Used Auto Dealership Database System

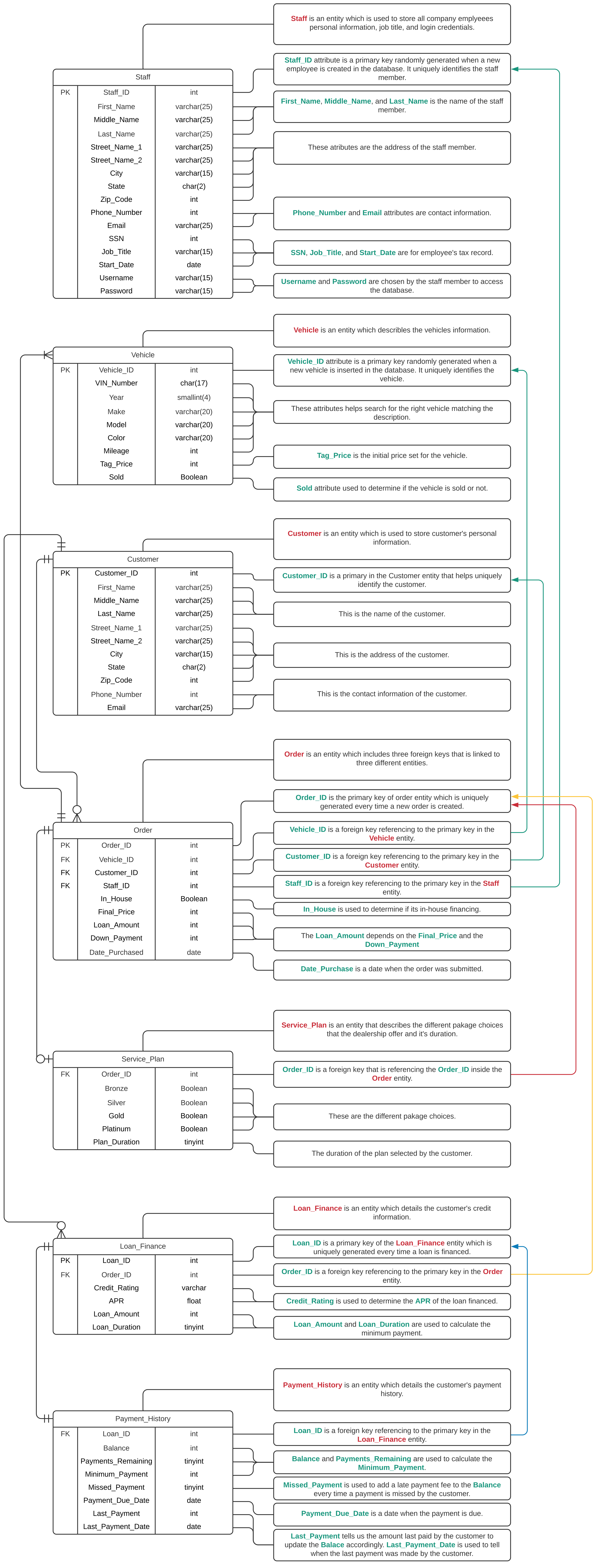
CS 4318 – Dr. Shengli Yuan

PHASE 2

Team Members:

Haider Qazi

Anh Dang



Staff is an entity which is used to store all company emplyeees personal information, job title, and login credentials.

Staff		
PK	Staff_ID	int
	First_Name	varchar(25)
	Middle_Name	varchar(25)
	Last_Name	varchar(25)
	Street_Name_1	varchar(25)
	Street_Name_2	varchar(25)
	City	varchar(15)
	State	char(2)
	Zip_Code	int
	Phone_Number	int
	Email	varchar(25)
	SSN	int
	Job_Title	varchar(15)
	Start_Date	date
	Username	varchar(15)
	Password	varchar(15)

Staff_ID attribute is a primary key randomly generated when a new employee is created in the database. It uniquely identifies the staff member.

First_Name, **Middle_Name**, and **Last_Name** is the name of the staff member.

These atributes are the address of the staff member.

Phone_Number and **Email** attributes are contact information.

SSN, **Job_Title**, and **Start_Date** are for employee's tax record.

Username and **Password** are chosen by the staff member to access the database.

Vehicle is an entity which describles the vehicles information.

Vehicle		
PK	Vehicle_ID	int
	VIN_Number	char(17)
	Year	smallint(4)
	Make	varchar(20)
	Model	varchar(20)
	Color	varchar(20)
	Mileage	int
	Tag_Price	int
	Sold	Boolean

Vehicle_ID attribute is a primary key randomly generated when a new vehicle is inserted in the database. It uniquely identifies the vehicle.

These attributes helps search for the right vehicle matching the description.

Tag_Price is the initial price set for the vehicle.

Sold attribute used to determine if the vehicle is sold or not.

Customer is an entity which is used to store customer's personal information.

Customer		
PK	Customer_ID	int
	First_Name	varchar(25)
	Middle_Name	varchar(25)
	Last_Name	varchar(25)
	Street_Name_1	varchar(25)
	Street_Name_2	varchar(25)
	City	varchar(15)
	State	char(2)
	Zip_Code	int
	Phone_Number	int
	Email	varchar(25)

Customer_ID is a primary in the Customer entity that helps uniquely identify the customer.

This is the name of the customer.

This is the address of the customer.

This is the contact information of the customer.

Order is an entity which includes three foreign keys that is linked to three different entities.

Order		
PK	Order_ID	int
FK	Vehicle_ID	int
FK	Customer_ID	int
FK	Staff_ID	int
	In_House	Boolean
	Final_Price	int
	Loan_Amount	int
	Down_Payment	int
	Date_Purchased	date

Order_ID is the primary key of order entity which is uniquely generated every time a new order is created.

Vehicle_ID is a foreign key referencing to the primary key in the **Vehicle** entity.

Customer_ID is a foreign key referencing to the primary key in the **Customer** entity.

Staff_ID is a foreign key referencing to the primary key in the **Staff** entity.

In_House is used to determine if its in-house financing.

The **Loan_Amount** depends on the **Final_Price** and the **Down_Payment**

Date_Purchase is a date when the order was submitted.

Service_Plan is an entity that describes the different pakage choices that the dealership offer and it's duration.

Service_Plan		
FK	Order_ID	int
	Bronze	Boolean
	Silver	Boolean
	Gold	Boolean
	Platinum	Boolean
	Plan_Duration	tinyint

Order_ID is a foreign key that is referencing the **Order_ID** inside the **Order** entity.

These are the different pakage choices.

The duration of the plan selected by the customer.

Loan_Finance is an entity which details the customer's credit information.

Loan_Finance		
PK	Loan_ID	int
FK	Order_ID	int
	Credit_Rating	varchar
	APR	float
	Loan_Amount	int
	Loan_Duration	tinyint

Loan_ID is a primary key of the **Loan_Finance** entity which is uniquely generated every time a loan is financed.

Order_ID is a foreign key referencing to the primary key in the **Order** entity.

Credit_Rating is used to determine the **APR** of the loan financed.

Loan_Amount and **Loan_Duration** are used to calculate the minimum payment.

Payment_History is an entity which details the customer's payment history.

Payment_History		
FK	Loan_ID	int
	Balance	int
	Payments_Remaining	tinyint
	Minimum_Payment	int
	Missed_Payment	tinyint
	Payment_Due_Date	date
	Last_Payment	int
	Last_Payment_Date	date

Loan_ID is a foreign key referencing to the primary key in the **Loan_Finance** entity.

Balance and **Payments_Remaining** are used to calculate the **Minimum_Payment**.

Missed_Payment is used to add a late payment fee to the **Balance** every time a payment is missed by the customer.

Payment_Due_Date is a date when the payment is due.

Last_Payment tells us the amount last paid by the customer to update the **Balace** accordingly. **Last_Payment_Date** is used to tell when the last payment was made by the customer.

Actors/Users:

1. Owner
2. General Manager
3. Assistant Manager
4. Salesman
5. Assistant

Use Cases:

1. Use case name: INSERT a new Staff Member

   Actor/User: Owner

   Steps:

   a) User clicks on Staff Tab.
   b) User clicks on “New Staff” button.
   c) A new Staff ID is generated and displayed.
   d) Prompt user to enter Name, Address, Contact Information, DOB, Gender, Job Title, Salary, SSN, and Start Date.
   e) All information is displayed. Ask for confirmation.
   f) User clicks on “Confirm” button.
   g) New Staff entry is created, and his/her login credentials are generated.

   ```
   INSERT INTO `new_schema`.`Staff` (`First_Name`,  `Middle_Name`,
   `Last_Name`,`Street_Name_1`, `City`, `State`,  `Zip_Code`, `Phone_Number`, `SSN`,
   `Job_Title`)
   VALUES ('Justin', 'R', 'Smith', '12345 Street Name Dr', 'Houston', 'TX', '77370', '2811236950',
   '435134354', 'Salesman');
   ```

2. Use case name: DELETE a Staff Member

   Actor/User: Owner

   Steps:

   a) User clicks on Staff Tab.
   b) List of Staff Members is displayed.
   c) User selects the Staff Member to delete.
   d) User clicks on “Delete Staff” button.
   e) User clicks on “Confirm” button.
   f) Selected Staff Member is deleted from the Staff Table.

   ```
   DELETE FROM `new_schema`.`Staff`

   WHERE (`Staff_ID` = '103');
   ```

3. Use case name: UPDATE a Staff Member information.

   Actor/User: Owner

   Steps:

   a) Click on Staff Tab.
   b) List of Staff Members is displayed.
   c) User selects the Staff Member to update.
   d) Staff Member information is displayed.
   e) User updates the Staff Member information.
   f) User clicks “Update” button.
   g) User clicks “Confirm” button.
   h) Staff Member information is updated.

```
UPDATE `new_schema`.`Staff`

SET `Middle_Name` = 'T'

WHERE (`Staff_ID` = '104');
```

4. Use case name: QUERY a Staff Member.
   Actor/User: Owner
   Steps:
   a) User clicks on Search Staff Tab
   b) User is prompted with a search form fields.
   c) User enters Staff ID.
   d) The Staff Member is Highlighted.
   e) User Selects the Staff Member and information is displayed.

```
SELECT * FROM new_schema.Staff

WHERE Staff_ID

IN (104);
```

5. Use case name: INSERT a new Vehicle

   Actor/User: General Manager

   Steps:

   a) User clicks on Vehicle Tab.
   b) User clicks on “New Vehicle” button.
   c) A new Vehicle ID is generated and displayed.
   d) Prompt user to enter VIN number, Year, Make, Model, Color, Mileage, and Tag Price.
   e) All information is displayed. Ask for confirmation.

f) User clicks on “Confirm” button.
g) New vehicle entry is created.

```
INSERT INTO `new_schema`.`Vehicle` (`VIN_Number`, `Year`, `Make`, `Model`, `Color`, `Mileage`, `Tag_Price`)

VALUES ('7GJDV75HF75HF75FH', '2019', 'Toyota', 'Camry', 'Silver', '52000', '19000');
```

6. Use case name: DELETE a Vehicle

Actor/User: General Manager

Steps:

a) User clicks on Vehicle Tab.
b) List of Vehicles is displayed.
c) User selects the Vehicle to delete.
d) User clicks on “Delete Vehicle” button.
e) User clicks on “Confirm” button.
f) Selected Vehicle is deleted from the Vehicle Table.

```
DELETE FROM `new_schema`.`Vehicle`

WHERE (`Vehicle_ID` = '4');
```

7. Use case name: UPDATE a Vehicle information.

Actor/User: General Manager

Steps:

a) Click on Vehicle Tab.
b) List of Vehicle is displayed.
c) User selects the Vehicle to update.
d) Vehicle information is displayed.
e) User updates the Vehicle information.
f) User clicks “Update” button.
g) User clicks “Confirm” button.
h) Vehicle information is updated.

```
UPDATE `new_schema`.`Vehicle`

SET `Mileage` = '9000', `Tag_Price` = '17000'

WHERE (`Vehicle_ID` = '4');
```

8. Use case name: QUERY a Vehicle.
   Actor/User: General Manager
   Steps:
   a) User clicks on Search Vehicle Tab
   b) User is prompted with a search form fields.
   c) User enters Vehicle ID.
   d) The Vehicle is Highlighted.
   e) User Selects the Vehicle and information is displayed.

```
SELECT * FROM new_schema.Vehicle

WHERE Vehicle_ID

IN (4);
```

9. Use case name: INSERT a new Customer

   Actor/User: Assistant Manager

   Steps:

   a) User clicks on Customer Tab.
   b) User clicks on “New Customer” button.
   c) A new Customer ID is generated and displayed.
   d) Prompt user to enter Name, Address, and Contact Information.
   e) All information is displayed. Ask for confirmation.
   f) User clicks on “Confirm” button.
   g) New Customer entry is created.

```
INSERT INTO `new_schema`.`Customer` (`First_Name`, `Middle_Name`, `Last_Name`,
`Street_Name_1`, `City`, `State`, `Zip_Code`, `Phone_Number`)
VALUES ('Tom', 'T', 'Cruz', '4355 Hollywood Blvd', 'Los Angeles', 'CA', '34368', '3123456789');
```

10. Use case name: DELETE a Customer

    Actor/User: Assistant Manager

    Steps:

    a) User clicks on Customer Tab.
    b) List of Customers is displayed.
    c) User selects the Customer to delete.
    d) User clicks on “Delete Customer” button.
    e) User clicks on “Confirm” button.
    f) Selected Customer is deleted from the Customer Table.

```
DELETE FROM `new_schema`.`Customer`

WHERE (`Customer_ID` = '3');
```

11. Use case name: UPDATE a Customer information.

    Actor/User: Assistant Manager

    Steps:

    a) Click on Customer Tab.
    b) List of Customer is displayed.
    c) User selects the Customer to update.
    d) Customer information is displayed.
    e) User updates the Customer information.
    f) User clicks “Update” button.
    g) User clicks “Confirm” button.
    h) Customer information is updated.

```
UPDATE `new_schema`.`Customer`

SET `Phone_Number` = '8569785234'

WHERE (`Customer_ID` = '3');
```

12. Use case name: QUERY a Customer.
    Actor/User: Assistant
    Steps:
    a) User clicks on Search Customer Tab
    b) User is prompted with a search form fields.
    c) User enters Customer ID.
    d) The Customer is Highlighted.
    e) User Selects the Customer and information is displayed.

```
SELECT * FROM new_schema.Customer

WHERE Customer_ID

IN (3);
```

13. Use case name: INSERT a new Order

    Actor/User: Salesman

    Steps:

a) User clicks on Order Tab.
b) User clicks on “New Order” button.
c) A new Order ID is generated and displayed.
d) Prompt user to enter Vehicle ID, Customer ID, and Financial Information.
e) All information is displayed. Ask for confirmation.
f) User clicks on “Confirm” button.
g) New Order entry is created.

```
INSERT INTO `new_schema`.`Order` (`Vehicle_ID`, `Customer_ID`, `Staff_ID`, `Down_Payment`,
`Final_Price`, `In_House`, `Loan_Amount`)

VALUES ('104', '100', '0004', '9000', '15000', '1', '6000');
```

14. Use case name: DELETE an Order

Actor/User: Salesman

Steps:

a) User clicks on Order Tab.
b) List of Order is displayed.
c) User selects the Order to delete.
d) User clicks on “Delete Order” button.
e) User clicks on “Confirm” button.
f) Selected Order is deleted from the Order Table.

```
DELETE FROM `new_schema`.`Order`

WHERE (`Order_ID` = '5');
```

15. Use case name: UPDATE an Order information.

Actor/User: Salesman

Steps:

a) Click on Order Tab.
b) List of Order is displayed.
c) User selects the Order to update.
d) Order information is displayed.
e) User updates the Order information.
f) User clicks “Update” button.
g) User clicks “Confirm” button.
h) Order information is updated.

```
UPDATE `new_schema`.`Order`

SET `Down_Payment` = '2000'

WHERE (`Order_ID` = '3');
```

16. Use case name: QUERY an Order.

    Actor/User: Assistant

    Steps:

    a) User clicks on Search Order Tab
    b) User is prompted with a search form fields.
    c) User enters Order ID.
    d) The Order is Highlighted.
    e) User Selects the Order and information is displayed.

```
SELECT * FROM new_schema.Order

WHERE Order_ID

IN (3);
```

17. Use case name: INSERT a new Service Plan

    Actor/User: Assistant Manager

    Steps:

    a) User clicks on Order Tab.
    b) User clicks “Search Order” button.
    c) User input Order ID.
    d) Order information is displayed.
    e) User clicks on add service plan.
    f) User is prompted to select the Bronze, Silver, Gold, Platinum and its duration.
    g) User clicks on “Confirm” button.
    h) New Service Plan entry is created.

```
INSERT INTO `new_schema`.`Service_Plan` (`Order_ID1`, `Bronze`, `Silver`, `Gold`, `Platinum`,
`Plan_Duration`,)

VALUES ('0005', '1', '0', '0', '0', '12');
```

18. Use case name: DELETE a Service Plan

    Actor/User: Assistant Manager

Steps:

a) User clicks on Order Tab.

b) User clicks “Search Order” button.

c) User input Order ID.

d) Order information is displayed.

e) User clicks on Delete service plan.

f) User clicks on “Confirm” button.

g) Service Plan is Deleted.

```
DELETE FROM `new_schema`.`Service_Plan`

WHERE (`Order_ID1` = '0005');
```

19. Use case name: UPDATE Service Plan duration.

Actor/User: Assistant Manager

Steps:

a) User clicks on Order Tab.

b) User clicks “Search Order” button.

c) User input Order ID.

d) Order information is displayed.

e) User clicks on Update service plan.

f) User change service plan duration.

g) User clicks on “Confirm” button.

h) Service Plan is Updated.

```
UPDATE `new_schema`.`Service_Plan`

SET `Plan_Duration` = '24'

WHERE (`Order_ID1` = '0006');
```

20. Use case name: QUERY a Service Plan.

Actor/User: Assistant

Steps:

a) User clicks on Search Order Tab.

b) User is prompted with a search form fields.

c) User enters Order ID.

d) User clicks on “Service Plan” button.

e) Service Plan information is displayed.

```
SELECT * FROM new_schema.Service_Plan

WHERE Order_ID1

IN (0007);
```

21. Use case name: INSERT a new Loan Finance.

    Actor/User: General Manager

    Steps:

    a) User clicks on Loan Finance Tab.
    b) User clicks on “New Loan” button.
    c) New Loan ID is generated and displayed.
    d) Prompt user to enter Order ID, Credit Rating, APR, Loan Amount, and Loan Duration.
    e) All information is displayed. Ask user for confirmation.
    f) User clicks on “Confirm” button.
    g) New Loan Finance entry is created.

```
INSERT INTO `new_schema`.`Loan_Finance` (`Loan_ID`, `Order_ID`, `Loan_Amount`, `Loan_Duration`, `Credit_Rating`, `APR`, `Balance`)

VALUES ('0005', '0007', '9000', '12', 'Good', '12', '15000');
```

22. Use case name: DELETE a Loan Finance

    Actor/User: General Manager

    Steps:

    a) User clicks on Loan Finance Tab.
    b) List of Loans is displayed.
    c) User selects the Loan to delete.
    d) User clicks on “Delete Loan” button.
    e) User clicks on “Confirm” button.
    f) Selected Loan entry is deleted from the Loan Finance Table.

```
DELETE FROM `new_schema`.`Loan_Finance`

WHERE (`Loan_ID` = '00020');
```

23. Use case name: UPDATE Loan Finance.

    Actor/User: General Manager

Steps:

a) User clicks on Loan Finance Tab.
b) List of Loans is displayed.
c) User selects the Loan to update.
d) Loan Finance information is displayed.
e) User updates the Loan Finance information.
f) User clicks “Update” button.
g) User clicks “Confirm” button.
h) Loan information is updated.

```
UPDATE `new_schema`.`Loan_Finance`

SET `Loan_Duration` = '16'

WHERE (`Loan_ID` = '00067');
```

24. Use case name: QUERY a Loan Finance.
    Actor/User: Assistant
    Steps:
    a) User clicks on Loan Finance Tab.
    b) User clicks on “Search Loan” button.
    c) User is prompted with a search form fields.
    d) User enters Loan ID.
    e) The Loan is Highlighted.
    f) User Selects the Loan and information is displayed.

```
SELECT * FROM new_schema.Loan_Finance

WHERE Loan_ID

IN (00076);
```

25. Use case name: INSERT Last Payment to Payment History Table.

    Actor/User: Assistant Manager

    Steps:

    a) User clicks on Loan Finance Tab.
    b) User clicks on “Search Loan” button.
    c) User is prompted with a search form field.
    d) User enters the Loan ID.
    e) All information is displayed.
    f) User clicks on “Payment History” button.

g) Payment History information is displayed.
h) User clicks on Last Payment entry.
i) User updates the Last Payment information value.
j) Ask for confirmation.
k) User clicks on “Confirm” button.
l) New last payment entry is created.

```
INSERT INTO `new_schema`.`Payment_History (`’Last_Payment’)

VALUES (‘315’);
```

26. Use case name: DELETE Balance from Payment History Table.

    Actor/User: Assistant Manager

    Steps:

    a) User clicks on Loan Finance Tab.
    b) User clicks on “Search Loan” button.
    c) User is prompted with a search form field.
    d) User enters the Loan ID.
    e) All information is displayed.
    f) User clicks on “Payment History” button.
    g) Payment History information is displayed.
    h) User clicks on “Delete Balance” button.
    i) Balance attribute value turns to NULL.

```
DELETE FROM `new_schema`.`Payment_History`

WHERE (`Loan_ID` = '00027');
```

27. Use case name: UPDATE Payment History.

    Actor/User: Assistant Manager

    Steps:

    a) User clicks on Loan Finance Tab.
    b) User clicks on “Search Loan” button.
    c) User is prompted with a search form field.
    d) User enters the Loan ID.
    e) All information is displayed.
    f) User clicks on “Payment History” button.
    g) Payment History information is displayed.
    h) User clicks on Last Payment entry.

i) User deletes the Last Payment information value.
j) Ask for confirmation.
k) User clicks on “Confirm” button.
l) Last Payment information is updated in the Payment History Table.

```
UPDATE `new_schema`.`Payment_History`
SET `Last_Payment` = '400'
WHERE (`Loan_ID` = '00050');
```

28. Use case name: QUERY a Minimum Payment from Payment History Table.
    Actor/User: Assistant
    Steps:
    a) User clicks on Loan Finance Tab.
    b) User clicks on “Search Loan” button.
    c) User is prompted with a search form field.
    d) User enters the Loan ID.
    e) All information is displayed.

```
SELECT * FROM new_schema.Payment_History
WHERE Loan_ID
IN (00085);
```

29. Use case name: UPDATE BOTH Loan Finance and Payment History with ONE entry.

    Actor/User: Salesman

    Steps:

    a) User clicks on Loan Finance Tab.
    b) User clicks on “Search Loan” button.
    c) User enters the Loan ID and Selects the Loan.
    d) Loan Finance information is displayed.
    e) User changes the Loan Duration attribute value.
    f) User clicks on “Update” button.
    g) Loan Duration entry is Updated in the Loan Finance Table.
    h) The Payments Remaining and Minimum Payment is also updated under Payment History.

```
UPDATE `new_schema`.`Loan_Finance`
```

```
SET `Loan_Duration` = '16'

WHERE (`Loan_ID` = '00060');
```

30. Use case name: UPDATE BOTH Service Plan and Payment History Balance with ONE ENTRY.

    Actor/User: Salesman

    Steps:

    a) User clicks on Order Tab.
    b) User clicks on “Search Order” button.
    c) User enters Order ID.
    d) Order is highlighted and user selects the order.
    e) Order information is displayed.
    f) User clicks on “Service Plan” button.
    g) User updates the service package value.
    h) User clicks “Confirm”.
    i) Balance under Payment History entity is updated also.

```
UPDATE `new_schema`.`Service_Plan`

SET `Silver` = '1’

WHERE (`Order_ID’ = '0006');
```

31. Use case name: UPDATE BOTH Order Down Payment and Payment History Balance with one ENTRY.

    Actor/User: Salesman

    Steps:

    a) User clicks on Order Tab.
    b) User clicks “Search Order” button.
    c) User enters Order ID.
    d) Order is highlighted and user selects the order.
    e) Order information is displayed.
    f) User update Down Payment attribute value.
    g) User clicks “Update” button.
    h) User clicks “Confirm” button.
    i) Balance attribute under Payment History is also updated.

```
UPDATE `new_schema`.`Order`
```

```
SET `Down_Payment` = '12000'

WHERE (`Order_ID` = '0007');
```

32. Use case name: CREATE Order entry and Sold attribute under Vehicle is UPDATED.

    Actor/User: Salesman

    Steps:

    a) User clicks on Order Tab.
    b) User clicks on “New Order” button.
    c) User enters Vehicle ID and order information.
    d) New Order entry is created.
    e) Sold attribute under Vehicle is updated to True.

```
INSERT INTO `new_schema`.`Order’ (‘Vehicle_ID’)

VALUES (‘007’);
```

33. Use case name: QUERY Vehicle information inside Order.

    Actor/User: Assistant

    Steps:

    a) User clicks on Order Tab.
    b) User clicks “Search Order” button.
    c) User enters Order ID.
    d) Order is highlighted and user selects the order.
    e) Vehicle information is displayed inside Order.

```
SELECT * FROM new_schema.Order

WHERE Order_ID

IN (00087);
```

34. Use case name: QUERY Staff information inside Order.

    Actor/User: Assistant

    Steps:

    a) User clicks on Order Tab.
    b) User clicks “Search Order” button.

c) User enters Order ID.

d) Order is highlighted and user selects the order.

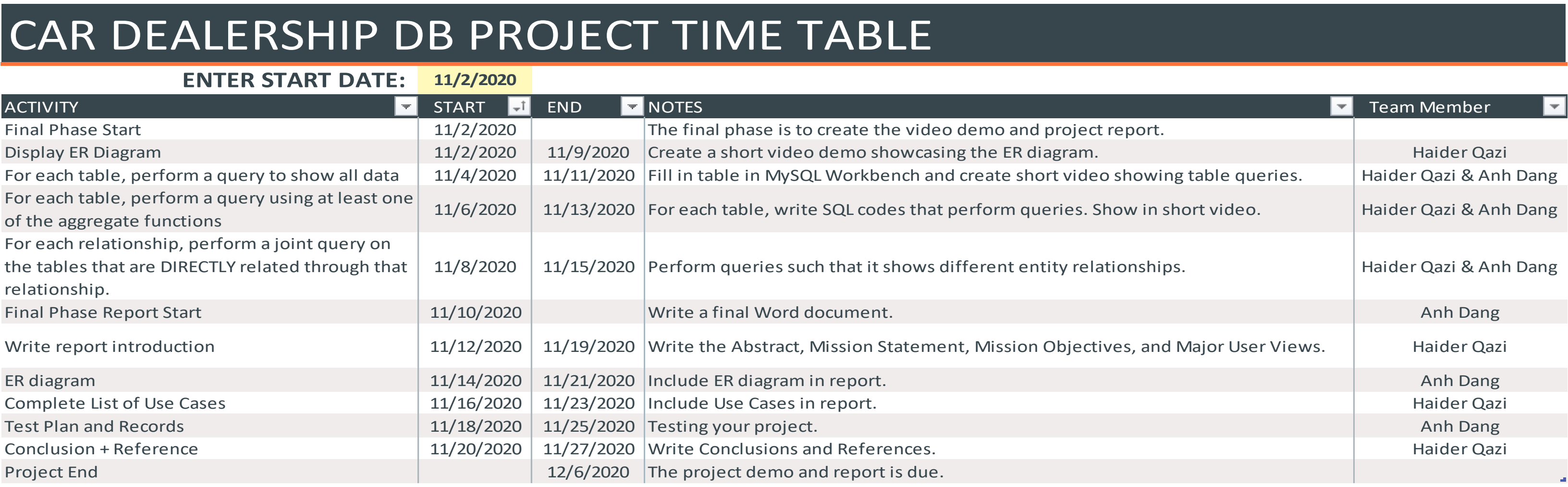
e) Staff information is displayed inside Order.

```
SELECT * FROM new_schema.Order

WHERE Order_ID

IN (0008);
```

Project Timetable



CAR DEALERSHIP DB PROJECT TIME TABLE

ENTER START DATE: **11/2/2020**

ACTIVITY	START	END	NOTES	Team Member
Final Phase Start	11/2/2020		The final phase is to create the video demo and project report.	
Display ER Diagram	11/2/2020	11/9/2020	Create a short video demo showcasing the ER diagram.	Haider Qazi
For each table, perform a query to show all data	11/4/2020	11/11/2020	Fill in table in MySQL Workbench and create short video showing table queries.	Haider Qazi & Anh Dang
For each table, perform a query using at least one of the aggregate functions	11/6/2020	11/13/2020	For each table, write SQL codes that perform queries. Show in short video.	Haider Qazi & Anh Dang
For each relationship, perform a joint query on the tables that are DIRECTLY related through that relationship.	11/8/2020	11/15/2020	Perform queries such that it shows different entity relationships.	Haider Qazi & Anh Dang
Final Phase Report Start	11/10/2020		Write a final Word document.	Anh Dang
Write report introduction	11/12/2020	11/19/2020	Write the Abstract, Mission Statement, Mission Objectives, and Major User Views.	Haider Qazi
ER diagram	11/14/2020	11/21/2020	Include ER diagram in report.	Anh Dang
Complete List of Use Cases	11/16/2020	11/23/2020	Include Use Cases in report.	Haider Qazi
Test Plan and Records	11/18/2020	11/25/2020	Testing your project.	Anh Dang
Conclusion + Reference	11/20/2020	11/27/2020	Write Conclusions and References.	Haider Qazi
Project End		12/6/2020	The project demo and report is due.	