CHAPTER 1

INTRODUCTION

1.1 PREAMBLE

Vehicle Sales Management System is software which is helpful for customer, who wants to know the warranty period of a vehicle. Vehicle Sales Management System is a content management system written for any operating systems which focused in the area of adding, editing and deleting the manufacturer details, model, and price. In this software an employee can register as a user and he can manage the customer details, vehicle details.

In this system, an admin can add a new vehicle details to the database. Whenever the admin wants to modify these details he can update new values through this software. The main advantage is that this module is helpful for the customers to get details of the vehicles such as warranty period.

Vehicle Sales Management System is developed and customized for employee and customer. Its modules support most type of vehicles.

Vehicle Sales Management System uses PHP as the front end and MYSQL as the back end.

1.2 PROBLEM STATEMENT

Current system does not have a feature of customer login. Customer has to always ask the employee for the details of his/her vehicle details such as warranty period, vehicle price, vehicle insurance details.

Disadvantages:

The following are the disadvantages of current system

1. It is difficult for customer to know vehicle related details.
2. It is difficult to maintain the system by the employees.
3. No proper details of the vehicle.
4. It is tedious for adding and updating details.

1.3 Proposed Solution

Proposed system has the feature of customer login using which a customer can know every specific detail related to a vehicle. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the employees to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance.

Advantages:

The following are the advantages of proposed system

1. Easy to process requests
2. Can generate required reports easily
3. Easy to manage historical data in a secure manner
4. Centralized database helps in avoiding conflict

Functions of ADMIN:

This module provides administrator related functionalities. Administrator can view the registered customers and their details, the current vehicle available . In this module admin can upload the information about the employee, customer, vehicle such as manufacturer details, model details, customer details. Admin will give the responses to client based on their requirements always fulfilling their needs.

This module also requires the admin logged in to keep all the customer information confidential and not disclose it to any other third party , not compromising the data to unauthorized access.

Function of USER:

This module is about users of this portal. By using this module user, the customer can view the customer details. User must be registered with the system. By using this login id user will log on to this portal and do all selling related works of the car to the customer. Customer registration details must be kept confidential after that he/she will give their personal details through the website. Customer therefore using its login id can view their details and vehicle details such as warranty period and insurance related details.

CHAPTER 2

ANALYSIS AND SYSTEM REQUIREMENTS

2.1 LITERATURE SURVEY

[1] A vehicle sales management system in improving the relational use of customer requirements.

Authors: T.Shang, X.Ren, R.Ma

This report first reviews the key constraints on vehicle utilization, before examining opportunities to optimize transport operations. Within the remit of this report, three key areas for improved efficiencies within the industry are identified: logistical efficiency, vehicle utilization and driver training and behavior.

Environmental issues will increasingly influence the way employees do their jobs.Currently, there is a growing field of governmental literature offering advice and guidance.

The scope of this review is limited by the availability of literature and time. As a broad study, it does not present the full range of literature on the state of vehicle management, but attempts to give an overview of the main concerns and areas for improvement.

This report highlights limitations faced by employees in attempting to operate vehicles efficiently and draws together environmentally-related literature that offers guidance to them.

[2] Practice of administration management: A review of Literature

Authors: Ayad k.ali

As a literature review it aims to synthesize previous work rather than develop new perspectives. It should provide a foundation for future research in this field.

A vehicle sales management system is a bundled management information system created specifically for automotive industry car dealerships or large equipment manufacturers. These systems often contain software that caters to the needs of finance, sales parts, and inventory and administration components of running the dealership. A typical installation usually includes a central server which stores all data, allowing multi-user access for 50 or more client computers.

2.2 SYSTEM REQUIREMENT SPECIFICATION

- Should describe functional and non-functional requirements so that they are understandable by system users who don’t have detailed technical knowledge.
- User requirements are defined using natural language, tables and diagrams.

Software Requirement Specification

Programming Languages : PHP, HTML, CSS, JS, and MYSQL.

Web Technologies : Html,CSS,PHP.

Database : MySQL

Server : WAMPserver 3.0

Hardware Requirements Specification

Processor : Intel core 2 duo

Hard Disk : 50 GB

Ram : 1 GB

CHAPTER 3

SYSTEM DESIGN AND ANALYSIS

3.1 PRELIMNARY DESIGN

3.1.1 Entity – Relationship Diagram:

This depicts relationship between data objects. The attribute of each data objects noted in the entity- relationship diagram can be described using a data object description. Data flow diagram serves two purposes:

1. To provide an indication of how data are transformed as they move through
the system.

2. To depict the functions that transformation the data flow.

The basic components of entity relationship diagram/model are listed and defined below:

Data Objects: A data object is a representation of almost any composite information that must be understood by the software. By composite information, we mean something that has a number of different properties or attributes. A data object encapsulates data only there is no reference within a data object to
Operations that act on the data.

Attributes: Attributes define the properties of a data object and take on one of three different characteristics

They can be used to:

1. Name an instance of data object.

2. Describe the instance.

3. Make reference to another instance in other table.

Relationships: Data objects are connected to one another in a variety of different ways. We can define a set of object relationship pairs that define the relevant relationships.

Cardinality:

The data model must be capable of representing the number of occurrences of objects in a given relationship. The cardinality of an object relationship pair is

1. One-T0-One (1:1): An occurrence of object ‘A’ can relate to one and only one occurrence of object ‘B’ and vice versa.

2. One-To-Many (1:N): One occurrence of object ‘A’ can relate to one or may occurrences of object ‘B’ but an occurrence of object ‘B’ can relate to only one occurrence of object ‘A’**.**

3. Many-To-Many (M: N): An occurrences of ‘B’ and an occurrence of ‘B’ can relate to one or many occurrence of ‘A’.

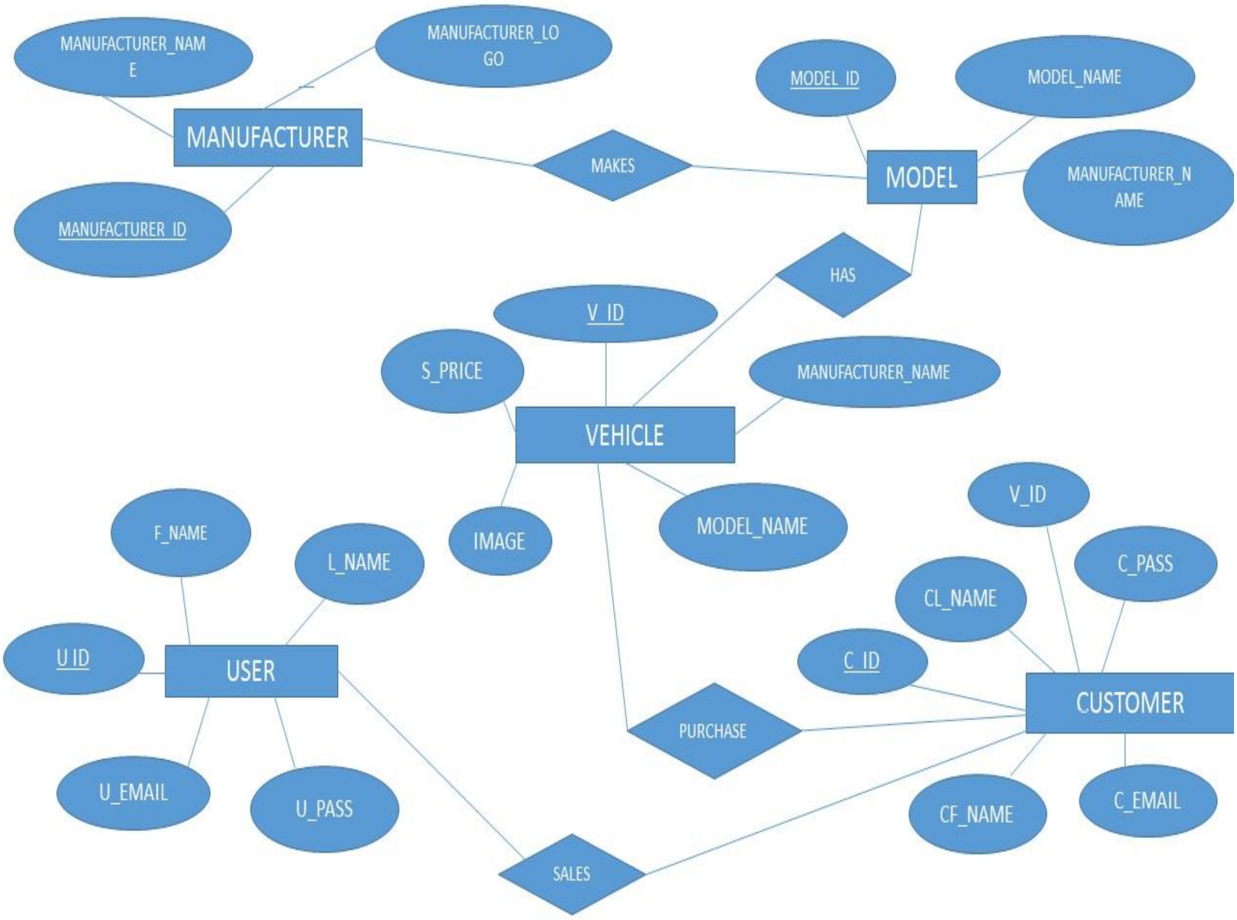

MANUFACTURER_NAME
MANUFACTURER_LOGO
MANUFACTURER
MANUFACTURER_ID
MAKES
MODEL_ID
MODEL_NAME
MODEL
MANUFACTURER_NAME
HAS
V_ID
S_PRICE
MANUFACTURER_NAME
VEHICLE
IMAGE
MODEL_NAME
F_NAME
L_NAME
U ID
USER
U_EMAIL
U_PASS
V_ID
C_PASS
CL_NAME
C_ID
CUSTOMER
PURCHASE
CF_NAME
C_EMAIL
SALES

Fig: 3.1.1.1

3.1.2 SCHEMA DIAGRAM

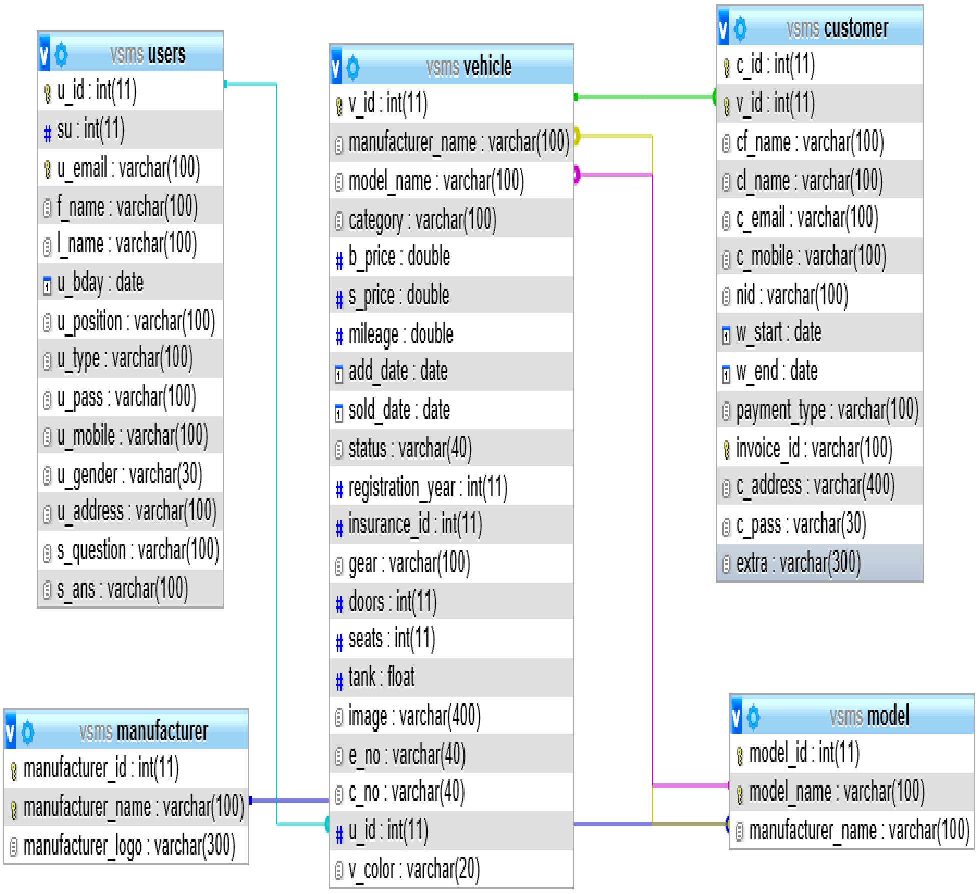


Fig: 3.1.2.1

3.1.3 Flowchart

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

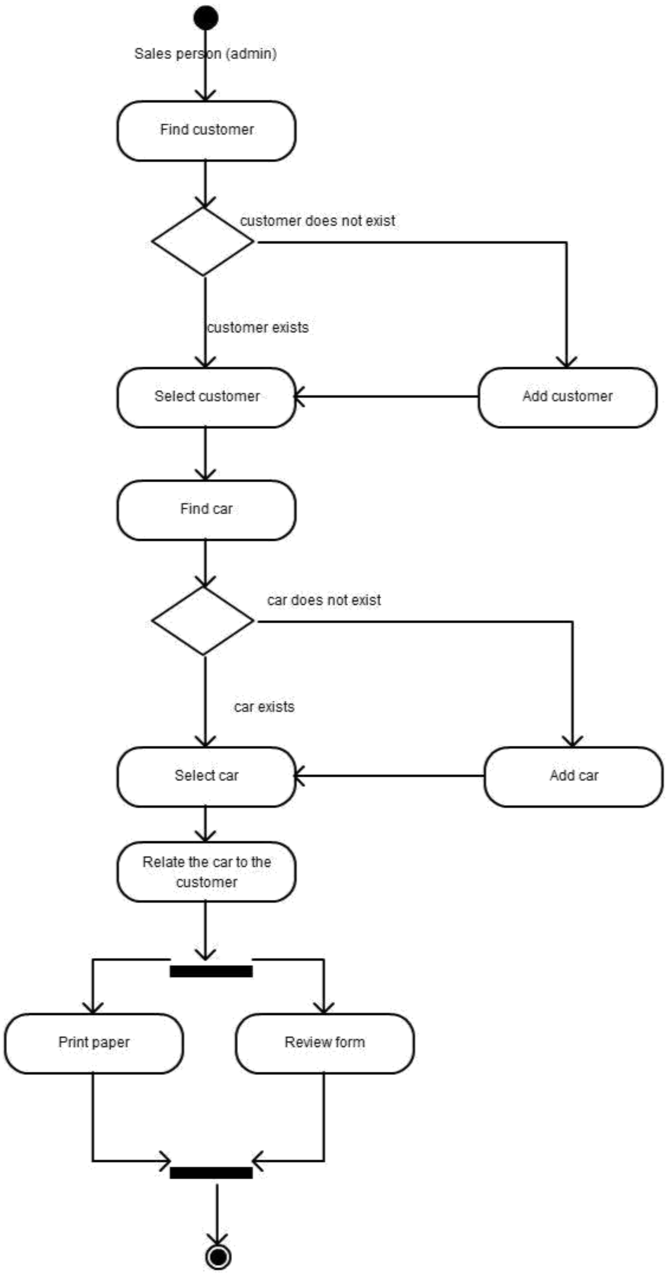


Fig: 3.1.3.1

3.1.4 DATABASE DESCRIPTION

Customer

This table consists of all details related to the customer and his vehicle.

Column	Type	Null	Default	Comments
c_id	int(11)	No		
v_id	int(11)	No		
cf_name	varchar(100)	No		
cl_name	varchar(100)	No		
c_email	varchar(100)	No		
c_mobile	varchar(100)	No		
nid	varchar(100)	Yes	*NULL*	
w_start	Date	No		
w_end	Date	No		
payment_type	varchar(100)	No		
invoice_id	varchar(100)	Yes	*NULL*	
c_address	varchar(400)	Yes	*NULL*	
c_pass	varchar(30)	No		
extra	varchar(300)	Yes	*NULL*	

Manufacture

This table consists details of the manufacture who made the vehicle.

Column	Type	Null	Default	Comments
manufacturer_id	int(11)	No		
manufacturer_name	varchar(100)	No		
manufacturer_logo	varchar(300)	Yes	*NULL*	

Model

This table consists the details related to the model of vehicle.

Column	Type	Null	Default	Comments
model_id	int(11)	No		
model_name	varchar(100)	No		
manufacturer_name	varchar(100)	No		

Users

This table consists the details of all the employee in the company.

Column	Type	Null	Default	Comments
u_id	int(11)	No		
su	int(11)	Yes	0	
u_email	varchar(100)	No		
f_name	varchar(100)	No		
l_name	varchar(100)	No		
u_bday	Date	No		
u_position	varchar(100)	No		
u_type	varchar(100)	No		
u_pass	varchar(100)	No		
u_mobile	varchar(100)	No		
u_gender	varchar(30)	No		
u_address	varchar(100)	No		
s_question	varchar(100)	Yes	*NULL*	
s_ans	varchar(100)	Yes	*NULL*	

Vehicle

This table consists the various details and features of the vehicle.

Column	Type	Null	Default	Comments
v_id	int(11)	No		
manufacturer_name	varchar(100)	No		
model_name	varchar(100)	No		
category	varchar(100)	No		
b_price	double	No		
s_price	double	Yes	*NULL*	
mileage	double	No		
add_date	date	No		
sold_date	date	Yes	*NULL*	
status	varchar(40)	No		
registration_year	int(11)	No		
insurance_id	int(11)	Yes	*NULL*	
gear	varchar(100)	No		
doors	int(11)	No		
seats	int(11)	No		
tank	float	No		
image	varchar(400)	Yes	*NULL*	
e_no	varchar(40)	No		
c_no	varchar(40)	No		
u_id	int(11)	Yes	*NULL*	
v_color	varchar(20)	Yes	*NULL*	

3.2 Normalization

Normalizing of data can be considered a process of analyzing the given relation schemas on their fd's and primary keys to achieve the desirable properties of minimizing redundancy, insertion, deletion and update anomalies.

3.2.1 First Normal Form (1nf)-

1nf disallows having a set of values as an attribute value for a single tuples.

The only attribute values permitted by 1nf are single atomic values.

customer

C_id	C_fname	C_lname	C_mobile
1	rahul	gupta	123456789,987654
2	ravi	sharma	4545667878

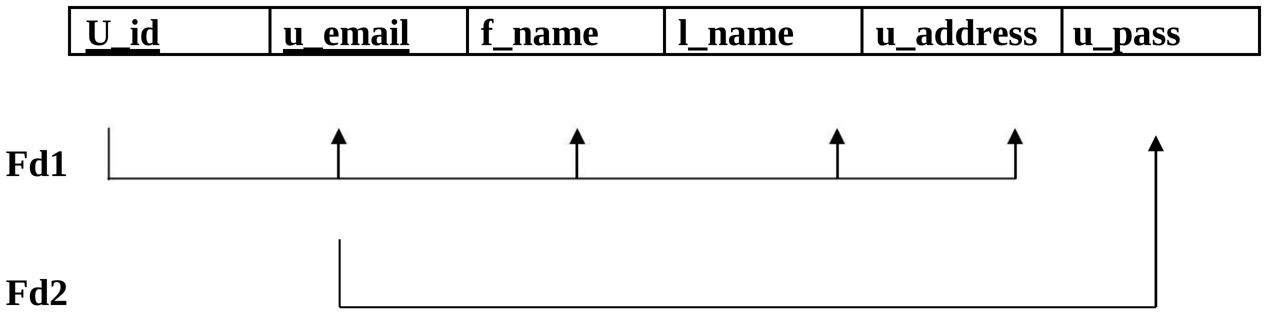
1 nf form

C_id	C_fname	C_lname	C_mobile
1	rahul	gupta	123456789
1	rahul	gupta	987654
2	ravi	sharma	4545667878

3.2.2 Second Normal Form (2nf)-

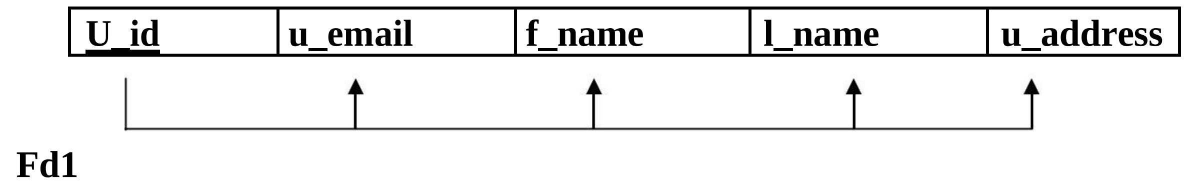
A relation schema R is in 2nf if every non prime attribute A in R is fully functionally dependent on the primary key of R.

User

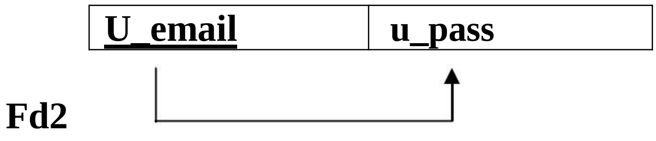


U_id	u_email	f_name	l_name	u_address	u_pass

2nf form



U_id	u_email	f_name	l_name	u_address

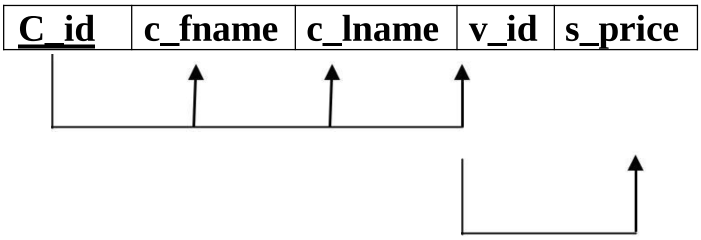


U_email	u_pass

3.2.3 Third Normal form (3nf)

A relation Schema R is in 3NF if it satisfies 2nf and no non prime attribute of R is transitively dependent on the primary key.

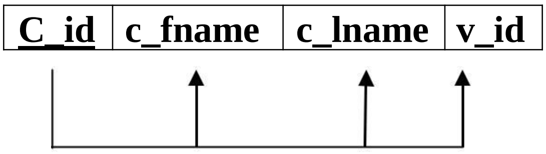
Customer



C_id	c_fname	c_lname	v_id	s_price

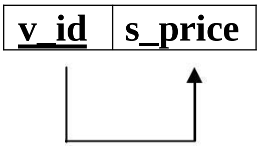
3nf form

Customer1-



C_id	c_fname	c_lname	v_id

Customer2



v_id	s_price

CHAPTER 4

IMPLEMENTATION

4.1 MODULE 1 DESCRIPTION

HTML:

Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as

`<img />` and `<input />` introduce content into the page directly. Others such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

PHP:

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page,but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

Front-End:

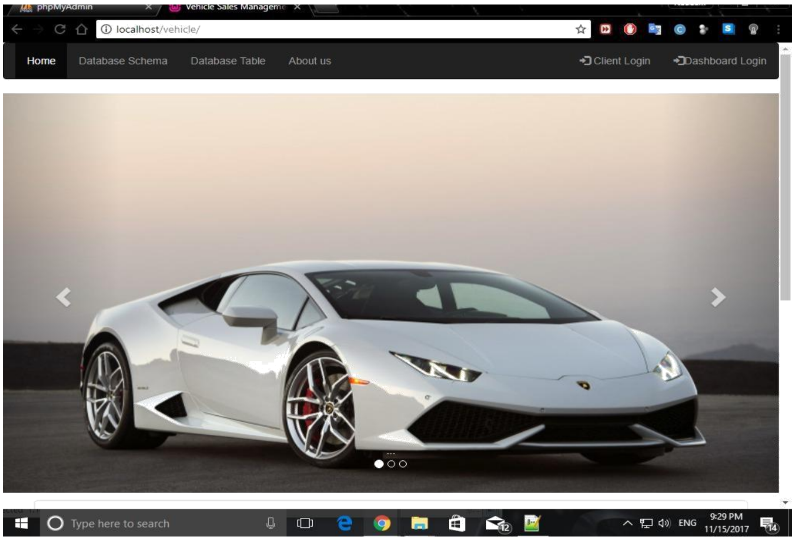
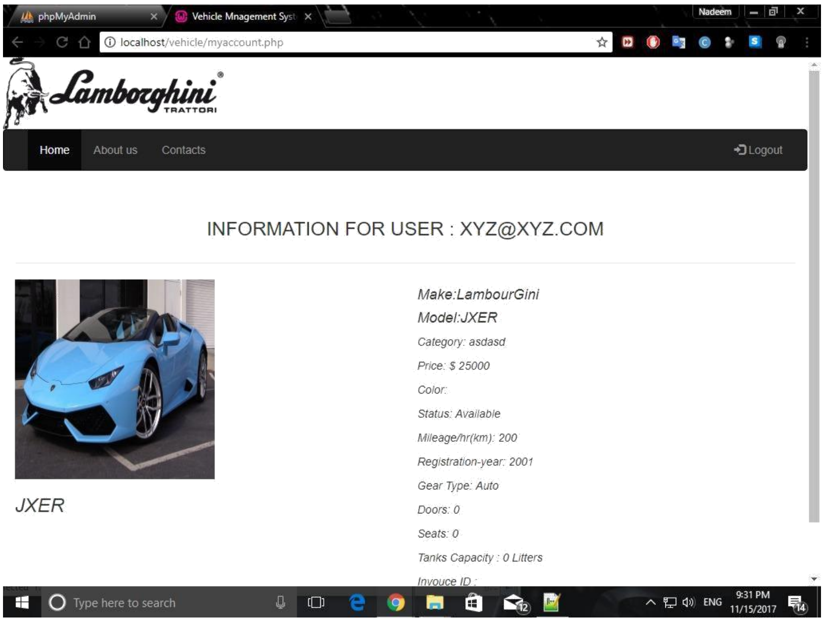


Fig: 4.1.1

- System shall allow the administrator, client to login and to add /delete /edit data related cars and its images.



INFORMATION FOR USER : XYZ@XYZ.COM

JXER

Make:LambourGini

Model:JXER

Category: asdasd

Price: $ 25000

Color:

Status: Available

Mileage/hr(km): 200

Registration-year: 2001

Gear Type: Auto

Doors: 0

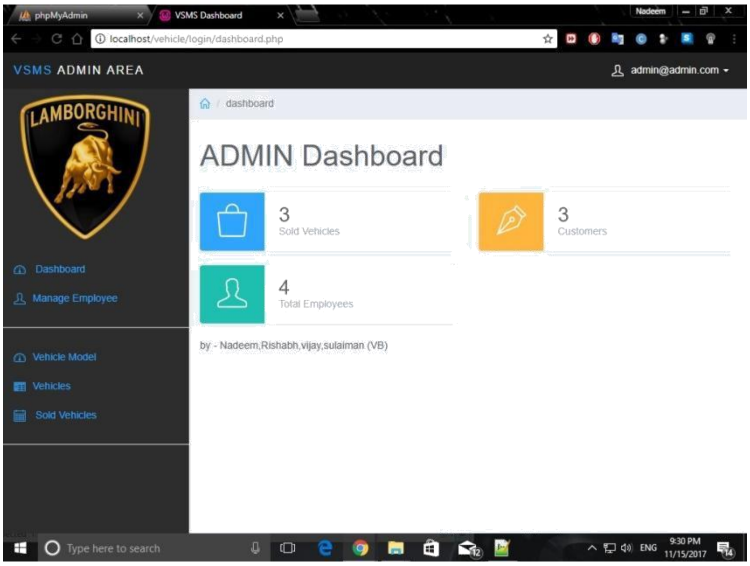
Seats: 0

Tanks Capacity : 0 Litters

Invouce ID :

Fig: 4.1.2

- This system allows the customer to login and see the vehicle details such as registration year,etc.



ADMIN Dashboard

3 Sold Vehicles

3 Customers

4 Total Employees

by - Nadeem,Rishabh,vijay,sulaiman (VB)

Fig: 4.1.3

- This system allows the admin to manage the employees and maintain the vehicle related details and sell the vehicle to customer.

4.2 MODULE 2 DESCRIPTION

MySQL:

We have used, in this project , MySQL which is an open-source relational database management system. MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook,Twitter, Flickr,and YouTube.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

Back-End:

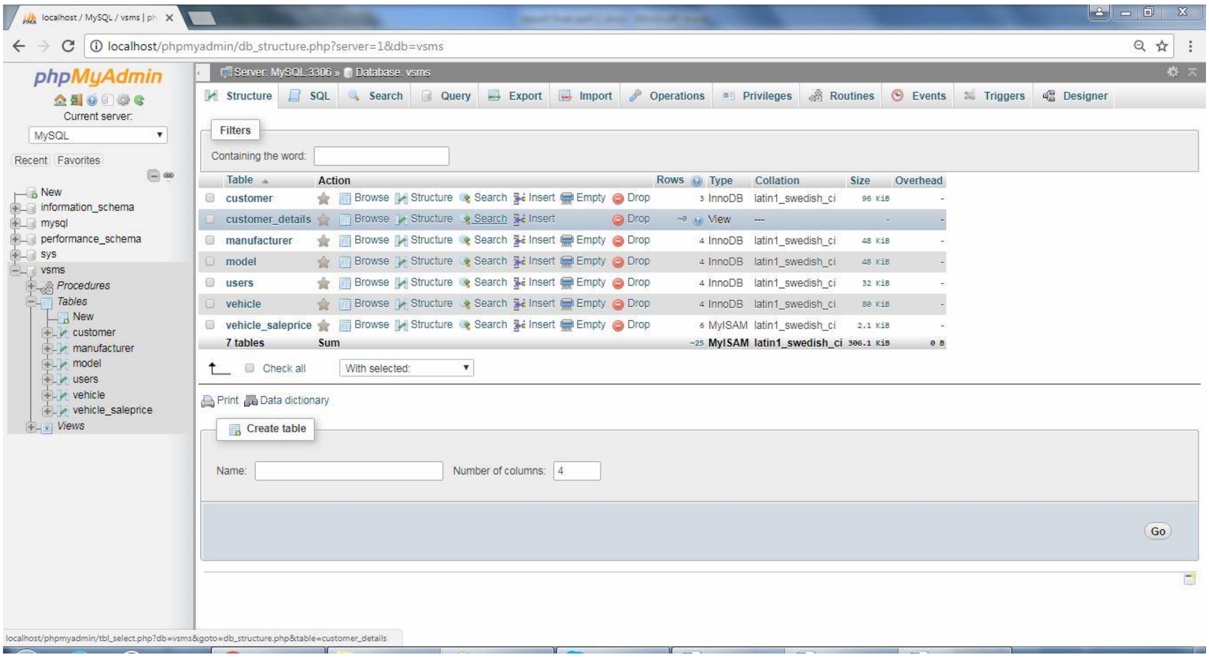
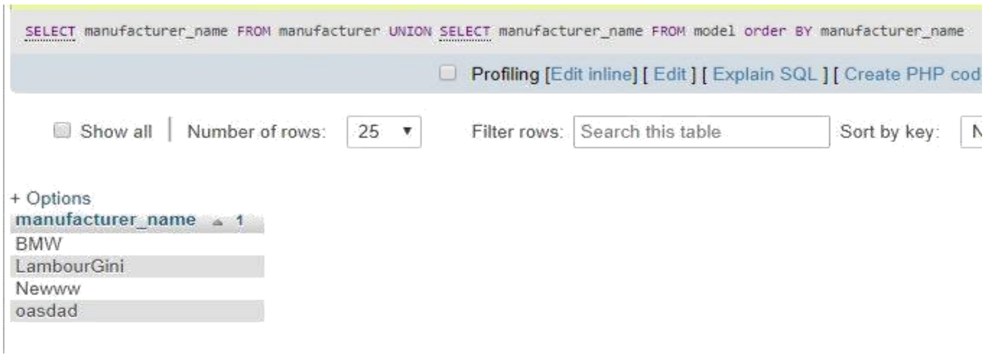


Table	Action	Rows	Type	Collation	Size	Overhead
customer	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	96 KiB	-
customer_details	Browse Structure Search Insert Drop	~0	View	---	-	-
manufacturer	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	48 KiB	-
model	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	48 KiB	-
users	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	32 KiB	-
vehicle	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	80 KiB	-
vehicle_saleprice	Browse Structure Search Insert Empty Drop	6	MyISAM	latin1_swedish_ci	2.1 KiB	-
7 tables	Sum	~25	MyISAM	latin1_swedish_ci	306.1 KiB	0 B

Fig 4.2.1

4.3 QUERIES AND THEIR OUTPUT:

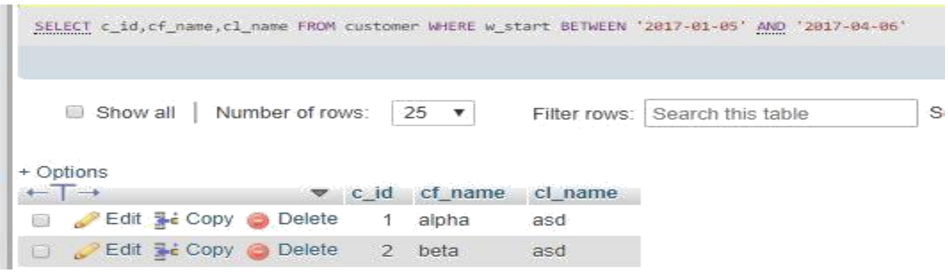


```
SELECT manufacturer_name FROM manufacturer UNION SELECT manufacturer_name FROM model order BY manufacturer_name
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP cod
□ Show all | Number of rows: 25 ▼ Filter rows: Search this table Sort by key: N
+ Options

manufacturer_name ▲ 1
BMW
LambourGini
Newww
oasdad

- To retrieve distinct manufacturer name from the table manufacturer and model ordered by the name using UNION.

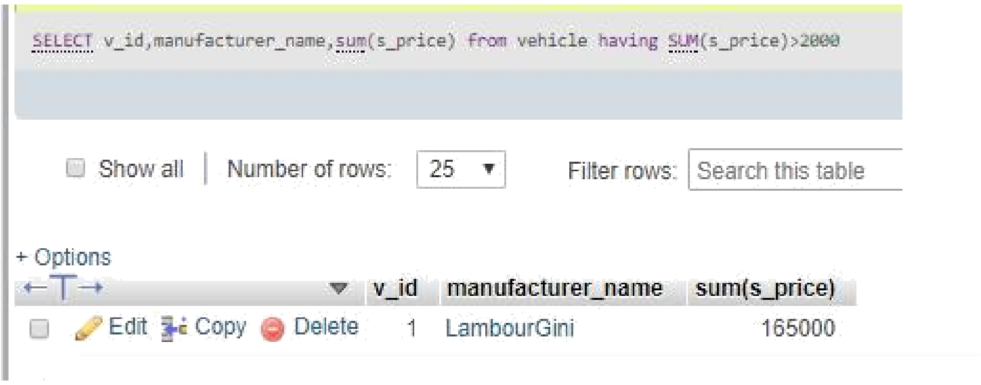


```
SELECT c_id,cf_name,cl_name FROM customer WHERE w_start BETWEEN '2017-01-05' AND '2017-04-06'
```

□ Show all | Number of rows: 25 ▼ Filter rows: Search this table S
+ Options
←T→

		c_id	cf_name	cl_name
□	Edit Copy Delete	1	alpha	asd
□	Edit Copy Delete	2	beta	asd

- To retrieve the warranty date BETWEEN 2017-01-15 and 2017-04-06.

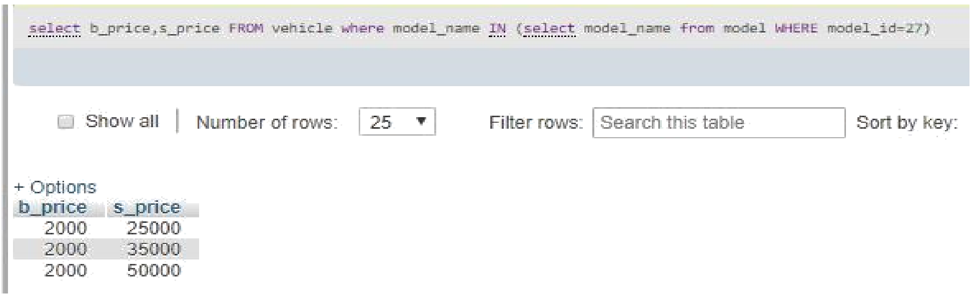


```
SELECT v_id,manufacturer_name,sum(s_price) from vehicle having SUM(s_price)>2000
```

□ Show all | Number of rows: 25 ▼ Filter rows: Search this table
+ Options
←T→

		v_id	manufacturer_name	sum(s_price)
□	Edit Copy Delete	1	LambourGini	165000

- To retrieve vehicle id, manufacturer name and sum of selling price using HAVING.

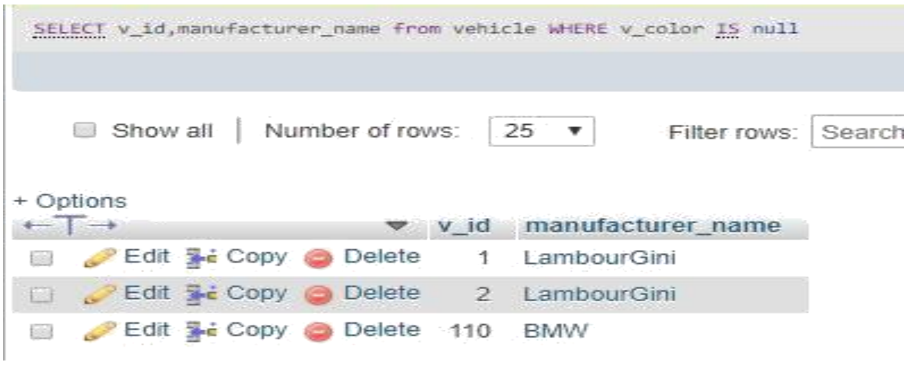


```
select b_price,s_price FROM vehicle where model_name IN (select model_name from model WHERE model_id=27)
```

□ Show all | Number of rows: 25 ▼ Filter rows: Search this table Sort by key:
+ Options

b_price	s_price
2000	25000
2000	35000
2000	50000

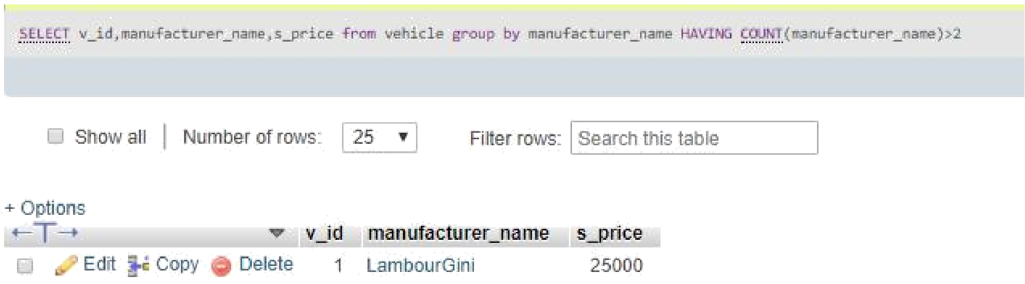
- To the base and selling price vehicle where model id is 27 using IN



```
SELECT v_id,manufacturer_name from vehicle WHERE v_color IS null
```

v_id	manufacturer_name
1	LambourGini
2	LambourGini
110	BMW

- To retrieve the vehicle id and manufacture name where vehicle color IS null.

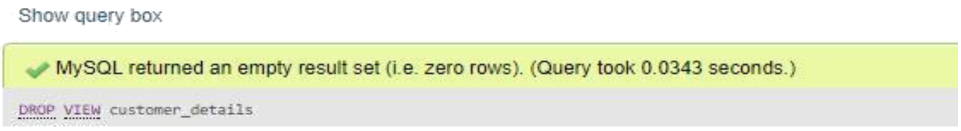


```
SELECT v_id,manufacturer_name,s_price from vehicle group by manufacturer_name HAVING COUNT(manufacturer_name)>2
```

□ Show all | Number of rows: 25 ▼ Filter rows: Search this table
+ Options
Edit Copy Delete

v_id	manufacturer_name	s_price
1	LambourGini	25000

- To retrieve selling price, v_id from vehicle having manufacture name more than 2 times

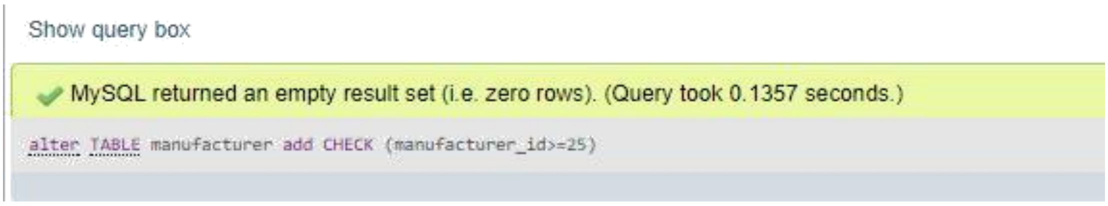


Show query box

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0343 seconds.)

```
DROP VIEW customer_details
```

- To drop the view customer details created.

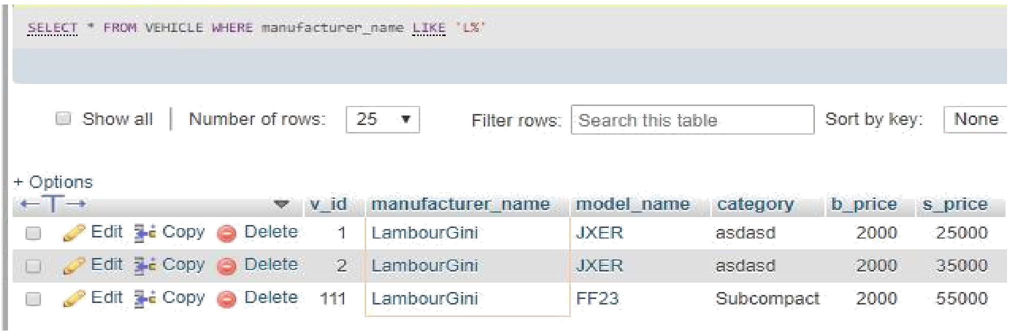


Show query box

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1357 seconds.)

```
alter TABLE manufacturer add CHECK (manufacturer_id>=25)
```

- Using CHECK keep the manufacture id always greater than 25.



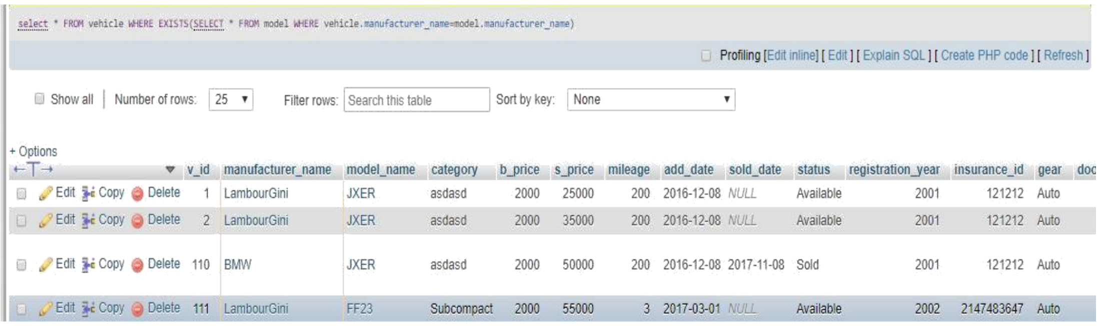
```
SELECT * FROM VEHICLE WHERE manufacturer_name LIKE 'L%'
```

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None

+ Options

←T→	v_id	manufacturer_name	model_name	category	b_price	s_price
□ Edit Copy Delete	1	LambourGini	JXER	asdasd	2000	25000
□ Edit Copy Delete	2	LambourGini	JXER	asdasd	2000	35000
□ Edit Copy Delete	111	LambourGini	FF23	Subcompact	2000	55000

- To retrieve all the manufacture name starting with L using LIKE.



```
select * FROM vehicle WHERE EXISTS(SELECT * FROM model WHERE vehicle.manufacturer_name=model.manufacturer_name)
```

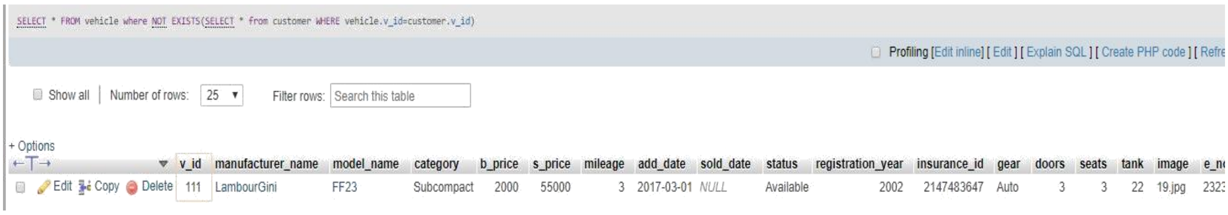
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

+ Options

←T→	v_id	manufacturer_name	model_name	category	b_price	s_price	mileage	add_date	sold_date	status	registration_year	insurance_id	gear	doc
□ Edit Copy Delete	1	LambourGini	JXER	asdasd	2000	25000	200	2016-12-08	NULL	Available	2001	121212	Auto	
□ Edit Copy Delete	2	LambourGini	JXER	asdasd	2000	35000	200	2016-12-08	NULL	Available	2001	121212	Auto	
□ Edit Copy Delete	110	BMW	JXER	asdasd	2000	50000	200	2016-12-08	2017-11-08	Sold	2001	121212	Auto	
□ Edit Copy Delete	111	LambourGini	FF23	Subcompact	2000	55000	3	2017-03-01	NULL	Available	2002	2147483647	Auto	

- To check the existence of a same entry for a particular attribute in the 2 tables specified.

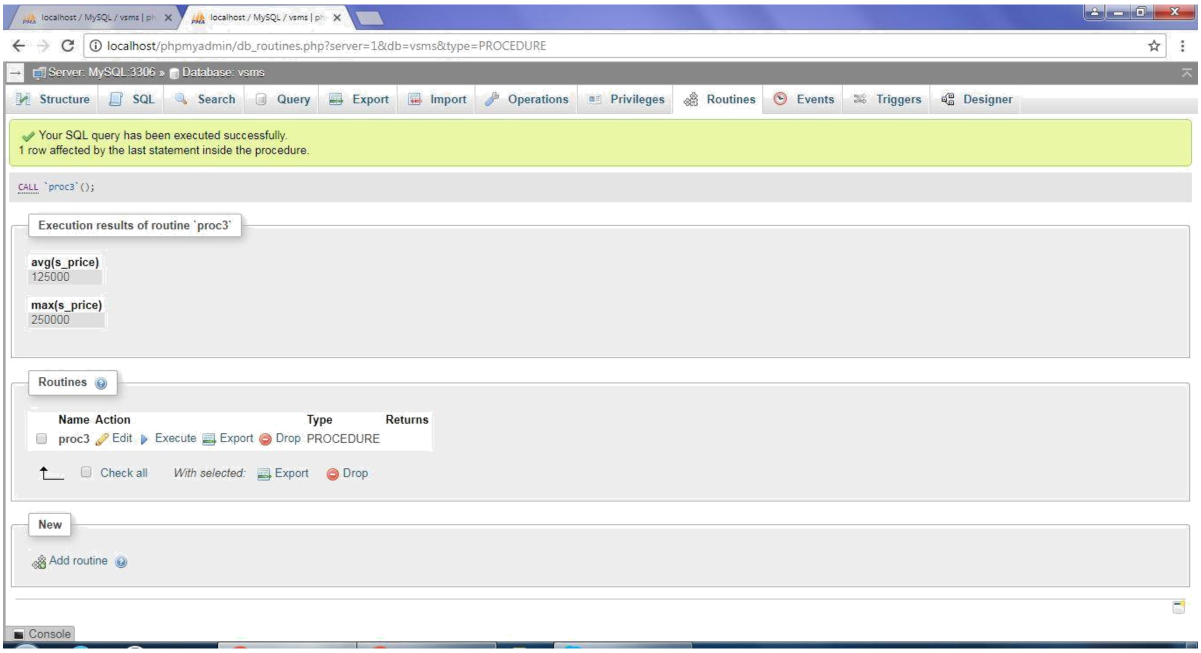


```
SELECT * FROM vehicle where NOT EXISTS(SELECT * from customer WHERE vehicle.v_id=customer.v_id)
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refre
□ Show all | Number of rows: 25 ▼ Filter rows: Search this table
+ Options

	v_id	manufacturer_name	model_name	category	b_price	s_price	mileage	add_date	sold_date	status	registration_year	insurance_id	gear	doors	seats	tank	image	e_n
□ Edit Copy Delete	111	LambourGini	FF23	Subcompact	2000	55000	3	2017-03-01	NULL	Available	2002	2147483647	Auto	3	3	22	19.jpg	2323

- To check the non existence of vehicle id in the vehicle table.



localhost / MySQL / vsms | ph
localhost / MySQL / vsms | ph
localhost/phpmyadmin/db_routines.php?server=1&db=vsms&type=PROCEDURE
Server: MySQL:3306 » Database: vsms
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Designer

Your SQL query has been executed successfully.
1 row affected by the last statement inside the procedure.

```
CALL `proc3`();
```

Execution results of routine `proc3`

avg(s_price)
125000

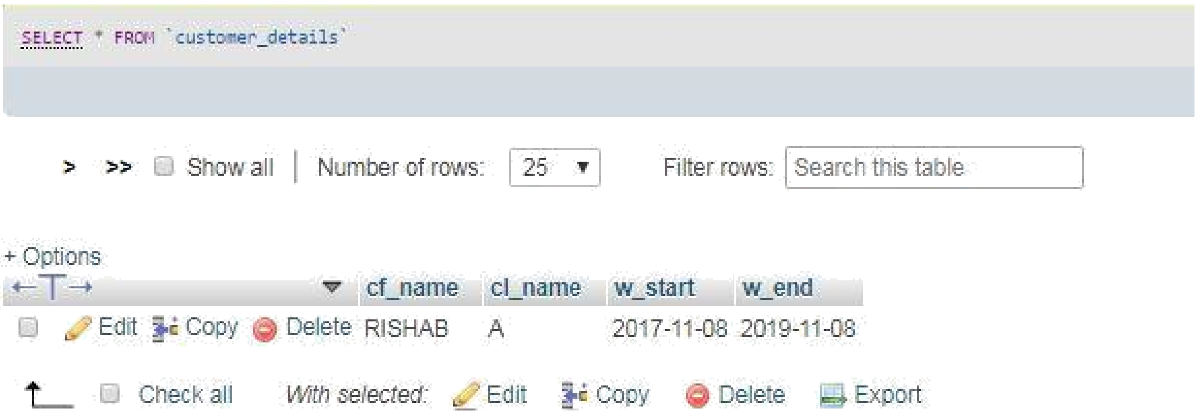
max(s_price)
250000

Routines

	Name	Action	Type	Returns
□	proc3	Edit Execute Export Drop	PROCEDURE	

□ Check all With selected: Export Drop
New
Add routine
Console

- **Stored Procedure:** To find the average of selling price by the use of stored procedure.



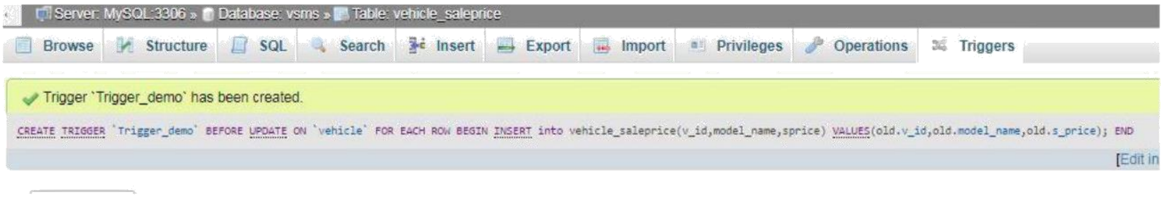
```
SELECT * FROM `customer_details`
```

> >> □ Show all | Number of rows: 25 ▼ Filter rows: Search this table
+ Options

	cf_name	cl_name	w_start	w_end
□ Edit Copy Delete	RISHAB	A	2017-11-08	2019-11-08

□ Check all With selected: Edit Copy Delete Export

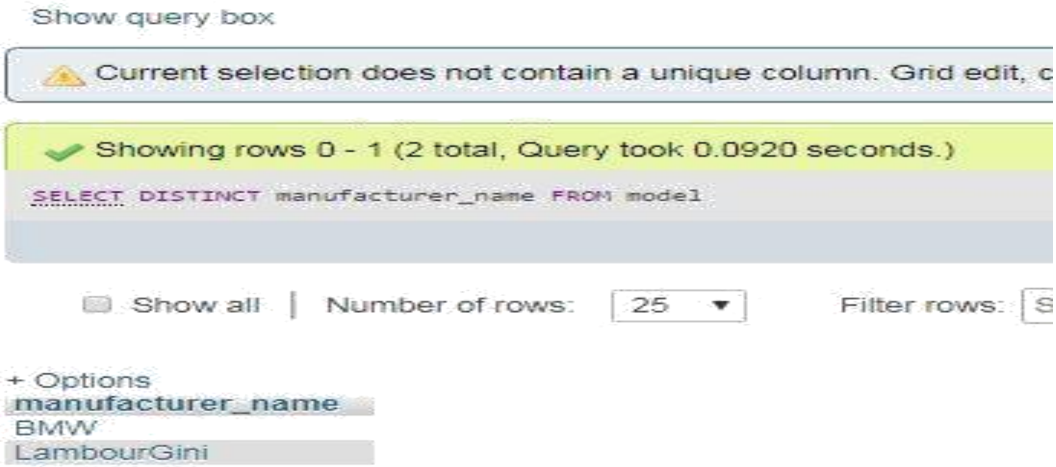
- To create a view named customer_details with attributes cfname,cl_name,w_start



✓ Trigger `Trigger_demo` has been created.

```
CREATE TRIGGER `Trigger_demo` BEFORE UPDATE ON `vehicle` FOR EACH ROW BEGIN INSERT into vehicle_saleprice(v_id,model_name,sprice) VALUES(old.v_id,old.model_name,old.s_price); END
```

- **Trigger** – using this we can keep a track of the changes in the selling price of the vehicles.



⚠ Current selection does not contain a unique column. Grid edit, c

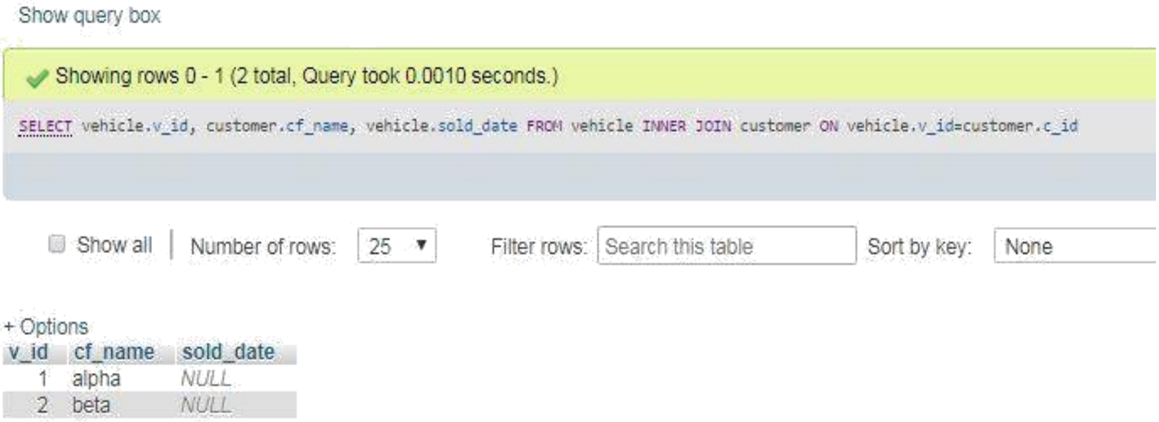
✓ Showing rows 0 - 1 (2 total, Query took 0.0920 seconds.)

```
SELECT DISTINCT manufacturer_name FROM model
```

□ Show all | Number of rows: 25 ▼ Filter rows: S

manufacturer_name
BMW
LambourGini

- To select the DISTINCT manufacturer_name from model .



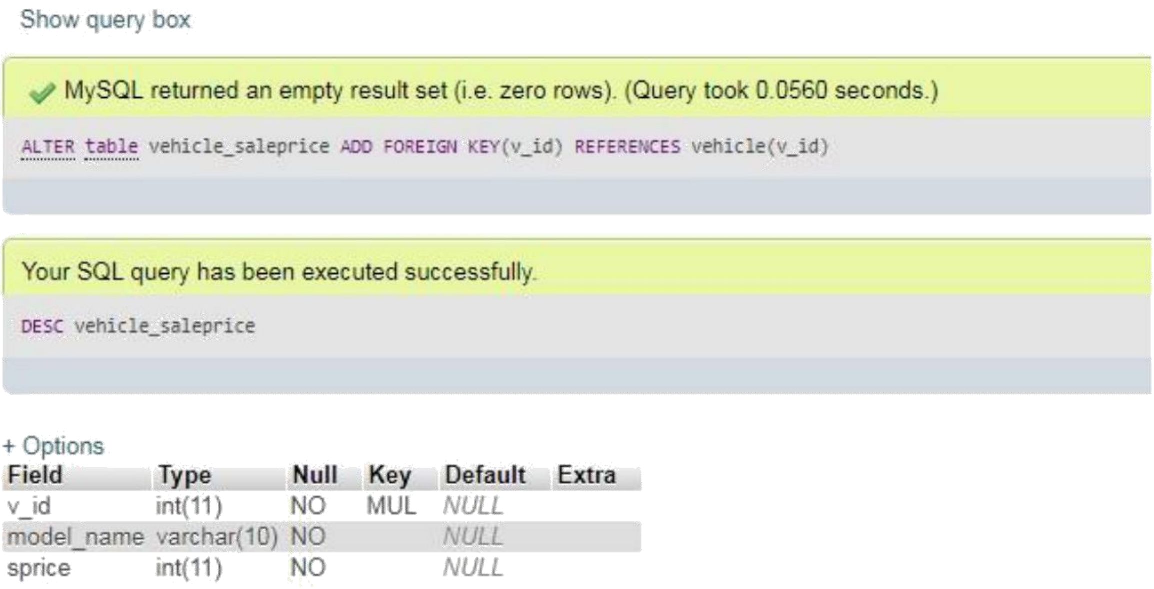
✓ Showing rows 0 - 1 (2 total, Query took 0.0010 seconds.)

```
SELECT vehicle.v_id, customer.cf_name, vehicle.sold_date FROM vehicle INNER JOIN customer ON vehicle.v_id=customer.c_id
```

□ Show all | Number of rows: 25 ▼ Filter rows: Search this table Sort by key: None

v_id	cf_name	sold_date
1	alpha	NULL
2	beta	NULL

- Select v_id,cf_name,sold_date using JOIN.



Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0560 seconds.)

```
ALTER table vehicle_saleprice ADD FOREIGN KEY(v_id) REFERENCES vehicle(v_id)
```

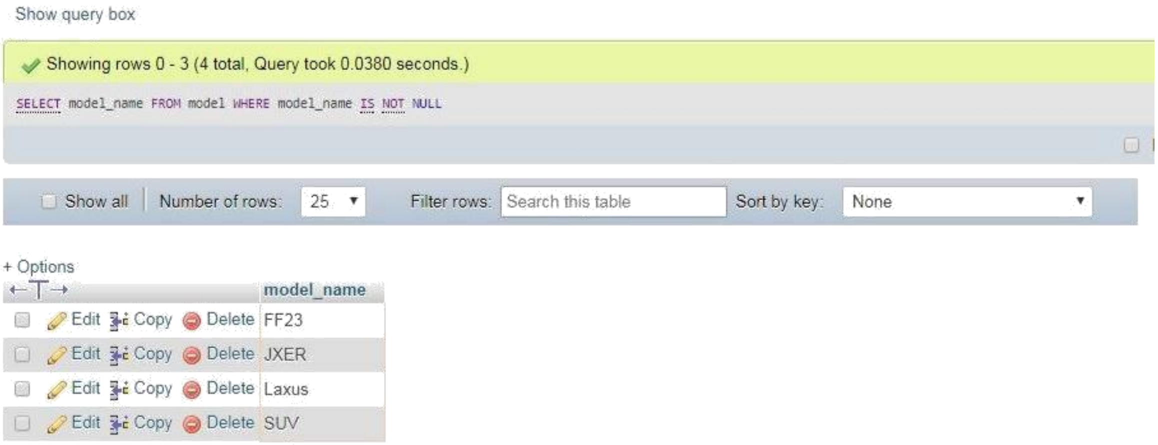
Your SQL query has been executed successfully.

```
DESC vehicle_saleprice
```

+ Options

Field	Type	Null	Key	Default	Extra
v_id	int(11)	NO	MUL	NULL	
model_name	varchar(10)	NO		NULL	
sprice	int(11)	NO		NULL	

- To add a Foreign KEY to model name in the table vehicle_salesprice.



Show query box

✓ Showing rows 0 - 3 (4 total, Query took 0.0380 seconds.)

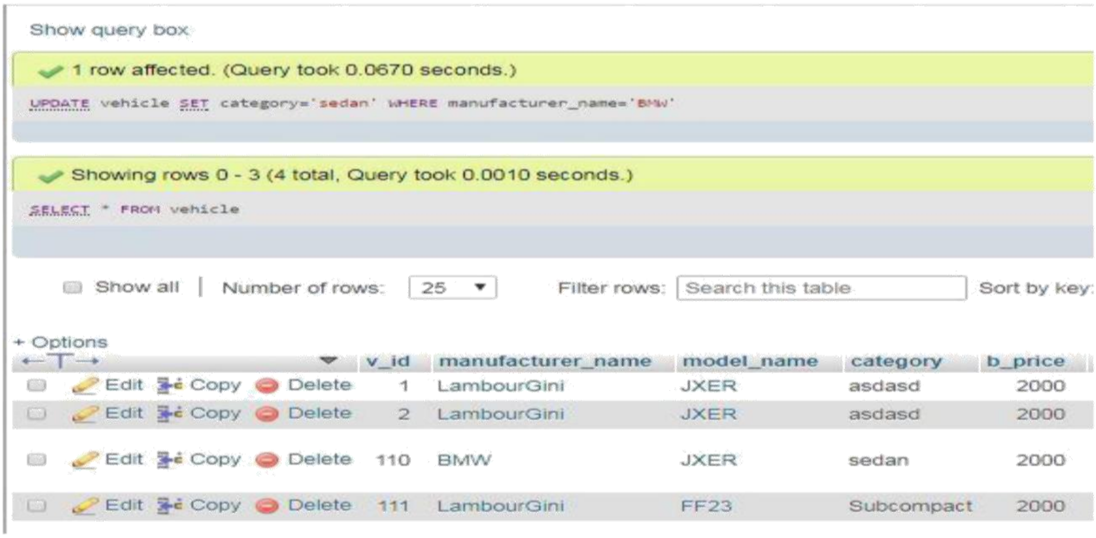
```
SELECT model_name FROM model WHERE model_name IS NOT NULL
```

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	model_name
□ Edit Copy Delete	FF23
□ Edit Copy Delete	JXER
□ Edit Copy Delete	Laxus
□ Edit Copy Delete	SUV

- To select model name that is not NULL.



Show query box

✓ 1 row affected. (Query took 0.0670 seconds.)

```
UPDATE vehicle SET category='sedan' WHERE manufacturer_name='BMW'
```

✓ Showing rows 0 - 3 (4 total, Query took 0.0010 seconds.)

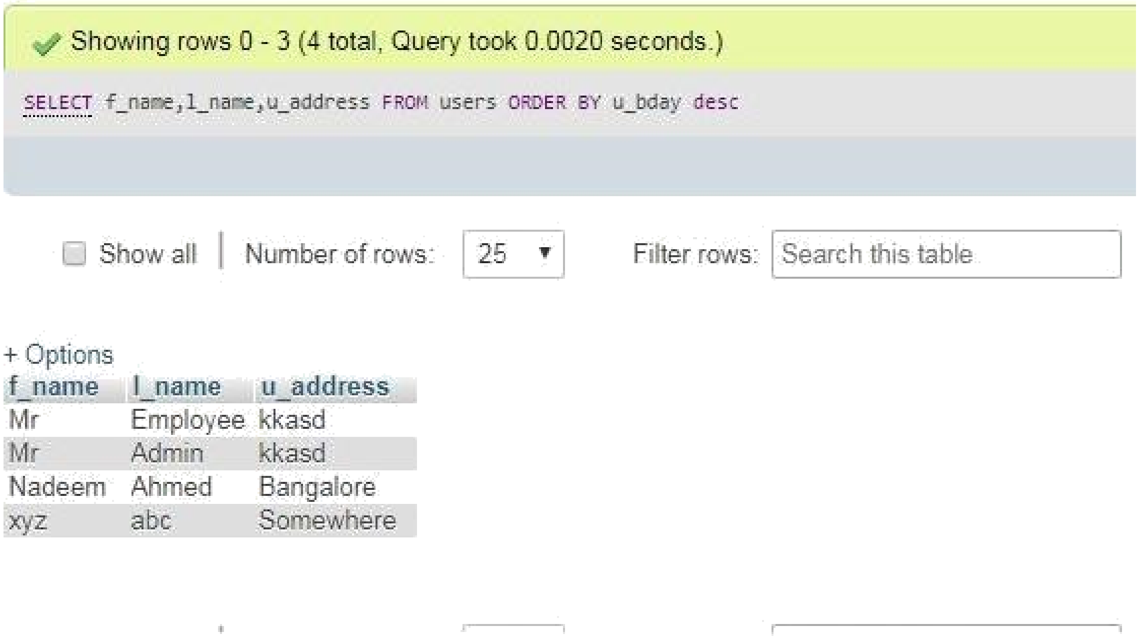
```
SELECT * FROM vehicle
```

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key:

+ Options

				v_id	manufacturer_name	model_name	category	b_price
□	Edit	Copy	Delete	1	LambourGini	JXER	asdasd	2000
□	Edit	Copy	Delete	2	LambourGini	JXER	asdasd	2000
□	Edit	Copy	Delete	110	BMW	JXER	sedan	2000
□	Edit	Copy	Delete	111	LambourGini	FF23	Subcompact	2000

- To UPDATE the category of the manufacturer name BMW to Sedan**.**



✓ Showing rows 0 - 3 (4 total, Query took 0.0020 seconds.)

```
SELECT f_name,l_name,u_address FROM users ORDER BY u_bday desc
```

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table

+ Options

f_name	l_name	u_address
Mr	Employee	kkasd
Mr	Admin	kkasd
Nadeem	Ahmed	Bangalore
xyz	abc	Somewhere

- To retrieve the first name, last name ,u address from user ORDERED BY the u_bday.

CHAPTER 5

TESTING

5.1 INTRODUCTION TO TESTING

Unit Testing:

Unit testing focuses verification effort on the smallest unit of software design that is the module. Using procedural design description as a guide, important control paths are tested to uncover errors within the boundaries of the module. The unit test is normally white box testing oriented and the step can be conducted in parallel for multiple modules.

Validation Testing:

At the end of integration testing software is completely assembled as a package. Validation testing is the next stage, which can be defined as successful when the software functions in the manner reasonably expected by the customer. Reasonable expectations are those defined in the software requirements specifications. Information contained in those sections form a basis for validation testing approach.

Integration Testing:

Integration testing is a systematic technique for constructing the program structure, while conducting test to uncover errors associated with the interface. The objective is to take unit tested methods and build a program structure that has been dictated by design.

5.2 TEST CASES

Test id	Test case	Expected result	Actual Result	Status
1	Login	User is successfully Authenticated.	User is successfully Authenticated.	Pass
2	Sell	Vehicle is Successfully sold.	Vehicle is Successfully sold.	Pass
3	Addition of vehicle	New vehicle Successfully added.	New vehicle Successfully added.	Pass
4	Manage employee	Insertion of new employee done Successfully.	Insertion of new employee done Successfully.	Pass

Description

1. User enters the login id and password and is successfully logged in.

2. Admin sells a vehicle to customer by entering the customer details.

3. Admin/Employee adds new model of vehicle and its various details.

4. Admin manages the details related to the employee.

CONCLUSION

The Vehicle Management application has rich user interface so that novice users can access easily. This application provides the management reports like Occupancy report, Approval status report, to track the usage of company’s transport facility. Our project is only a humble venture to satisfy the needs in a library. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

REFERENCES

[1] Database system models, languages, design and application programming, ramez elmasri and shamkant b.navathe, 7th edition,2017, Pearson.

[2] www.projectsgeek.com

[3] Silberschatz korth and sudharshan,database system concepts,6th edition,mc-graw hill 2013.

[4] www.W3schools.org

[5] https://www.stackoverflow.com

[6] www.quora.co.in

[7] Paper 1: A vehicle sales management system in improving the relational use of customer requirements, Authors: T.Shang, X.Ren, R.Ma,2017 Edition

[8] Paper 2: Practice of administration management: A review of Literature ,Author: Ayad k.Ali,2016 Edition