\_week` int(10)

,`Role` varchar(18)

);

-- --------------------------------------------------------

--

-- Stand-in structure for view `viewFeatures`

--

CREATE TABLE IF NOT EXISTS `viewFeatures` (

`Feature` varchar(200)

);

-- --------------------------------------------------------

--

-- Stand-in structure for view `viewInventory`

--

CREATE TABLE IF NOT EXISTS `viewInventory` (

`VIN` int(100)

,`Year` int(4)

,`Make` varchar(200)

,`Model` varchar(20)

,`Car\_condition` enum('New','Used')

,`Mileage` decimal(9,2)

,`Description` varchar(200)

,`Features` text

);

-- --------------------------------------------------------

--

-- Stand-in structure for view `viewNewInventory`

--

CREATE TABLE IF NOT EXISTS `viewNewInventory` (

`VIN` int(100)

,`Year` int(4)

,`Make` varchar(200)

,`Model` varchar(20)

,`Description` varchar(200)

,`Price` int(20)

,`Features` text

);

-- --------------------------------------------------------

--

-- Stand-in structure for view `viewServices`

--

CREATE TABLE IF NOT EXISTS `viewServices` (

`Service` varchar(200)

);

-- --------------------------------------------------------

--

-- Stand-in structure for view `viewUsedInventory`

--

CREATE TABLE IF NOT EXISTS `viewUsedInventory` (

`VIN` int(100)

,`Year` int(4)

,`Make` varchar(200)

,`Model` varchar(20)

,`Mileage` decimal(9,2)

,`Description` varchar(200)

,`Price` int(20)

,`Features` text

);

-- --------------------------------------------------------

--

-- Structure for view `viewCustList`

--

DROP TABLE IF EXISTS `viewCustList`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY INVOKER VIEW `viewCustList` AS select concat(`Customer`.`First\_Name`,' ',`Customer`.`Last\_Name`) AS `Name`,`Customer`.`Email` AS `Email`,concat(`Customer`.`Addr\_Line1`,' ',`Customer`.`City`,', ',`Customer`.`State`,' ',`Customer`.`ZIP`) AS `Address` from `Customer`;

-- --------------------------------------------------------

--

-- Structure for view `viewEmployeeList`

--

DROP TABLE IF EXISTS `viewEmployeeList`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY INVOKER VIEW `viewEmployeeList` AS select `e`.`Emp\_id` AS `EmployeeID`,concat(`e`.`First\_Name`,' ',`e`.`Last\_Name`) AS `Name`,`e`.`Email` AS `Email`,`e`.`SSN` AS `SSN`,`e`.`Salary` AS `Salary`,`e`.`Hours\_per\_week` AS `Hours\_per\_week`,(case when `e`.`Emp\_id` in (select `Dealership\_Manager`.`dealership\_manager\_id` from `Dealership\_Manager`) then 'Dealership Manager' when `e`.`Emp\_id` in (select `Inventory\_Manager`.`inventory\_manager\_id` from `Inventory\_Manager`) then 'Inventory Manager' when `e`.`Emp\_id` in (select `Service\_Manager`.`service\_manager\_id` from `Service\_Manager`) then 'Service Manager' when `e`.`Emp\_id` in (select `Serviceman`.`serviceman\_id` from `Serviceman`) then 'Serviceman' when `e`.`Emp\_id` in (select `Salesman`.`salesman\_id` from `Salesman`) then 'Salesman' end) AS `Role` from `Employee` `e`;

-- --------------------------------------------------------

--

-- Structure for view `viewFeatures`

--

DROP TABLE IF EXISTS `viewFeatures`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY INVOKER VIEW `viewFeatures` AS select `Feature\_List`.`feature\_name` AS `Feature` from `Feature\_List`;

-- --------------------------------------------------------

--

-- Structure for view `viewInventory`

--

DROP TABLE IF EXISTS `viewInventory`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY INVOKER VIEW `viewInventory` AS select `Inventory`.`Vehicle\_Identification\_Number` AS `VIN`,year(`Car\_Detail`.`YEAR`) AS `Year`,`Car\_Detail`.`Make` AS `Make`,`Car\_Detail`.`Model` AS `Model`,`Car\_Detail`.`Car\_condition` AS `Car\_condition`,`Car\_Detail`.`Mileage` AS `Mileage`,`Inventory`.`Description` AS `Description`,group\_concat(ifnull(concat\_ws(',',`fl`.`feature\_name`),'-') separator ',') AS `Features` from (((`Inventory` join `Car\_Detail` on((`Inventory`.`Vehicle\_Identification\_Number` = `Car\_Detail`.`Vehicle\_Identification\_Number`))) left join `Car\_Features` `cf` on((`cf`.`VIN` = `Inventory`.`Vehicle\_Identification\_Number`))) left join `Feature\_List` `fl` on((`fl`.`feature\_id` = `cf`.`feature\_id`))) group by `Inventory`.`Vehicle\_Identification\_Number`;

-- --------------------------------------------------------

--

-- Structure for view `viewNewInventory`

--

DROP TABLE IF EXISTS `viewNewInventory`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY INVOKER VIEW `viewNewInventory` AS select `Inventory`.`Vehicle\_Identification\_Number` AS `VIN`,year(`Car\_Detail`.`YEAR`) AS `Year`,`Car\_Detail`.`Make` AS `Make`,`Car\_Detail`.`Model` AS `Model`,`Inventory`.`Description` AS `Description`,`Inventory`.`Price` AS `Price`,group\_concat(ifnull(concat\_ws(',',`fl`.`feature\_name`),'-') separator ',') AS `Features` from (((`Inventory` join `Car\_Detail` on((`Car\_Detail`.`Vehicle\_Identification\_Number` = `Inventory`.`Vehicle\_Identification\_Number`))) left join `Car\_Features` `cf` on((`cf`.`VIN` = `Inventory`.`Vehicle\_Identification\_Number`))) left join `Feature\_List` `fl` on((`fl`.`feature\_id` = `cf`.`feature\_id`))) where (`Car\_Detail`.`Car\_condition` = 'New') group by `Inventory`.`Vehicle\_Identification\_Number`;

-- --------------------------------------------------------

--

-- Structure for view `viewServices`

--

DROP TABLE IF EXISTS `viewServices`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY INVOKER VIEW `viewServices` AS select `Service\_List`.`service\_name` AS `Service` from `Service\_List`;

-- --------------------------------------------------------

--

-- Structure for view `viewUsedInventory`

--

DROP TABLE IF EXISTS `viewUsedInventory`;

CREATE ALGORITHM=UNDEFINED SQL SECURITY INVOKER VIEW `viewUsedInventory` AS select `Inventory`.`Vehicle\_Identification\_Number` AS `VIN`,year(`Car\_Detail`.`YEAR`) AS `Year`,`Car\_Detail`.`Make` AS `Make`,`Car\_Detail`.`Model` AS `Model`,`Car\_Detail`.`Mileage` AS `Mileage`,`Inventory`.`Description` AS `Description`,`Inventory`.`Price` AS `Price`,group\_concat(ifnull(concat\_ws(',',`fl`.`feature\_name`),'-') separator ',') AS `Features` from (((`Inventory` join `Car\_Detail` on((`Car\_Detail`.`Vehicle\_Identification\_Number` = `Inventory`.`Vehicle\_Identification\_Number`))) left join `Car\_Features` `cf` on((`cf`.`VIN` = `Inventory`.`Vehicle\_Identification\_Number`))) left join `Feature\_List` `fl` on((`fl`.`feature\_id` = `cf`.`feature\_id`))) where (`Car\_Detail`.`Car\_condition` = 'Used') group by `Inventory`.`Vehicle\_Identification\_Number`;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `Car\_Detail`

--

ALTER TABLE `Car\_Detail`

ADD CONSTRAINT `Car\_Detail\_ibfk\_1` FOREIGN KEY (`Added\_by`) REFERENCES `Inventory\_Manager` (`inventory\_manager\_id`);

--

-- Constraints for table `Car\_Features`

--

ALTER TABLE `Car\_Features`

ADD CONSTRAINT `carf\_f1\_vin` FOREIGN KEY (`VIN`) REFERENCES `Inventory` (`Vehicle\_Identification\_Number`),

ADD CONSTRAINT `carf\_f2\_fid` FOREIGN KEY (`feature\_id`) REFERENCES `Feature\_List` (`feature\_id`);

--

-- Constraints for table `Car\_sold`

--

ALTER TABLE `Car\_sold`

ADD CONSTRAINT `carsold\_csvin\_vin` FOREIGN KEY (`Vehicle\_Identification\_Number`) REFERENCES `Car\_Detail` (`Vehicle\_Identification\_Number`),

ADD CONSTRAINT `fk\_cust\_num` FOREIGN KEY (`Customer\_id`) REFERENCES `Customer` (`customer\_id`);

--

-- Constraints for table `Dealership\_Manager`

--

ALTER TABLE `Dealership\_Manager`

ADD CONSTRAINT `Dealership\_Manager\_ibfk\_1` FOREIGN KEY (`dealership\_manager\_id`) REFERENCES `Employee` (`Emp\_id`);

--

-- Constraints for table `Inventory`

--

ALTER TABLE `Inventory`

ADD CONSTRAINT `Inventory\_ibfk\_1` FOREIGN KEY (`Vehicle\_Identification\_Number`) REFERENCES `Car\_Detail` (`Vehicle\_Identification\_Number`);

--

-- Constraints for table `Inventory\_Manager`

--

ALTER TABLE `Inventory\_Manager`

ADD CONSTRAINT `Inventory\_Manager\_ibfk\_1` FOREIGN KEY (`inventory\_manager\_id`) REFERENCES `Employee` (`Emp\_id`),

ADD CONSTRAINT `Inventory\_Manager\_ibfk\_2` FOREIGN KEY (`Reports\_to`) REFERENCES `Dealership\_Manager` (`dealership\_manager\_id`);

--

-- Constraints for table `Salesman`

--

ALTER TABLE `Salesman`

ADD CONSTRAINT `Salesman\_ibfk\_1` FOREIGN KEY (`salesman\_id`) REFERENCES `Employee` (`Emp\_id`),

ADD CONSTRAINT `Salesman\_ibfk\_2` FOREIGN KEY (`Reports\_to`) REFERENCES `Inventory\_Manager` (`inventory\_manager\_id`);

--

-- Constraints for table `Serviceman`

--

ALTER TABLE `Serviceman`

ADD CONSTRAINT `serviceman\_ibfk\_1` FOREIGN KEY (`serviceman\_id`) REFERENCES `Employee` (`Emp\_id`),

ADD CONSTRAINT `Servicemn\_Manager\_ibfk\_2` FOREIGN KEY (`Reports\_to`) REFERENCES `Service\_Manager` (`service\_manager\_id`);

--

-- Constraints for table `Service\_History`

--

ALTER TABLE `Service\_History`

ADD CONSTRAINT `fk\_cust\_id` FOREIGN KEY (`Customer\_id`) REFERENCES `Customer` (`customer\_id`),

ADD CONSTRAINT `Service\_History\_fk\_4` FOREIGN KEY (`service\_id`) REFERENCES `Service\_List` (`service\_id`),

ADD CONSTRAINT `Service\_History\_ibfk\_1` FOREIGN KEY (`Vehicle\_Identification\_Number`) REFERENCES `Car\_sold` (`Vehicle\_Identification\_Number`),

ADD CONSTRAINT `Service\_History\_ibfk\_2` FOREIGN KEY (`Serviced\_by`) REFERENCES `Serviceman` (`serviceman\_id`);

--

-- Constraints for table `Service\_Manager`

--

ALTER TABLE `Service\_Manager`

ADD CONSTRAINT `Service\_Manager\_ibfk\_1` FOREIGN KEY (`service\_manager\_id`) REFERENCES `Employee` (`Emp\_id`),

ADD CONSTRAINT `Service\_Manager\_ibfk\_2` FOREIGN KEY (`Reports\_to`) REFERENCES `Dealership\_Manager` (`dealership\_manager\_id`);

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

## ***SPRINT 3***

## ***Refined User Stories***

|  |  |
| --- | --- |
| **Story ID** | **Story Description** |
| US1 | As a Dealership manager, I want to add/remove Inventory Manager |
| US2 | As a Dealership manager, I want to add/remove Service Manager |
| US3 | As a Dealership manager, I want to add/remove Serviceman |
| US4 | As a Dealership manager, I want to add/remove Salesman |
| US5 | As a Guest, I want to register with the dealership and become a Registered Customer. |
| US6 | As an Inventory Manager, I want to add new/used cars to the new car inventory |
| US7 | As an Inventory Manager, I want to remove new/used cars from the inventory |
| US8 | As an Inventory Manager, I want to see all new/used car inventory |
| US9 | As an Inventory Manager, I want to see the information about a specific new/used car |
| US10 | As a Salesman, I want to see all new/used car inventory |
| US11 | As a Salesman, I want to see information about a specific new/used car |
| US12 | As a Customer(guest)/Registered Customer, I want to see brief information about the available new/used cars |
| US13 | As a Service Manager, I want to see history of cars serviced |
| US14 | As a Service Manager, I want to see the service history of a specific car |
| US15 | As a Service Manager, I want to add/update a car service history |
| US16 | As a Serviceman, I want to update a car’s service history |
| US17 | As a Serviceman, I want to view the service history of a car |
| US18 | As a Registered Customer I want to see the history of my car servicing process |
| US19 | As a Dealership manager, I want to see a list of all employees/customers |
| US20 | As a Dealership manager, I want to see information about a specific employee/customer |
| US21 | As a Guest/Registered Customer, I want to browse information about the available services |
| US22 | As a Registered Customer, I want to review my purchase history. |
| US23 | As a Registered Customer, I should be able to buy cars from catalog. |
| US24 | As an Inventory Manager, I want to add my working hours to my daily schedule sheet |
| US25 | As an Inventory Manager, I want to see my working and earning history |
| US26 | As a Salesman, I want to add my working hours to my daily schedule sheet |
| US27 | As a Salesman, I want to see my working and earning history |
| US28 | As a Service Manager, I want to add my working hours to my daily schedule sheet |
| US29 | As a Service Manager, I want to see my working and earning history |
| US30 | As a Serviceman, I want to add my working hours to my daily schedule sheet |
| US31 | As a Serviceman, I want to see my working and earning history |
| US32 | As a Dealership manager, I want to add my working hours to my daily schedule sheet |
| US33 | As a Dealership Manager, I want to see my working and earning history |
| US34 | As a Service Manager, I want to access to Serviceman working history |
| US35 | As a Inventory Manager, I want to access Salesman working history. |
| US36 | As a Dealership Manager, I want to access all employee working history. |

## ***REQUIREMENTS***

|  |  |
| --- | --- |
| US23 | As a Registered Customer, I should be able to buy cars from the dealership’s inventory. |
| US5 | As a Guest, I want to register with the dealership and become a Registered Customer so that I can buy cars and view my car’s service history. |
| US6,7 | As a Inventory manager, I want to add/remove cars from inventory so that the inventory reflects the most up-to-date sales information. |
| US16 | As a Serviceman, I want to update a car’s service history so that it reflects the latest services performed on the customer’s car. |
| US1,2,3,4 | As a Dealership manager, I want to add or remove employees from the system. This will help us keep track of our employee information. |
| US18 | As a Registered Customer I want to see the history of my car servicing process, so I can keep track of my car servicing. |
| US20 | As a Dealership manager, I want to see information about a specific employee/customer. |
| US37 | As a Customer, I want to be able to filter inventory by price to make my my car search easier. |
| US38 | As a Customer, I want to be able to filter inventory by year to make my my car search easier. |
| US39 | As a service manager, I want to be able to add new services to the service list so that my customers can see an updated list of what services are offered. |

## Conceptual Design

Entity: **Car\_Detail**

Attributes:

* Vehicle Identification Number
* Year
* Make
* Model
* Mileage
* Car\_Condition

Entity: **Inventory**

Attributes:

* Price
* Description

Entity: **Cars\_Sold**

Attributes:

* Sold\_on
* price\_sold

Entity: **Service\_History**

Attributes:

* Invoice\_id
* Service\_date
* Description
* Price

Entity: **Employee**

Attributes:

* Name [composite]
  + First\_name
  + Middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Hours\_per\_week
* Status [enum ‘Working’ of ‘Fired’]

Entity: **Dealership\_Manager**

Attributes:

* start\_date

Entity: **Inventory\_Manager**

Attributes:

* Commission\_percentage
* Reports\_to

Entity: **Service\_Manager**

Attributes:

* Start\_date
* Reports\_to

Entity: **Serviceman**

Attributes:

* Reports\_to

Entity: **Salesman**

Attributes:

* Reports\_to
* Commission\_percentage

Entity: **Features\_List**

Attributes:

* Feature\_name

Entity: **Service\_List**

Attributes:

* Service\_name

Entity: **Customer**

Attributes:

* Name [composite]
  + First\_name
  + Middle\_name
  + Last\_name
* Email
* Address

Entity: **Commission\_Earned**

Attributes:

* Commission

Entity:**MakeModel**

Attributes:

* Make\_name
* Model\_name

Relationships:

* Dealership\_Manager **registers** Inventory\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory\_Manager has total participation
* Dealership\_Manager **IS-A** Employee
  + Cardinality : one to one
  + Relationship: Overlapping
  + Participation:
    - Employee has partial participation
    - Dealarship\_Manager has total participation
* Service\_Manager **IS-A** Employee
  + Cardinality : one to one
  + Relationship: Overlapping
  + Participation:
    - Employee has partial participation
    - Service\_Manager has total participation
* Inventory\_Manager **IS-A** Employee
  + Cardinality : one to one
  + Relationship: Overlapping
  + Participation:
    - Employee has partial participation
    - Inventory\_Manager has total participation
* ServiceMan **IS-A** Employee
  + Cardinality : one to one
  + Relationship: Overlapping
  + Participation:
    - Employee has partial participation
    - ServiceMan has total participation
* Salesman **IS-A** Employee
  + Cardinality : one to one
  + Relationship: Overlapping
  + Participation:
    - Employee has partial participation
    - Salesman has total participation
* Dealership\_Manager **removes** Inventory\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory\_Manager has total participation
* Dealership\_Manager **registers** Service\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Service\_Manager has total participation
* Dealership\_Manager **removes** Service\_Manager
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Service\_Manager has total participation
* Inventory\_Manager **adds** new/used Inventory
  + Cardinality : one to many
  + Participation:
    - Inventory\_Manager has partial participation
    - Inventory has total participation
* Inventory\_Manager **removes** new/used Inventory
  + Cardinality : one to many
  + Participation:
    - Inventory\_Manager has partial participation
    - Inventory has total participation
* Inventory\_Manager **views** Inventory
  + Cardinality : one to many
  + Participation:
    - Dealership\_Manager has partial participation
    - Inventory has partial participation
* Serviceman **adds** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Serviceman **updates** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Serviceman **views** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Serviceman **reports to** Service\_Manager
  + Cardinality: many to one
  + Participation:
    - Serviceman has total participation
    - Service\_Manager has partial participation
* ServiceMan **adds** Service\_History
  + Cardinality : one to many
  + Participation:
    - ServiceMan has partial participation
    - Service\_History has total participation
* ServiceMan **updates** Service\_History
  + Cardinality : one to many
  + Participation:
    - ServiceMan has partial participation
    - Service\_History has total participation
* ServiceMan **views** Service\_History
  + Cardinality : one to many
  + Participation:
    - ServiceMan has partial participation
    - Service\_History has total participation
* Service\_Manager **views** Service\_History
  + Cardinality : one to many
  + Participation:
    - Service\_Manager has partial participation
    - Service\_History has total participation
* Car\_Sold **has** Service\_History
  + Cardinality : one to many
  + Participation:
    - Car has total participation
    - Service\_Histoy has total participation
* Inventory **IS-A** Car\_Detail
  + Cardinality : one to one
  + Participation:
    - Inventory has total participation
    - Car\_Detail is Partial participation
* Car\_Sold **IS-A** Car\_Detail
  + Cardinality : one to one
  + Participation:
    - Car\_Detail has partial participation
    - Car\_Sold Detail is Total participation
* Customer **bought** Car\_sold
  + Cardinality: one to many
  + Participation:
    - Customer has partial participation
    - Car\_sold has total participation
* Salesman **reports to** Inventory\_Manager
  + Cardinality: many to one
  + Participation:
    - Salesman has total participation
    - Inventory\_Manager has partial participation
* Salesman **adds to** Commission\_Earned
  + Cardinality: one to one
  + Participation:
    - Salesman has total participation
    - Commission\_Earned has total participation

## Logical Design

Table: **Car\_Detail**

Columns:

* Vehicle Identification Number [primary\_key]
* Year
* Make
* Model
* Mileage
* Car\_Condition
* Added\_by[foreign key; references **inventory\_manager\_id** from **Inventory\_Manager**]

Highest normalization level: 4NF

Justification (if below 4NF): NA

Indexes:

Index Vehicle\_Identification\_Number: clustered

Columns: Vehicle\_Identification\_Number

Justification: Primary Key

Car\_Condition: non clustered

Columns: Car\_condition

Justification: Looking for a new or used car is something common.

Table: **MakeModel**

Columns

* model\_name
* make\_name

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index Inv\_Veh\_Id\_Num: Clustered

Columns: Vehicle\_Identification\_Number

Justification: Used in GetSpecificInventory procedure.

Table: **Inventory**

Columns:

* Vehicle Identification Number [foreign key; references Vehicle Identification Number from **Car\_Detail**] [primary key]
* Description
* Price

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index Inv\_Veh\_Id\_Num: Clustered

Columns: Vehicle\_Identification\_Number

Justification: Used in GetSpecificInventory procedure.

Table: **Car\_Sold**

Columns:

* Vehicle Identification Number [foreign key; references Vehicle Identification Number from **Car\_Detail**] [primary key]
* Sold\_on
* Price\_sold
* Sold\_by[foreign key; references **salesman\_id** from **Salesman**]
* Customer\_id[foreign key; references **customer\_id** from Customer]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index Sold\_Veh\_Id\_Num: Clustered

Columns: Vehicle\_Identification\_Number,

Justification: Used in GetCarSold info.

Index SoldBy\_CustomerId: Non Clustered

Columns: Sold\_by, Cutomer\_id,

Justification: Used in procedure GetCarSold info joins based on Sold\_by and customer\_id. We select car sold for a particular customer and get salesman details for the car sold.

Table: **Car\_Features**

Columns:

* VIN [foreign key; references **VIN** from **Inventory**]
* Feature\_Id [foreign key; references **Feature\_id** from **Features\_List**]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index VIN\_ID: Clustered

Columns: Vehicle\_Identification\_Number,Feature\_id,

Justification: used in ViewUsedInventory, ViewInventory and ViewNewInventory views.

Table: **Features\_List**

Columns:

* Feature\_id
* Feature\_name

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index Feature\_id : Clustered

Columns: Feature\_id

Justification: get features on id, used in ViewUsedInventory, ViewInventory and ViewNewInventory views.

Table: **Services\_List**

Columns

* Service\_id
* Service\_name

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index ServiceID: Clustered

Columns: Service\_id,

Justification: Used in procedure GetCarServiceHistory.

Table:**Service\_History**

Columns:

* Invoice\_id
* Vehicle\_Identification\_Number [foreign key; references Vehicle Identification Number from **Car\_sold**]
* Service\_date
* Service\_id[foreign key; references **Service\_id** from **Services\_List**]
* Price
* Serviced\_by[foreign key; references **serviceman\_id** from **Serviceman**]
* Customer\_id[foreign key; references **customer\_id** from **Customer**]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index ServiceId\_His: Non Clustered

Columns: Service\_id

Justification: Used in procedure GetCarServiceHistory. Joins Service\_List table based on Service\_id.

Index ServiceBy\_His: Non Clustered

Columns: Serviced\_by

Justification: Used in GetCarServiceHistory info joins based on Sold\_by and customer\_id.

Index VIN\_His: Non Clustered

Columns: Vehicle\_Identification\_Number

Justification: Used in GetCarServiceHistory info joins based on Sold\_by and customer\_id.

Table: **Employee**

Columns:

* Emp\_id
* Name [composite]
  + First\_name
  + middle\_name
  + Last\_name
* Email
* SSN
* Salary
* Hours\_per\_week
* Status [enum ‘Working’ of ‘Fired’]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index Emp\_Id: Clustered

Columns: Emp\_id

Justification: fetch employee details based on id.

Table: **Dealership\_Manager**

Columns:

* start\_date
* dealership\_manager\_id [foreign key; references **Emp\_id** from **Employee**]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index DealMan\_Id : Clustered

Columns: dealership\_manager\_id

Justification: used in ViewEmployeeList view.

Table: **Inventory\_Manager**

Columns:

* Commission\_percentage
* Reports\_to[foreign key; references **dealership\_manager\_id** from **Dealership\_Manager**]
* inventory\_manager\_id[foreign key; references **Emp\_id** from **Employee**]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index Invt\_Man : Clustered

Columns: inventory\_manager\_id

Justification: used in ViewEmployeeList view.

Table: **Service\_Manager**

Columns:

* Reports\_to[foreign key; references **dealership\_manager\_id** from **Dealership\_Manager**]
* service\_manager\_id[foreign key; references **Emp\_id** from **Employee**]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index ServMan\_Id: Clustered

Columns: service\_manager\_id

Justification: used in ViewEmployeeList view.

Table: **Serviceman**

Columns:

* serviceman\_id[foreign key; references **Emp\_id** from **Employee**]
* Reports\_to[foreign key; references **service\_manager\_id** from **Service\_Manager**]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index serviceman\_Id : Clustered

Columns: serviceman\_id

Justification: used in ViewEmployeeList view.

Table: **Salesman**

Columns:

* salesman\_id[foreign key; references **Emp\_id** from **Employee**]]
* Commission\_percentage
* Reports\_to[foreign key; references **inventory\_manager\_id** from **Inventory\_manager**]

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index SaleMan\_Id: Clustered

Columns: salesman\_id

Justification: used in ViewEmployeeList view and calculateCommision function.

Table: **Customer**

Columns:

* Customer\_id
* Name [composite]
  + First\_name
  + middle\_name
  + Last\_name
* Email
* Address
  + First\_line
  + Second\_line
  + City
  + State
  + Zip\_code

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index CustId: Clustered

Columns: customer\_id,

Justification:Used in GetCustomerPurchaseHistory and GetCustomerServiceHistory procedure.

Table: **Commision\_Earned**

Columns:

* Salesman\_Id[foreign key; references **salesman\_id** from **Salesman**]
* Commision\_Earned

Highest normalization level: 4NF

Justification (if below 4NF):NA

Indexes:

Index SaleManId: Clustered

Columns: Salesman\_id,

Justification: Used in calculateCommision function.

## ***VIEWS AND STORED PROGRAMS***

**View**: viewCustList

Goal: This view contains a list of all the dealership’s customers as well as some of their personal information. It can be useful for the Dealership manager to see information about his/her customers.

**View**: viewEmployees

Goal: This view contains a list of all the dealership’s employees as well as some of their personal information. It will primarily be used by the Dealership manager to keep track of his current employees.

**View**: viewInventory

Goal: This view will list information about all the current cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**View**: viewNewInventory

Goal: This view will list information about all the current new cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**View**: viewUsedInventory

Goal: This view will list information about all the current used cars that are for sale. It will be used primarily by the salesman and the customers so that they can easily find a car that they may be interested in.

**Stored procedure**: GetCarSoldInfo

Parameters: In car\_vin

Goal: This procedure will return the information about a specific car purchase given the car’s VIN. It will be useful for the customer and salesman to get information about a past transacation.

**Stored procedure**: GetCustomerPurchaseHistory

Parameters: In cust\_id

Goal: This procedure will return a customer’s purchase history given his/her customer\_id. It will be useful to the salesman in identifying candidates that may be interested in buying another car.

**Stored procedure**: GetCarServiceHistory

Parameters: In car\_vin

Goal: This procedure will get the service history of a specific car. It will be used by the customer and service managers so that they can see if a car may need servicing in the future.

**Stored procedure**: GetCustomerServiceHistory

Parameters: In cust\_id

Goal: This procedure will return the service history of a customer’s car. This will be helpful to the customer in keeping track of when his car may need servicing.

**Stored procedure**: GetSpecificInventory

Parameters: In car\_vin

Goal: This procedure will provide information about a specific car for sale given a VIN. This will be useful to the salesman and customers because it will allow them to focus on a specific car in the inventory.

**Stored procedure**: BuyCar

Parameters: In car\_vin, In pricesold, In selldate, In salesman\_id, In cust\_id

Goal: This procedure will allow a salesman to help the customer buy a car. It will move the car information from inventory to car\_sold table, calculate the salesman’s commission and add it to his running total of commission earned so far.

**Stored procedure**: RegisterCustomer

Parameters: In first, In mid, In last, In em, In a1, In a2, In ci, In st, In zi

Goal: This procedure will allow a customer to register with the dealership. Once he is registered he/she can buy a car and get it serviced as well as keep track of purchase and service history.

**Stored procedure**: UpdateServiceHistory

Parameters: In car\_vin, In date, In cost, In servicedby, In serviceid, In cust\_id

Goal: This procedure allows the service manager to add a new servicing record to a car's service history. This means that after the dealership has serviced a customer’s car, the customer can then see the updated service history of the car.

**Stored procedure**: AddFeatureToCar

Parameters: In vin, In feature\_id

Goal: This procedure will allow the inventory manager to add a feature to a car. For example, if he/she adds a new car to the inventory and needs to add the features of that car, then he/she can call this procedure and add the new car’s features.

**Stored procedure**: AddInventory

Parameters: In makename, In modelname, In vin, In carcondition, In miles, In addedby, In time, In cardescription, In carprice

Goal: This procedure will allow the inventory manager to add a new car to the inventory. It will check to see if that make model exists in the MakeModel table, if not it will first add this new model to that table and then add the new car information to the Car\_detail table and Inventory table.

**Stored procedure**: AddInvMan

Parameters: In first, In mid, In last, In sal, In social, In ema, In hpw, In comm

Goal: This procedure will allow the Dealership manager to add a new Inventory Manager to the system.

**Stored procedure**: AddServManager

Parameters: In first, In mid, In last, In sal, In social, In ema, In hpw

Goal: This procedure will allow the Dealership manager to add a new Service Manager to the system.

**Stored procedure**: AddServiceman

Parameters: In first, In mid, In last, In sal, In social, In ema, In hpw

Goal: This procedure will allow the Dealership manager to add a new serviceman to the system.

**Stored procedure**: AddSalesman

Parameters: In first, In mid, In last, In sal, In social, In ema, In hpw, In comm

Goal: This procedure will allow the Dealership manager to add a new salesman to the table.

***Creating Indexes***

1- Table : Car\_Detail

INDEXES

Vehicle\_Identification\_Number

**CREATE** **UNIQUE** **INDEX** Vehicle\_Identification\_Number **ON** Car\_Detail(

Vehicle\_Identification\_Number)

Car\_Condition

**CREATE** **INDEX** Car\_Condition **ON** Car\_Detail(Car\_Condition)

2- Table : Inventory

Inv\_Veh\_Id\_Num

**CREATE UNIQUE INDEX** Inv\_Veh\_Id\_Num **ON** Inventory(Vehicle\_Identification\_Number);

3- Table : Car\_sold

INDEXES

Sold\_Veh\_Id\_Num

**CREATE** **UNIQUE** **INDEX** Vehicle\_Identification\_Number **ON** Car\_sold(

Vehicle\_Identification\_Number);

**CREATE** **INDEX** SoldBy\_CustomerId **ON** Car\_sold

(Sold\_By,Customer\_Id);

4- Table : Car\_Features

INDEXES

VIN\_FID

**CREATE UNIQUE INDEX VIN\_FID ON Car\_Features(VIN,Feature\_id);**

5- Table : Feature\_List

Indexes

Feature\_Id

**CREATE** **UNIQUE** **INDEX** Feature\_Id **ON** Feature\_List(Feature\_id);

6- Table : Service\_List

**CREATE** **UNIQUE** **INDEX** ServiceID **ON** Service\_List(

Service\_id);

7- Table : Service\_History

ServiceId\_His

**CREATE** **INDEX** ServiceId\_His **ON** Service\_History

(Service\_Id);

ServiceBy\_His

**CREATE** **INDEX** ServiceBy\_His **ON** Service\_History

(Serviced\_by);

VIN\_His

**CREATE** **INDEX** VIN\_His **ON** Service\_History

(Vehicle\_Identification\_Number);

8- Table : Employee

Emp\_Id

**CREATE** **UNIQUE** **INDEX** Emp\_Id **ON** Employee(

Emp\_id);

9- Table :

DealMan\_Id

**CREATE** **UNIQUE** **INDEX** DealMan\_Id **ON** Dealership\_Manager(

dealership\_manager\_id);

10- Table : Inventory\_Manager

Invt\_Man

**CREATE** **UNIQUE** **INDEX** Invt\_Man **ON** Inventory\_Manager(

inventory\_manager\_id);

11- Table : Service\_Manager

ServMan\_Id

**CREATE** **UNIQUE** **INDEX** ServMan\_Id **ON** Service\_Manager(

service\_manager\_id);

12- Table : Serviceman

Serviceman\_Id

**CREATE** **UNIQUE** **INDEX** Serviceman\_Id **ON** Serviceman(

serviceman\_id);

13- Table : Salesman

SaleMan\_Id

**CREATE** **UNIQUE** **INDEX** SaleMan\_Id **ON** Salesman(

salesman\_id);

14- Table : Customer

CustId

**CREATE** **UNIQUE** **INDEX** CustId **ON** Customer(

customer\_id);

15- Table : Comission\_Earned

SaleManId

**CREATE** **UNIQUE** **INDEX** SaleManId **ON** Commission\_Earned(

salesman\_id);