

R.V.C.E, COMPUTER SCIENCE AND
ENGINEERING DEPARTMENT

SHOWROOM AUTOMATION SYSTEM

D.B.M.S PROJECT FOR VI SEMESTER BE
MARCH – AUGUST 2001

DESIGNED AND DEVELOPED BY:

SRIVAS N CHENNU (1RV98CS86)

And

VISHWAS N (1RV98CS104)

CERTIFICATE



This is to certify that the project titled **Show Room Automation System** has been satisfactorily completed by **Srivas N. Chennu (Roll No. 1RV98CS086)** in fulfillment of the requirements for the Database Applications Lab for VI Semester B.E.(CSE) course during the academic semester from March 2001 to July 2001 as prescribed by the Visveswaraiah Technological University

Signature of the Teacher
In charge

DATE:

Signature of the
Head of Department

DATE:

ACKNOWLEDGEMENT

I wish to express my deepest gratitude to our H.O.D Mr. B.I.Khodaipur and Mr. Satish, for their support and guidance throughout our project in DBMS.

I am thankful of the technical staff for the co-operation and assistance provided by them.

Last of all I thank our management for providing infrastructure and allocating enough time on the machines for us to complete our project successfully.

Vishwas. N

Srivas N Chennu

CONTENTS

- Introduction
- SRS
- Entity Relationship Diagram
- Table Description
- Database Schema
- Forms and Reports
- Conclusion
- Bibliography

INTRODUCTION

1.1 INTRODUCTION TO D. B. M. S

Databases and database technologies are having a major impact on the growing use of computers. Databases play a critical role in almost all areas where computers are used including business, engineering, medicine, law, education and science.

Database Management System (DBMS)

Definition:

A DBMS is a collection of programs that enables to create, maintain and manipulate a database. The DBMS is hence a general-purpose software system that facilitates the process of defining, constructing, and manipulating databases for various applications.

Database:

A database is a collection of related information. Defining a database involves specifying the data type, attributes and constraints for the data to be stored.

Constructing a database is the process of storing itself on some storage medium like disk or tape that can be handled by the DBMS. Manipulating a database includes such functions like querying a database to retrieve specific data, updating the database to reflect the changes in the mini world and generating reports from the data.

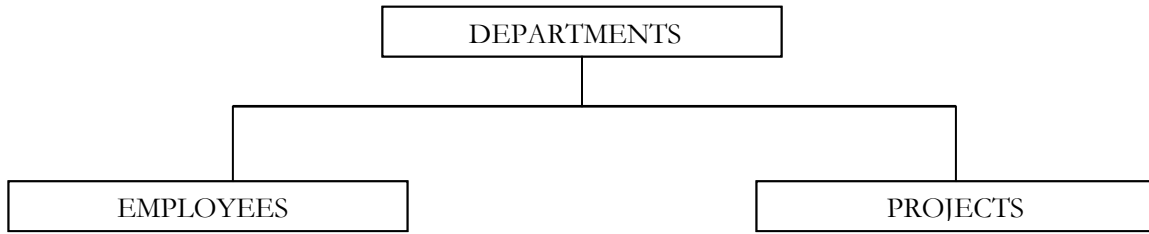
Data Model:

A data model is a set of concepts that can be used to describe the structure of a database.

Data models are classified into:

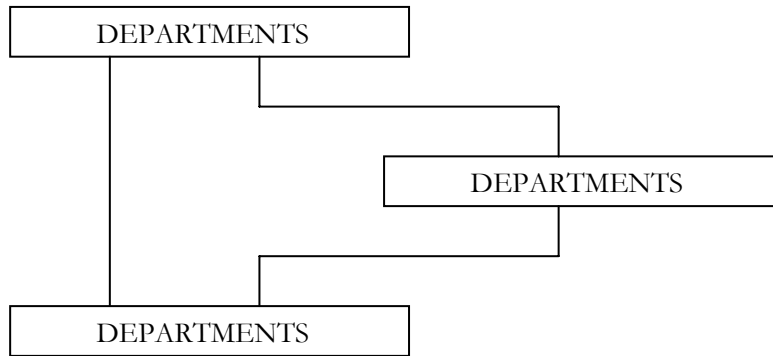
Hierarchical Data Model:

This model represents data as hierarchy trees. The drawback with this data model is that it is tedious to reorganize the tree, as the hierarchy has to be maintained.



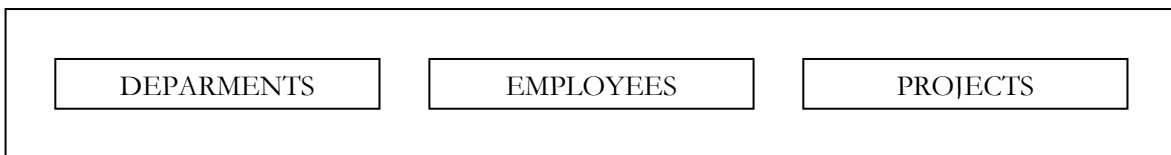
Network Data Model:

This model represents data having multiple parent-child relationships. The drawback of this model is that it is difficult to reorganize and to program in database manipulation techniques.



Relational Data Model:

It combines the advantages of hierarchical and network data models. Here databases are represented as collections of relations. All parent child relations are eliminated. It consists of built-in query language.



The DBMS is hence general-purpose software that facilitates the process of defining, constructing and manipulating databases for various applications.

A DBMS represents complex relationships with data. It has controlled data redundancy and enforces integrity between the various entities.

Database System Users:

Database Developers:

They are computer scientists or software professionals who try to improve the performance of the DBMS software by optimizing the various access techniques to the database.

Database Administrator:

The responsibility of the data base administrator is to administrate the resources such as the database itself, the DBMS and related software. The DBA is responsible for creating and removing logins providing access permissions to the database for use. He also is responsible for allocating and monitoring the software and hardware resources to the various users.

End Users:

They are persons whose job is to access the database query, update or manipulate it to perform the various transactions.

System Analysts and Application Programmers:

System analysts determine the requirements of the end users and develop applications for transactions to meet their requirements. Application programmers implement these specifications as programs. They create, test, debug and document these applications.



1.2 STRUCTURED QUERY LANGUAGE (SQL)

It is a non-procedural query language. It is simple and consists of different statements for variety of tasks like

1. Querying Data.
2. Inserting, Updating and deleting rows in a table.
3. Creating, Replacing, Dropping, Altering objects.
4. Controlling access to the database and its objects.
5. Guaranteeing.

SQL commands are classified into four categories:

DDL (Data Definition Language)

Used to create, alter, rename and drop the objects.

DML (Data Manipulation Language)

Used to insert, delete, update and select the records in a table.

DCL (Data Control Access Language)

Used to control access to the database and its objects.
Ex: GRANT, REVOKE.

TCL (Transaction Control Language)

Used to control transactions.
Ex: COMMIT, ROLLBACK.

1.3 INTRODUCTION TO ORACLE

ORACLE is a RDBMS. In the beginning Oracle (Relational Database Inc.) was just a database company, to be specific, with little or no services. The entire documentation set for Oracle consisted of three or four books (which now is over a hundred). In addition, Oracle had no canned application. Today the picture is different. This multibillion-dollar company has lots of products, lots of services and applications. Today Oracle Company, based in Redwood Shores, California manufactures software products and delivers services for electronic management of information. Oracle is a worldwide provider of database software. It does business in over 90 countries, around the world and their software runs on more than 100 different platforms. They are a major player on the information highway.

THE ORACLE SERVER

The Oracle server is a state-of-the-art information management environment. It is repository for very large amounts of data and gives the users rapid access to the data. The Oracle Server provides for the sharing of data between applications, the information stored in one place and used by many systems.

The Oracle Server runs dozens of different computers, supporting the following configurations:

- **Host Based.**

Users are connected directly to the same computer on which the database resides.

- **Client Server**

Users access the database from the PC via the network and the database sites on a separate computer.

- **Distributed Processing**

Users access a database that resides on more than one computer.

A database is spread across more than one machine and the users are unaware of the physical locations of data (location transparency).

Oracle Today

There are many significant features that have catapulted Oracle to the top of the growing information management vendor community.

Security Mechanisms

It's sophisticated mechanisms control access to sensitive data by an assortment of privileges. Users are given the rights to view, modify and create data based on the name they use to connect to the database. Customers use these mechanisms to ensure specified users get to see sensitive data while others are forbidden.

Back and Recovery

It provides fantastic backup and recovery features. Backup creates a secondary copy of the data; recovery restores the data from the backup. The Oracle backup and recovery mechanism minimize the down time and data loss if and when the problem arises. Another important factor is that these functions are automated and hence making the BDA job easier.

Space management

Oracle offers flexible space management. You can allocate disk space for the data and control of subsequent allocation by instructions.

Open Connectivity

It provides uninterrupted access to the database 24 hours a day, and connectivity to and from other vendor's software. Using add-ons to the Oracle database, you can work with information that resides in other databases, such as DB2, Sybase, MS-Access, etc. You are permitted to store data in Oracle database and access it from other software such as Vb, Power Builder, SQL, etc.

Development Tools

The Oracle Server, commonly referred to as the database engine, supports a wide range of development tools, end-user query tools, off-the-shelf applications and office-wide information management tools.

Oracle using its own tools – The Applications

As time went on, Oracle Corporation realized that there was a strong business opportunity available for off the shelf generic applications. So Oracle started to build financial applications. This had two major impacts on the rest of Oracle Environments.

First, Oracle Corporation was actually using its tools to build applications. The more they used, they improved and found new ways to improve the tools. The company started to live, eat and breathe their own tools. Rapidly the quality and functionality of the tools as well as Oracle server began to improve.

Second, as Oracle Corporation started to build application based on the relational model, companies started to reap the benefits of the new technology Oracle had developed.

Today Oracle has a division that puts out application in every major product area from accounting to manufacturing to industry specific solutions.

Services

The service wing of the Oracle Corporation is a major component. The Services offer the following:

- ❑ **Education:** Skilled instructors teach about the wide range of its products, relational database modeling, design and analysis theory.
- ❑ **Consulting:** Trained professionals are at your disposal to help facilitate corporate system solution using Oracle's products and Oracle's partner's technology. There is a network of companies that develop and market their own Oracle's products that own alongside.
- ❑ **Industries:** Solutions to industry-specific business requirements, with special attention to addressing key issues with Oracle based technology.

The Services Oracle provides can be grouped into four components:

1. Oracle Worldwide Customer Support.
2. Oracle Education.
3. Oracle Industries.
4. Oracle Consulting.

An increasing portion of Oracle business is generated in the services area each year. The services enable the customer to implement dependable, state-of-the-art technology system that deploys the Oracle Cooperative Server Technology. Services are geared toward knowledge transfer and lasting success of system solutions realized in partnership with Oracle professionals.

1.4 DEVELOPER 2000

Oracle Developer 2000 is high-end platform independent suit of tools. Oracle developer 2000 components are geared towards rapid applications development (RAD) and productivity. These applications range from the individual workstation to the corporate distribute database. The three main components of developer 2000 are Oracle forms, Oracle Reports and Oracle Graphics.

ORACLE FORMS

Oracle forms is a powerful 4GL for creating interactive applications. It is the most important member of the Developer 2000 application development suit. Usually, any application created using Developer 2000 is multi-user, client/server, and platform independent application. However, the developer can change the innate nature of a forms application.

Oracle forms as said earlier is the backbone of the Developer 200 suit and is the tool with the ability to interactive data screens and menus. Forms can launch other forms, other Developer tools, C-programs and Applications. It is also complete enough to create complex database application without any other tools.

The basic objects can be created Oracle forms, menus, and libraries. Forms and menus are usually integrated in an application. A menu functions through its attachment to a form application. A menu functions through its attachment to form an application. A menu functions through its attachment to a form application. Forms and menus are separate objects. This allows menus to be shared between forms, allows a form to use different menus vice versa. Libraries are simple repositories of PL/SQL code that can be used by the application.

There are three icons created in the Developer 2000 tools folder to access Oracle forms:

- Oracle Forms runtime
- Oracle Forms generate
- Oracle Forms designer

The Oracle forms designer is capable of calling both forms runtime and generate modules when needed, so that it is not necessary to swap between applications while developing a form.

Source code for a form, menu or library may be stored in a file and/or in Oracle database tables. When stored in a file, which is the usual practice, the source code filename will have an extension “.fmb”. From the fmb file an executable is generated directly, but can be loaded and run from Oracle forms runtime module. It cannot be modified unless the source code (the .fmb file) is available.

FORMS DESIGNER

The form designer gives an environment in which forms can be created, tested and debugged. The most important component of the forms designer is the object navigator. The object navigator provides access to the various components that make up a form, menu or library and provides access to the database objects such as triggers and procedures. The object navigator also appears in the Oracle graphics and Oracle reports components of the developer 2000.

The object navigator allows easy movement between the items in a form, menu or library. The object can be edited, modified or created through the object navigator. The main window of the object navigator shows each object or object type, preceded by +,-,* symbol.

1.5 NORMALIZATION

It is a technique that allows the designer to deduct redundancies in a given table. Dr. Codd proposed normalization. He says the process of applying the rule of normal forms is called normalization.

There are six degrees of normalizations. They are

- First Normal form (1nF)
- Second Normal form (2nF)
- Third Normal form (3nF)
- Boycee-Codd Normal form (BCNF)
- Fourth Normal form (4nF)
- Fifth Normal form (5nF)

Normal forms were developed primarily for use on relational database. They are useful tools for testing database record layouts in all the data models. Normalization takes care of update anomalies.

The product satisfies the Third Normal form.

First Normal Form (1nF)

A database satisfies 1nF if and only if it has no multi-valued, composite attributes or their combinations.

Second Normal Form (2nF)

A database satisfies 2nF if and only if there is full functional dependency that is removal of any attribute from a relation means that the relation does not hold anymore.

Third Normal Form (3nF)

A database satisfies 3nF if and only if there is no transitive dependencies, that is for every relation

$$X \rightarrow A \text{ holds in } R$$

X is a super key of R or A is a prime attribute of R.

1.6 SHOWROOM AUTOMATION SYSTEM

The Showroom Automation System has been designed using ORACLE as a backend tool and Developer 2000 as the front-end tool.

The use of ORACLE facilitates for the user

- Easy visualization and understanding of the database structure.
- Simple creation of any number of temporary relationships between tables.
- Freedoms as to how to query the database through the use of SQL or PL/SQL.
- Easy visualization of tables.
- Relational joins that provide temporary set of data from multiple tables in the model.

Use of Oracle's non-procedural structured query language enables the programmer to use the numerous tools available to support its system, thereby freeing the programmer from tedious programming tables and spend more time on the development of his work. This increases the productivity of the developer.

ORGANIZATION OVERVIEW:

The organization provides following services:

- Selling vehicles to customers.
- Tracking vehicle sale and bookings.
- Maintaining customer database.
- Maintaining stock details of vehicles available.
- Maintaining spares database.

Based on the availability of a particular vehicle the customer is allotted a vehicle or else put into the waiting list. The customer is required to fill in the registration form, furnishing personal details of like name, address, telephone and customer number. The entries in the waiting list are deleted only if the customer does not report by the date of delivery.

The showroom automation system is responsible for maintaining information of customer visiting the showroom. As and when the vehicles are bought, allocation takes place, which in turn leads to updating of the waiting list and stock. The system is also responsible for generating reports showing details of customers, the available stock, and vehicles allotted during a period of time.

Lastly the showroom automation system provides access to administrator, manager and clerk to whom following responsibilities are assigned:

- 1) Adding or removing vehicle details.
- 2) Adding new vehicle allotments.
- 3) Adding new stock.
- 4) Updating colors available.
- 5) Adding new spares and stock updating.



2. SOFTWARE REQUIREMENT SPECIFICATION (SRS)

Introduction:

Purpose

The purpose of this system is to develop an application to automate a vehicle showroom, which handles all kinds of transactions required.

Scope

The scope of this system is to implement the computerization of the vehicle details, vehicle allotment, vehicle bookings, spares management and also vehicle stock management.

References

- i. Oracle Reference Manual.
- ii. Commercial application development using ORACLE Developer 2000.
-By Ivan Bayross

Developer's responsibilities:

To develop a DBMS package to automate the transactions of a vehicle showroom and to store the transaction details

General Description:

Product function Overview:

This system is responsible for the following functions:

Vehicle Master: Querying the database to find the availability of a type of a particular vehicle type. The user is prompted with to enter the vehicle like name, model no and company. The user gets to know the status for different vehicles in each category.

Vehicle Stock: Taking inputs from user regarding the details of individual vehicles, unique properties like engine no, chassis no and color. Checking whether the customer can be allocated a vehicle based on the availability. This involves querying the database to find whether there is a vehicle suiting the customer. The allotment table and waiting list tables are updated correspondingly.

Customer: The customer details of purchase are compared with the database. The entries corresponding to them are updated as and when a Customer buys a car.

Vehicle Allotment: The user is prompted for Car details. The vehicle master and vehicle stock information is searched in the database. The result consists of Customer buying a particular vehicle after furnishing details if car type is available.

Waiting List: The user is prompted for Car details. The vehicle master and vehicle stock information is searched in the database. The result consists of Customer booking a particular vehicle after furnishing details if car type is not available.

Spare Stock: This contains the spares for a given vehicle and the quantity available.

Colors: This contains the different type of color available in each model.

Pictures: This contains the different pictures of cars.

User characteristics:

The main users of this system will be the showroom employees who are assumed computer literate.

General constraints:

It requires Developer 2000 platform. Runs on Windows environment with database mounted.

General assumptions and dependencies: Not applicable.

Functional Requirements:

General description of Input & Output files.

Input file 1: Vehicle Master

This file stores in it the various types of vehicles available in the market.

Input file 2: Stock

This file stores the various available vehicles in the showroom and is referenced during vehicle allotment process

Input file 3: Customer

This file allows adding/updating the details.

Input file 4: Colors

This file allows us to enter colors for each model.

Normalization:

Normal forms provide database designers with a framework for analyzing the input files or the schema based on their keys and the functional dependencies between their attributes. The input files mentioned above are normalized that is they are checked to see if they are in the defined normal forms.

The step-by-step normalization process for the system is as follows:

1st Normal Form: The system satisfies 1st Normal form because each attribute domain includes only the atomic values and the value of any attribute is single.

2nd Normal Form: The system satisfies 2nd normal form because, in any relation schema (input file) R, all non-prime attributes are fully functionally dependent on the primary key of the schema R.

3rd Normal Form: The system is in 3rd Normal Form because it is already in 2nd NF and none of the prime attributes of any relation schema R are transitively dependent on their primary keys.

4th Normal Form: Though the system has multi-valued attributes, they are split into separate tables and hence no multi-valued dependencies arise. Therefore, the question of 4th NF does not arise.

5th Normal Form: Since no join dependencies exist between schemas of the system, there is no need to do the 5th NF procedure.

□ **Reports**

Output File 1: Availability of Stock.

Output File 2: Customer information.

Output File 3: Vehicle master information.

Output File 4: Allotment details.

Output File 5: Waiting list details.

Output File 6: Color details

Detailed Input File Formats:

The input file formats for any database system are the descriptions of the tables created to store the records of the main inputs to the system.

□ Tables:

The format of the tables is:

- o Column wise: Field Name, Data type, Constraints.
- o Row wise: Different records within the table are stored row wise.

The different tables in the system are:

Vehicle Master, Stock, Waiting List, Customer, Colors, Vehicle Allotment and Pictures.

□ Forms:

These are the interfaces provided for the operators, who are employees of the Showroom Automation system to perform all the operations.

Detailed Output File Formats:

- **Reports:** These provide an overview of the progress and functionality of the system.

External Interface Requirements:

User Interface: This is a GUI (Graphical User Interface) based interface.

□ Error Messages:

Invalid login and password
Conflicting entries - duplicate entry
Access denied
Wrong operation - no vehicle entry

Performance constraints:

For any input file containing more than 30 master records with more than five detailed records, the reports should be printed in less than 1 minute.

Design constraints:

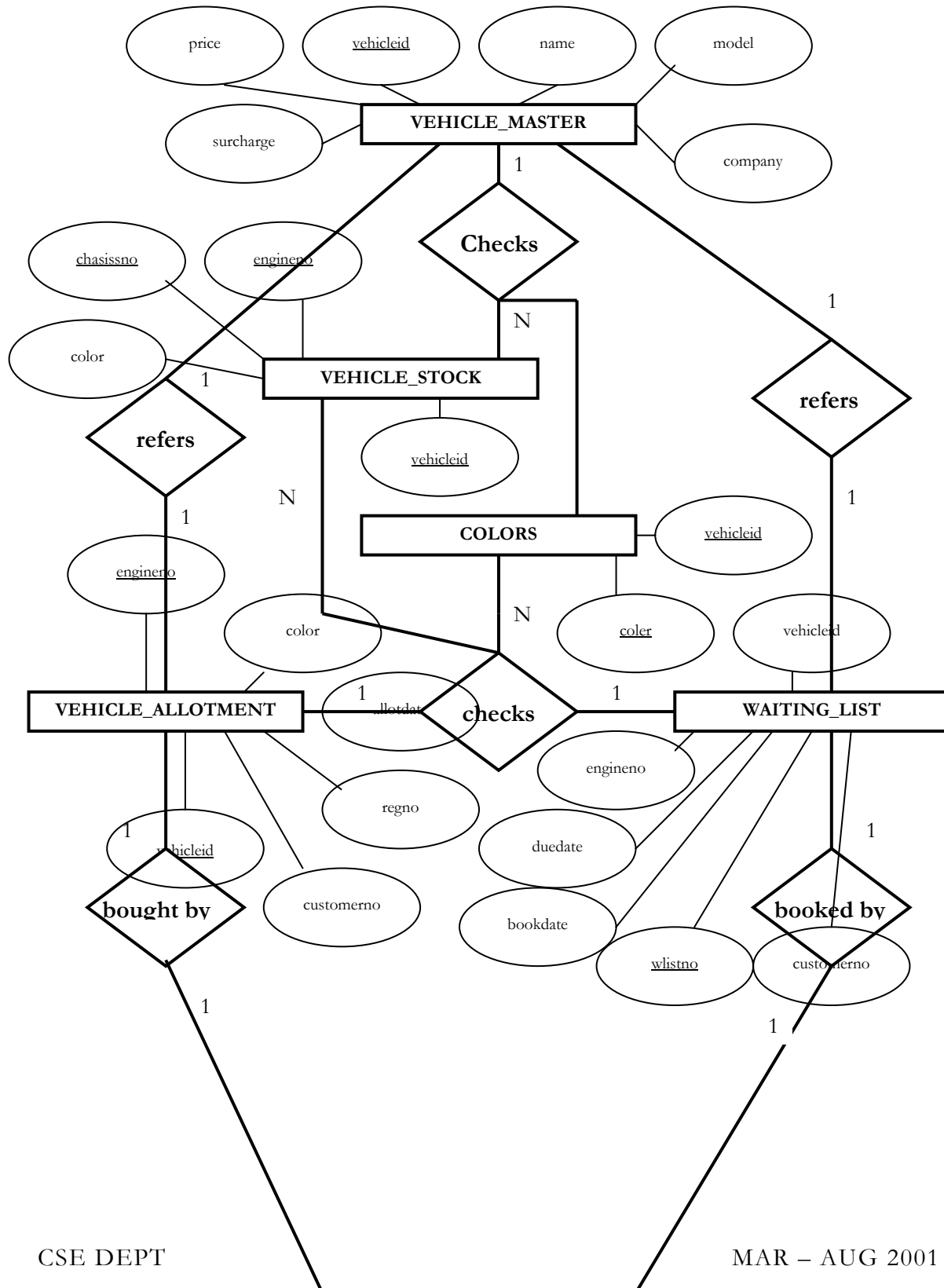
- Software constraints: Runs on a Windows 95 or above platform. Requires a D2K platform. ODBC to ORACLE 7.
- Hardware constraints: Needs an Intel Pentium or equivalent processor.

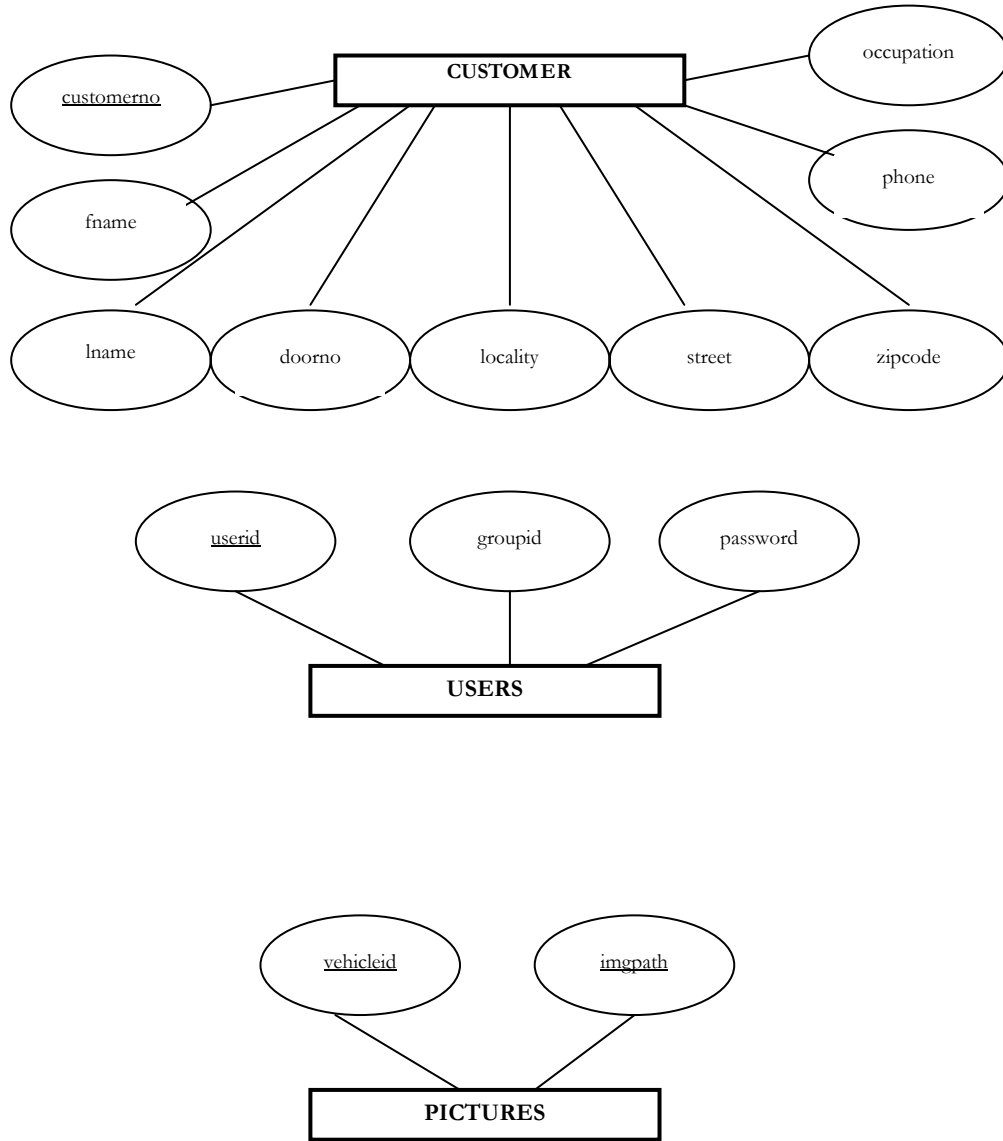
Acceptance criteria:

Before accepting this system, the developers will have to demonstrate that the system works correctly for arbitrary number of vehicles. The developers will have to show by suitable tests cases that all conditions are satisfied and all modules are working properly.



3. ENTITY RELATIONAL (E.R) DIAGRAM





4. TABLES DESCRIPTION

vehicle_master:

vehicleid	number (4)	not null
name	varchar2 (20)	not null
model	varchar2 (10)	not null
company	varchar2 (20)	not null
price	decimal (10,2)	not null
surcharges	decimal (8,2)	not null
description	varchar2 (80)	

primary key(vehicleid)
unique(name, model, company)

pictures

vehicleid	number (4)	not null
imgpath	varchar2 (80)	not null

primary key(vehicleid, imgpath)
foreign key(vehicleid) references vehicle_master(vehicleid)
unique(imgpath)

colors

vehicleid	number (4)	not null
color	varchar2 (20)	not null

primary key(vehicleid, color)
foreign key(vehicleid) references vehicle_master(vehicleid)

vehicle_stock

vehicleid number (4) not null
engineno varchar2 (20) not null
chassisno varchar2 (20) not null
color varchar2 (20) not null
primary key(vehicleid, engineno)
foreign key(vehicleid) references vehicle_master(vehicleid)
unique(chassisno)
foreign key(vehicleid, color) references colors(vehicleid, color)

customer

customerno number (10) not null
fname varchar2 (20) not null
mname varchar2 (20) not null
lname varchar2 (20) not null
doorno varchar2 (8) not null
street varchar2 (20)
locality varchar2 (20)
city varchar2 (20) not null
zipcode number (6) not null
phone1 number (10)
phone2 number (10)
occupation varchar2 (20)
primary key(customerno)

vehicle_allotment

vehicleid number (4) not null
engineno varchar2 (20) not null
chassisno varchar2 (20) not null
color varchar2 (20) not null
customerno number (10) not null
allotdate date not null
regno varchar2 (20) not null
primary key(vehicleid, engineno)
unique(regno)
foreign key(customerno) references customer(customerno)
foreign key(vehicleid) references vehicle_master(vehicleid)
unique(vehicleid, chassisno)
foreign key(vehicleid, color) references colors(vehicleid, color)

waiting_list

wlistno number (4) not null
vehicleid number (4) not null
color varchar2 (20) not null
bookdate date not null
duedate date not null
advanceamt decimal (10,2) not null
customerno number (10) not null
primary key(wlistno)
foreign key(vehicleid) references vehicle_master(vehicleid)
foreign key(customerno) references customer(customerno)
foreign key(vehicleid, color) references colors(vehicleid, color)

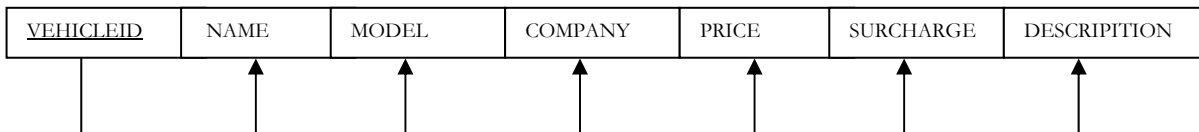
users

userid varchar2(20) not null
password varchar2(20)
groupid varchar2(20) not null
primary key(userid)

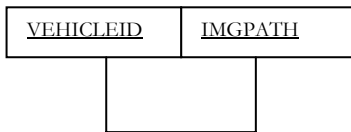


5. DATABASE SCHEMA

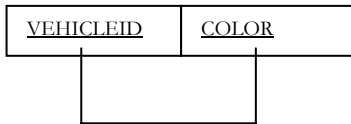
VEHICLE MASTER



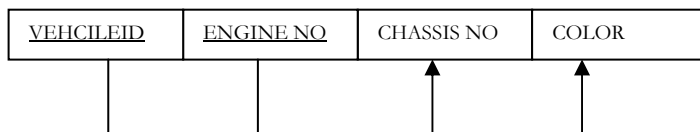
PICTURES



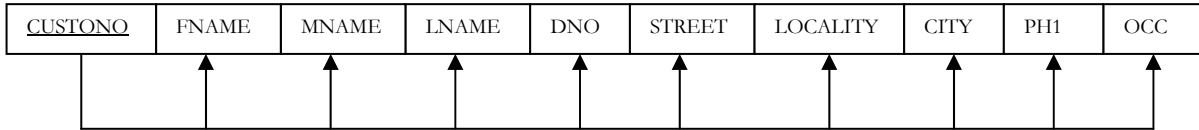
COLOR



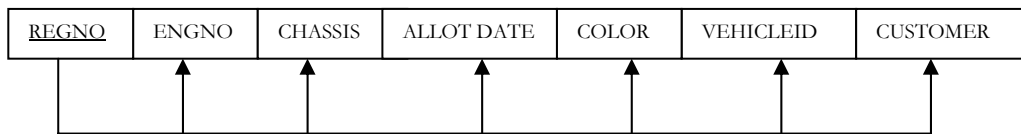
VEHICLE STOCK



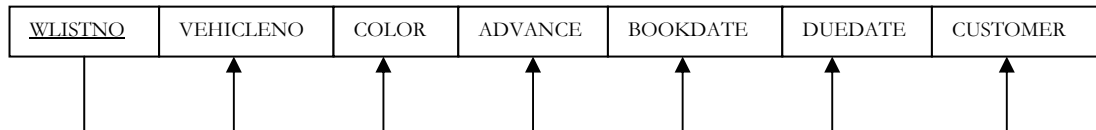
CUSTOMER



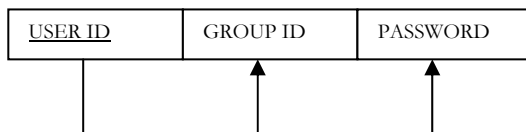
ALLOTMENT



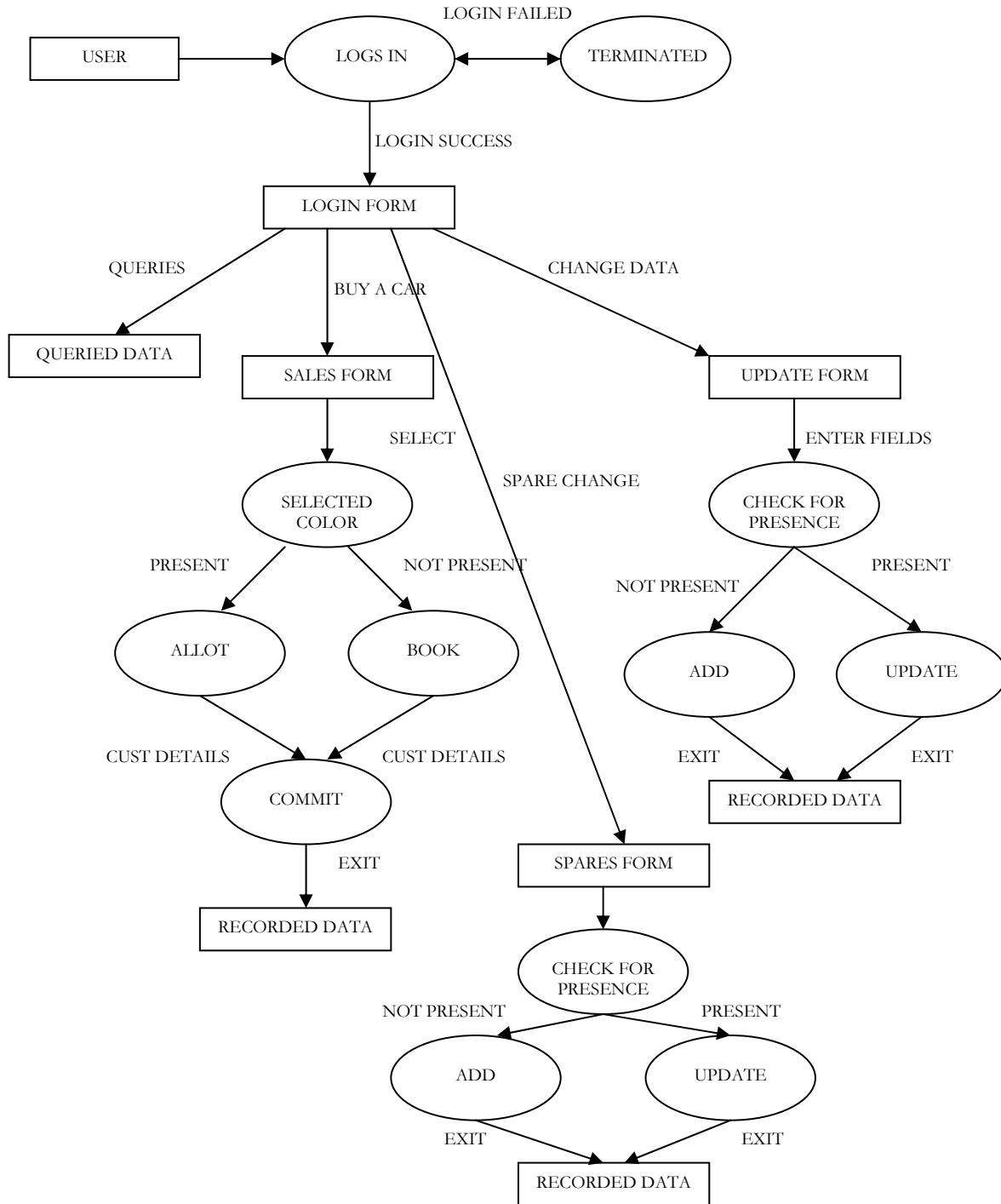
WAITING LIST



USERS



DATA FLOW DIAGRAM (D.F.D)



6. FORMS AND REPORTS

1. FORMS

SCHEMATIC REPRESENTATION :

- ❑ LOGIN

- ❑ MASTER

- ❑ ALLOTMENT

- ❑ CUSTOMER

- ❑ SALES

- ❑ QUERY

- ❑ USERS

DESCRIPTION OF TRIGGERS IN ALL THE FORMS

2. REPORTS

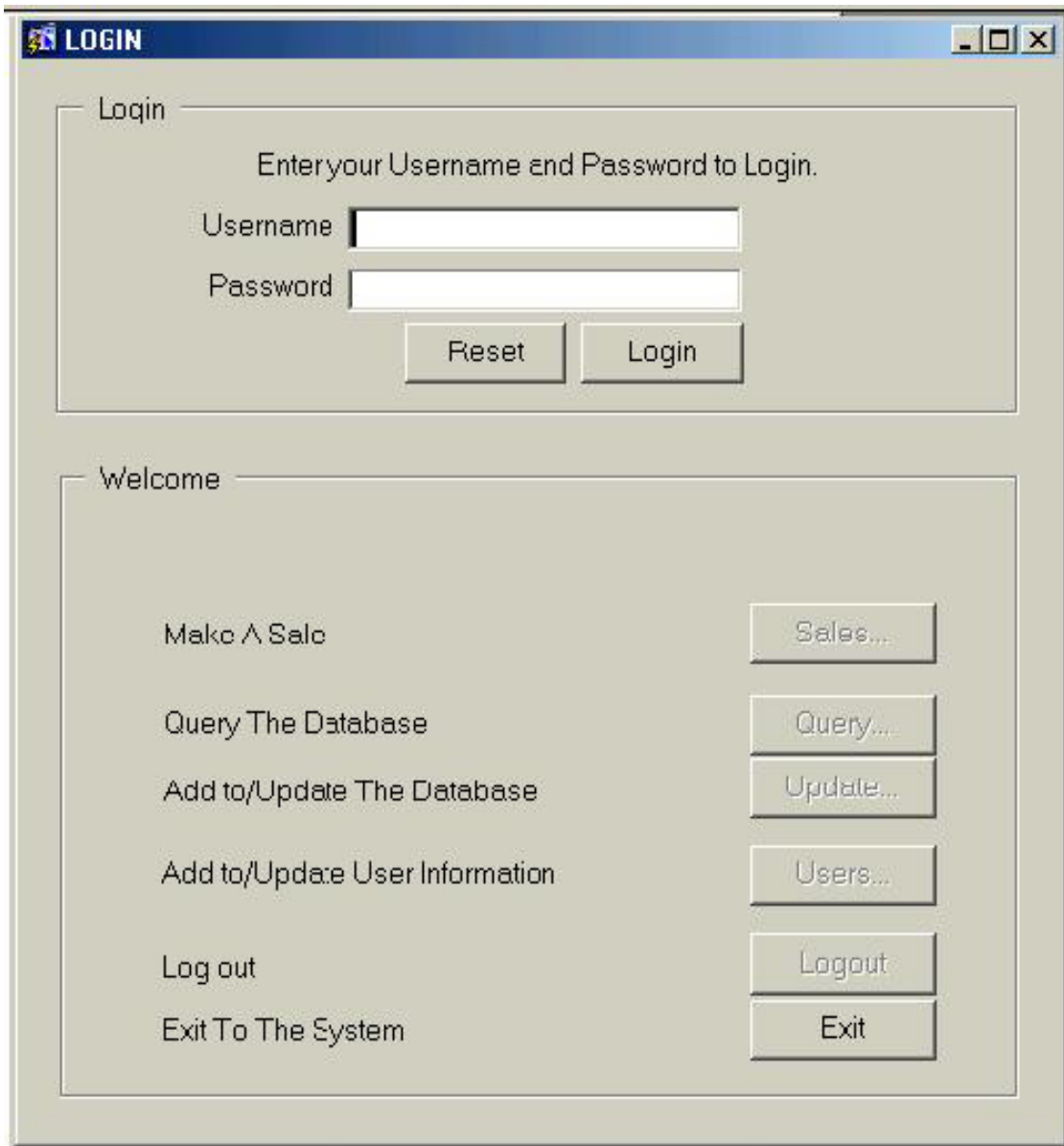
DESCRIPTION OF REPORTS WITH PARAMETERS

USER SCREENS

The screenshot shows a window titled "Vehicle Master" with a menu bar containing "Master", "Colors", "Stock", and "Pictures". The "Master" menu is currently selected. The main area contains the following elements:

- Car Name: Vehicles...
- Model:
- Company:
- Price:
- Surcharges:
- Description:
- Clear:
- New: Save: Update: Delete:
- Exit:

VEHICLE MASTER



LOGIN

The image shows a graphical user interface window titled "Query Window". It has four tabs: "Allotment", "Waiting List", "Customer", and "Vehicle Stock". The "Vehicle Stock" tab is selected. Inside the window, there is a section titled "Vehicle Stock" containing several input fields and a table. The input fields are for "Car Name", "Model", "Company", and "Total Stock". There is a "Vehicles..." button next to the "Car Name" field. Below the input fields is a table with three columns: "Engine no", "Chassis no", and "Color". The table has five rows, all of which are empty. To the right of the table is a vertical scrollbar. At the bottom of the window, there are two buttons: "Query" and "Exit".

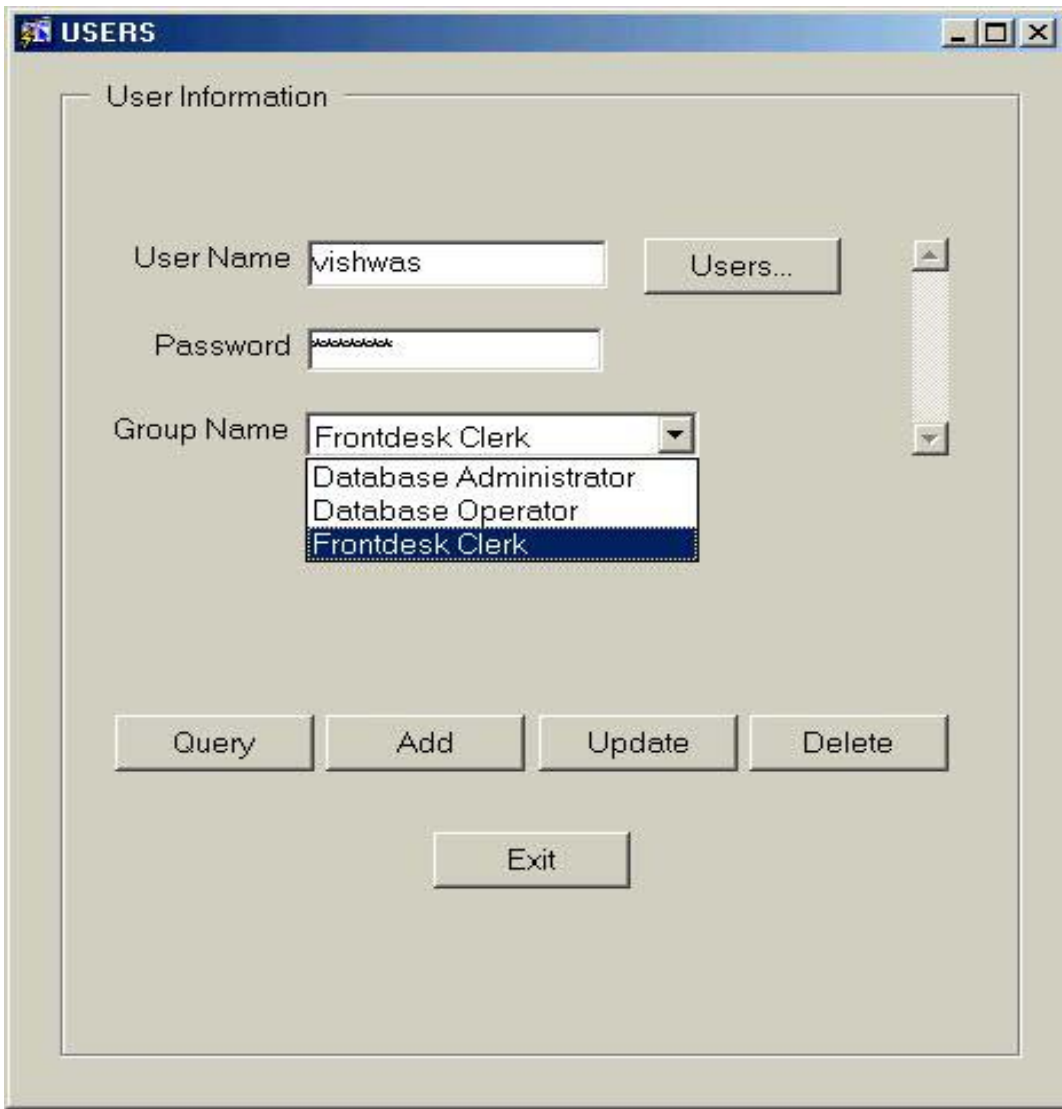
Engine no	Chassis no	Color

QUERY

The image shows a screenshot of a software application window titled "Sales". The window's main content area is titled "Vehicle Master Query". It features several input fields and buttons:

- Car Name:** A text input field.
- Model:** A text input field.
- Company:** A text input field.
- Price:** A text input field.
- Surcharges:** A text input field.
- Description:** A larger text input field.
- Vehicles...:** A button with a dotted border, located to the right of the Car Name field.
- Pictures...:** A button located below the Description field.
- Colors...:** A button located to the right of the Pictures... button.
- Allot Vehicle...:** A button located below the Pictures... button.
- Book Vehicle...:** A button located below the Allot Vehicle... button.
- Exit:** A button located to the right of the Book Vehicle... button.

SALES



USERS

REPORT SCREENS

Vehicle Master Report

Report run on: July 12, 2001 9:48 PM

Car Name	Model	Company	Price	Surcharges	Description
Corvette	SX	Chevrolet	38060	3	Convertible
Diablo	SV	Lamborghini	239900	5	Spoilers, Alloy Wheels
911 Turbo	Deluxe	Porsche	132000	4	Alloy Wheels and Sun Roof
F1	GTR	McLaren	500000	6	Carbon Fibre Body
F50	XLi	Ferrari	500000	3	Open Top
CLK-GTR	GV	Mercedes	450000	6	

Count: 6

VEHICLE MASTER REPORT

Vehicle Stock Report

Report run on: July 12, 2001 9:57 PM

Name	911 Turbo	Model	Deluxe
Company	Porsche		
Color	Glacier White		
Engine		Chassis	
D7686GHJ		VJ89TGYU6	
1			
Color	Mirage Metallic		
Engine		Chassis	
98GHJ99VGH		H8R5JK976F	
1			
Color	Ocean Jade		
Engine		Chassis	
8GBHJP78T		VG9M899YGB	
FEFFF3		DFSFSDF	
2			
Color	Pastel Yellow		
Engine		Chassis	
GG9H08UGF7		VBUQJ997GU	
1			
Color	Zenith Blue Metallic		
Engine		Chassis	
O076787FFJK		G09TYGH9GH	
1			
Count:	6		

VEHICLE STOCK REPORT

6.1 DESCRIPTION

COMMON TRIGGERS:

PRE-FORM:

```
SET_WINDOW_PROPERTY(FORMS_MDI_WINDOW,WINDOW_STATE,MAXIMIZE);  
SET_WINDOW_PROPERTY('LOGIN',WINDOW_STATE,MAXIMIZE);
```

The above trigger maximizes the window of any form opened.

FORM – ALLOTMENT

TRIGGER : WHEN-BUTTON-PRESSED

KEYS:

CANCEL :

```
exit_form (NO_COMMIT);  
Exits from the form
```

CONFIRM:

```
declare ret number;  
begin  
delete from vehicle_stock  
where vehicleid = :vehicle_allotment.vehicleid  
and engineno = :vehicle_allotment.engineno;  
commit_form;  
if (form_success) then  
ret := show_alert('allot_alert');  
else  
message('Allotment error.');
```

end if;
exit_form(NO_COMMIT);
end;

Deletes entry from vehicle stock and adds it to allotment

FORM – LOGIN

TRIGGER:WHEN-BUTTON-PRESSED

KEYS:

RESET:

```
go_field('password');
clear_item;
go_field('username');
clear_item;
Clears the items.
```

LOGIN:

declare

```
username users.userid%otype;
passwd users.password%otype;
groupname users.groupid%otype;
cnt number;
```

begin

```
username := :users.userid;
```

```
select count(*) into cnt
from users
where userid = :users.userid;
```

```
if cnt = 0 then
```

```
clear_form(NO_COMMIT);
:users.loginm := 'Invalid Username.';
go_field('users.userid');
```

```
else
```

```
select password, groupid into passwd, groupname
from users
where userid = :users.userid;
```

```
if passwd = :users.password then
clear_form(NO_COMMIT);
```

```
:users.loginm := 'Login Successful.';
```

```
:users.welcomem := 'Welcome ' || username || '. Select one of the options
below to continue.';
```

```
set_item_property('users.logout', ENABLED, PROPERTY_TRUE);
```

```
if groupname = 'administrator' then
```

```
set_item_property('users.sales', ENABLED, PROPERTY_TRUE);
```

```
set_item_property('users.query', ENABLED, PROPERTY_TRUE);
```

```
set_item_property('users.update', ENABLED, PROPERTY_TRUE);
```

```
set_item_property('users.users', ENABLED, PROPERTY_TRUE);
```

```
elsif groupname = 'operator' then
```

```
set_item_property('users.sales', ENABLED, PROPERTY_TRUE);
```

```
set_item_property('users.query', ENABLED, PROPERTY_TRUE);
```

```
set_item_property('users.update', ENABLED, PROPERTY_TRUE);
```

```
        elsif groupname = 'clerk' then
            set_item_property('users.sales', ENABLED, PROPERTY_TRUE);
            set_item_property('users.query', ENABLED, PROPERTY_TRUE);
        end if;
        go_field('users.sales');
    else
        :users.loginm := 'Invalid Password.';
        go_field('users.password'); clear_item;
    end if;
end if;

end;
```

Checks to see if the login name and password is valid and selects only the permissible buttons based on the groupid.

```
SALES :
    call_form('e:\sources\dbms\sales.fmx');
QUERY:
    call_form('e:\sources\dbms\query.fmx');
UPDATE:
    call_form('e:\sources\dbms\master.fmx');
USERS:
    call_form('e:\sources\dbms\users.fmx');
    Calls the respective form.
```

```
LOGOUT:
set_item_property('users.sales', ENABLED, PROPERTY_FALSE);
set_item_property('users.query', ENABLED, PROPERTY_FALSE);
set_item_property('users.update', ENABLED, PROPERTY_FALSE);
set_item_property('users.users', ENABLED, PROPERTY_FALSE);
go_field('users.userid');
set_item_property('users.logout', ENABLED, PROPERTY_FALSE);
:users.welcomem := ' ';
:users.loginm := 'Enter your Username and Password to Login.';
    Resets the login fields and user must login again.
```

```
EXIT:
exit_form(NO_COMMIT);
Exits
```

FORM – CUSTOMER

TRIGGER : WHEN-BUTTON-PRESSED

KEYS:

CANCEL:

```
exit_form(NO_COMMIT);  
    Exits without prompting.
```

CLEAR:

```
clear_form(NO_COMMIT);  
go_field('customer.fname');  
    Clears all the fields.
```

NEXT:

```
begin  
    :global.customerno := :customer.customerno;  
    exit_form(DO_COMMIT);  
end;  
    Writes into the customer table and assigns the global id.
```

FORM – MASTER

TRIGGER : WHEN-BUTTON-PRESSED
FOR ALL THE TAB PAGES

KEYS:

CLEAR:

```
clear_block(NO_COMMIT);  
    Clears
```

NEW:

```
clear_form(NO_COMMIT);  
go_field('vehicle_master.vehicles');  
    Readies the form for fresh input.
```

SAVE:

```
begin  
    select max(vehicleid) into :vehicle_master.vehicleid from vehicle_master;  
    if (:vehicle_master.vehicleid is null) then  
        :vehicle_master.vehicleid := 0;  
    end if;  
    :vehicle_master.vehicleid := :vehicle_master.vehicleid + 1;  
    commit_form;
```

```
go_field('vehicle_master.vehicles');  
end;
```

Inserts into the database.

UPDATE:

```
begin  
update vehicle_master  
set vehicle_master.price=:vehicle_master.price,  
vehicle_master.surcharges=:vehicle_master.surcharges,  
vehicle_master.description=:vehicle_master.description  
where vehicle_master.vehicleid = :vehicle_master.vehicleid;  
commit;  
if (form_success) then  
message('Update Successful.');
```

```
else
```

```
message('Update Error.');
```

```
end if;
```

```
go_field('vehicle_master.vehicles');
```

```
end;
```

Updates the record in the database.

DELETE:

```
begin  
delete from vehicle_master  
where vehicleid = :vehicle_master.vehicleid;  
commit;  
if (form_success) then  
message('Delete Successful.');
```

```
else
```

```
message('Delete Error.');
```

```
end if;
```

```
clear_block(NO_COMMIT);
```

```
go_field('vehicle_master.vehicles');
```

```
end;
```

Deletes the record in the database.

EXIT:

```
exit_form(NO_COMMIT);
```

Exits to the calling form.

FORM – QUERY

TRIGGER :WHEN-BUTTON-PRESSED
FOR ALL TAB PAGES

KEYS:

QUERY:

```
begin
if (:System.Mode = 'NORMAL') then
    do_key('Enter_Query');
else
    do_key('Execute_Query');
end if;
end;
    Enters into query mode and executes query.
EXIT:
exit_form(NO_COMMIT);
    Exits
```

FORM – SALES

TRIGGER: WHEN-BUTTON-PRESSED

KEYS:

VEHICLES:

```
declare
    lov_ret boolean;
    rg RecordGroup;
    ret Number;
    v_color varchar2(20);
begin
    set_item_property('colors.allot', ENABLED, PROPERTY_FALSE);
    set_item_property('colors.book', ENABLED, PROPERTY_FALSE);
    set_item_property('colors.colors', ENABLED, PROPERTY_FALSE);
    set_item_property('vehicle_master.pictures', ENABLED, PROPERTY_FALSE);
    lov_ret := show_lov('master_lov');
    if (lov_ret = TRUE) then
        set_item_property('colors.colors', ENABLED, PROPERTY_TRUE);
        set_item_property('vehicle_master.pictures', ENABLED, PROPERTY_TRUE);
        go_field('colors.colors');
    else
```

```
        go_field('vehicle_master.vehicles');  
    end if;  
end;
```

Displays the lov and the vehicle id of the selected vehicle is returned.

PICTURES:

```
:global.vehicleid := :vehicle_master.vehicleid;  
go_block('pictures');
```

Goes to the pictures form and displays the form.

COLORS:

```
declare
```

```
    ret boolean;  
    stock number;
```

```
begin
```

```
    set_item_property('colors.allot', ENABLED, PROPERTY_FALSE);  
    set_item_property('colors.book', ENABLED, PROPERTY_FALSE);  
    ret := show_lov('color_lov');
```

```
    if (ret = TRUE) then  
        :global.vehicleid := :vehicle_master.vehicleid;  
        :global.color := :colors.color;
```

```
        select count(*) into stock  
        from vehicle_stock  
        where vehicleid = :vehicle_master.vehicleid  
        and color = :colors.color;
```

```
        if (stock > 0) then  
            set_item_property('colors.allot', ENABLED, PROPERTY_TRUE);  
        elsif (stock = 0) then  
            set_item_property('colors.book', ENABLED, PROPERTY_TRUE);  
        end if;
```

```
    else  
        go_field('colors.colors');  
    end if;
```

```
end;
```

Selects the color available for selected vehicle.

ALLOT:

```
call_form('e:\sources\dbms\customer.fmx');  
call_form('e:\sources\dbms\allotment.fmx');  
go_field('vehicle_master.vehicles');  
    Calls the customer form and allotment.
```

BOOK:

```
call_form('e:\sources\dbms\customer.fmx');  
call_form('e:\sources\dbms\wlist.fmx');  
go_field('vehicle_master.vehicles');  
    Calls the customer and wlist form.
```

FORM – USERS

TRIGGERS:WHEN-BUTTON-PRESSED

KEYS: COMMON KEYS ARE NOT EXPLAINED

USERS:

```
declare  
    lov_ret boolean;  
begin  
    lov_ret := show_lov('user_lov');  
    if (lov_ret = TRUE) then  
        go_field('users.password');  
    else  
        go_field('users.users');  
    end if;  
end;  
    Adds new user.
```

TRIGGER:WHEN-VALIDATE-ITEM

LIST :

GROUP NAME::

```
declare  
--  
-- Begin default enforce data integrity constraint SYS_C001070 section  
--  
if not( :USERS.GROUPID IS NOT NULL ) then  
    message( 'WHEN-VALIDATE-ITEM trigger failed on field - ' || :system.trigger_field );  
    raise form_trigger_failure;  
end if;
```

FORM – WLIST

TRIGGERS: WHEN-BUTTON-PRESSED

KEYS: COMMON KEYS NOT EXPLAINED

CONFIRM:

```
declare
    ret number;
begin
    commit_form;
    if (form_success) then
        ret := show_alert('book_alert');
    else
        message('Could not enter into waiting list.');
```

end if;
exit_form(NO_COMMIT);
end;

Adds an entry into waiting list if successful.

REPORT QUERIES

Report Query Statements – Given below is a listing of the SQL Query statements which have been used to generate the Reports when requested by the operator.

- ❑ Vehicle Allotment Report – This Report contains a listing of the allotments made to various customers. This report is a parametric, and hence can generate allotment reports for specific vehicles also. The SQL Query statement for generating the report is given below.

```
select * from vehicle_allotment, vehicle_master, customer where
(vehicle_master.vehicleid = vehicle_allotment.vehicleid and customer.customerno =
vehicle_allotment.customerno and name like :nm and model like :mod and company like
:comp) or (vehicle_master.vehicleid = vehicle_allotment.vehicleid and
customer.customerno = vehicle_allotment.customerno and :nm is null and :mod is null
and :comp is null and regno is not null)
```

- ❑ Colors Report – This report contains information about the various colors available for every automobile type. The SQL statement for used for generating this report is –

```
select * from vehicle_master, colors where colors.vehicleid =  
vehicle_master.vehicleid
```

- ❑ Customer Report – This report contains the details of the customers, including their name, address and contact information. The SQL statement for this report is:

```
select * from customer
```

- ❑ Vehicle Master Report – This report contains information about the different vehicle models available for sale. The SQL query statement is given below.

```
select * from vehicle_master
```

- ❑ Vehicle Stock Report – The vehicle stock report contains a complete listing of the current stock of automobiles in the showroom at the time of generation of the report. The SQL query statement for this report is –

```
select * from vehicle_master, vehicle_stock where vehicle_stock.vehicleid =  
vehicle_master.vehicleid
```

- ❑ Waiting List Report – The waiting list report produces a list of the booking orders at the showroom. It contains information about the customer who has booked a vehicle, and the details of the corresponding vehicle. The report is grouped by the due date of the booking order. The SQL statement for generating the report is shown below.

```
select * from vehicle_master, customer, waiting_list  
where vehicle_master.vehicleid = waiting_list.vehicleid  
and customer.customerno = waiting_list.customerno order  
by duedate
```



7. CONCLUSION

The merits of the system are:

Generates report on details for maintaining customer information, existing vehicles stock information, spares management, user information, information about vehicle allotment and providing waiting list information.

Provides secure access and high information integrity among various base tables.

Administrator, manager and clerk can change/add information and details of records.



8. BIBLIOGRAPHY

- FUNDAMENTALS OF DATABASE SYSTEMS.
- -ELMASRI / NAVATHE
- ORACLE DEVELOPER.
- - IVAN BAYROSS
- SQL INSTALLATION AND USER GUIDE.

