

# INTRODUCTION

**THE** success and the longevity of any prominent business nowadays depend on powerful application infrastructures and effective, reliable management and development solutions.

In present, a wide range of IT companies are striving to provide business owners with better application management and new, state-of-the-art development solutions and tools, vital for increasing productivity and facilitating process development for any business. In order to improve the overall performance of commonly used application servers, and to enhance the efficiency of specific application programs, more and more IT companies are currently aiming to develop new, competitive software programs that support popular, commonly used databases. Such software products offer an extended set of features that help businesses implement better management and development solutions and also considerably reduce the time, effort and costs of process development.

Most software products implement support for My SQL databases, providing users with a complete set of improved administration and development solutions. Enhancing and correcting the features provided by the popular My SQL Manager application, My SQL Turbo Manager is by far

the best software solution for My SQL database management, administration and development in present. However, the standard My SQL Manager application can't effectively meet today's requirements, users needing more powerful and complex applications for creating, managing and interacting with various databases.

**"QUERY BROWSER"** for Oracle is a powerful Integrated Development Environment (IDE) for Oracle Databases. Using the intuitive "QUERY BROWSER" GUI will increase your code quality and reduce the development process time.

**"QUERY BROWSER"** for Oracle easily enables you to build and execute queries, build and execute scripts, compile and debug SQL code, create and modify database objects, import and export data, enable and database monitoring among other features.

The computer has change the way we work learn and communicate virtually every kind of organization throughout the world conducts business with computers. For many organizations computer based information system are now at the heart of daily activities and a major consideration in corporate decision-making.

Most software products implement support for My SQL databases, providing users with a complete set of improved administration and development solutions. Enhancing and correcting the features provided by the popular My SQL Manager application, My SQL Turbo Manager is by far the best software solution for My SQL database management, administration and development in present. However, the standard My SQL Manager application can't effectively meet today's requirements, users needing more powerful and complex applications for creating, managing and interacting with various databases.

The project report for this system provides a comprehensive view of the overall system and various steps involved in the process of its development. The project report is divided into number of sections. The very first section describes the existing system, their problems and the proposed system.

The second section gives detail information of all database design process. The structure of the database that is designed in ORACLE is discussed in detail with the help of data dictionary and other diagrams. Various requirements and constraints have also been discussed. The biggest portion of this report is dedicated to the interface design. The interface design of the system is Graphical User Interface (GUI). The interface is designed in to Microsoft visual basic.net 2005. The structure of the interface

is designed to provide several facilities along with the process for data storage, easy maintenance and retrieval of information, reports generation and a mean for analysis of the system outputs for better performance. The entire system is divided into several modules driven through a menu driven application. The other important section is to provide reports needed by the user.

**"QUERY BROWSER"** is a small software package that helps in managing the data. It provides a simple way to effectively store, retrieve and update the data without writing a single code. A user can easily create his own databank and store his/her data in Tabular form. As it works in windows environment it is user-Friendly Software. One can visually store, modify, search as well as locate one's own data. The key goal behind developing this software is to provide an easy and fast method of creating database and performing various operations on it. As all users are not computer professionals it prevents from illegal access of data by unauthorized user, by assigning a password to each databank. So that security could be maintained.

# OBJECTIVES

The objective of the project is to resolve the problems occurring in the existing system and providing a visual solution.

## WHY "QUERY BROWSER"?

- ✚ Easy to use customizable interface makes your work comfortable and productive.
- ✚ Multivendor environment support. You can work with different data sources simultaneously.
- ✚ Plug-ins support and set of free plug-ins will add flexibility to your work with this tool.
- ✚ Powerful built-in schema tools (browser, export, snapshots, and search).
- ✚ **Atomicity:-** each transaction is a separate and distinct entity. It happens in whole or not at all.
- ✚ **Consistency:-** each database change is partitioned from other transactions. Results from another transaction won't mysteriously pop up in the middle of another transaction. One transaction moves the database from one consistent state to the next.

🏠 **Isolation:-** A transaction is not available to other users until it has been committed. The results of a transaction are invisible to other transactions until the transaction is complete thus increasing the security on data.

🏠 **Durability:-** Once committed (completed), the results of a transaction are permanent and survive future system and media failures and thus ensuring maintenance and protection of data.

Advanced **"QUERY BROWSER"** makes your work with script easy and comfortable.

# **BACKGROUND**

**"QUERY BROWSER"** for any **database** is a powerful Integrated Development Environment (IDE) for any Databases. Using the intuitive QUERY BROWSER's GUI will increase your code quality and reduce the development process time.

## **"QUERY BROWSER" feature list:**

- 🌈 Generates data that perfectly matches to your database schema.
- 🌈 Intuitive, easy to use user interface makes it easy to start populating database immediately.
- 🌈 Intelligently generates appropriate data based on table and column names and data types.
- 🌈 Over 60 built in generators with sensible configuration option.
- 🌈 Inter-column dependency support, for example to generate record with matching email address and domain name.

## **TOOLS USED IN THE PROJECT**



# SURVEY OF TOOLS

## Software Tools :-

### Front-end Tool :



In this project “**JAVA**” has been chosen as the Front-end tool for the process of software development.

**Selection of “JAVA” as a front-end tool is done on the following facts:-**

The JDBC API is designed to allow developers to create JAVA code that can access almost any relational database without needing to continually rewrite their code. Despite standards set by the ANSI committee, each database system vendor has a unique way of connecting to its system. Any type of JAVA code, including JAVA Servlets, JSP Pages, Enterprise JAVA Beans and plain JAVA classes can use JDBC.

## **Java Database Connectivity (JDBC)**

The JDBC API is designed to allow developers to create JAVA code that can access almost any relational database without needing to continually rewrite their code. Despite standards set by the ANSI committee, each database system vendor has a unique way of connecting to its system. Any type of JAVA code, including JAVA Servlets, JSP Pages, Enterprise JAVA Beans and plain JAVA classes can use JDBC.

The JDBC API was first introduced with release 1.1 of JDK. JDK 1.5 contains JDBC 3.0 which is composed of the JAVA.SQL and JAVAX.SQL packages. JDBC provides application developers with a single API that is uniform and database independent. The API provides a standard to write to, as well as standard that takes all of the various application engines into account.

The API database independence is due to set of JAVA interfaces that are implemented by a driver. The driver takes care of translating the standard JDBC calls into the specific calls required by the database it supports. The application is written once and then moved to the various drivers. The application remains the same; the driver change. Driver may be used to develop the middle tier of a multi-tier database design.

In addition to providing developers with a uniform and database independent framework.

JDBC also provides a means of allowing developers to retain the specific functionality that their database vendor offers.

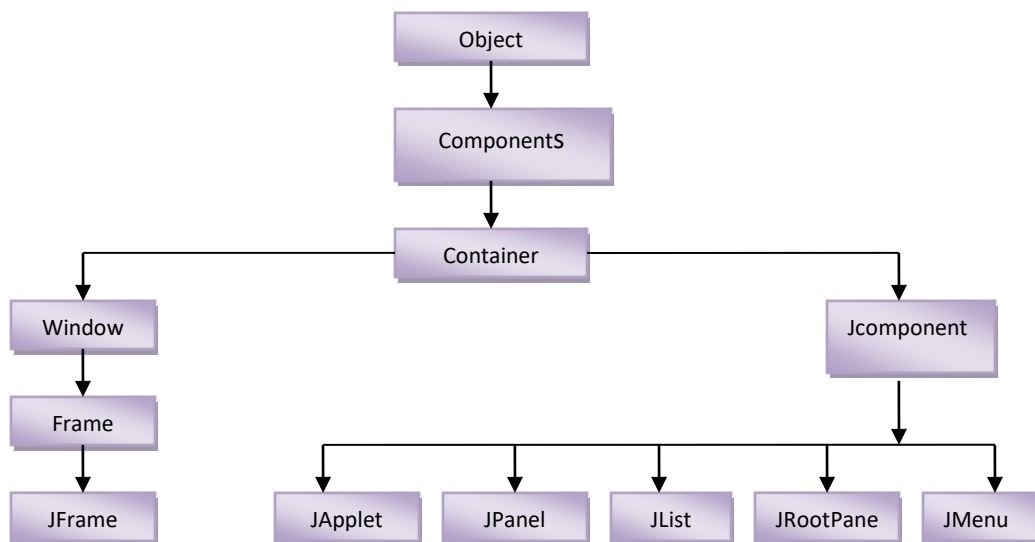
JDBC drivers must support ANSI SQL-2 entry level, but JDBC allows to developers passing query strings directly to the connected driver. These strings may or may not be ANSI SQL or SQL at all. The use of these strings is up to underlying driver.

JDBC is not a derivative of Microsoft's open database connectivity (ODBC) specification. **JDBC is written entirely in JAVA; ODBC is a 'C' interface.** Both JDBC and ODBC however are based on the X/open SQL command level interface (CLI). **This Javasoft provides a JDBC-ODBC bridges that translate JDBC to ODBC implementation alone with native methods is very small and efficient.**

### **JAVA SWING JFC**

**Swing** is a set of packages built on the top of the AWT that provides you with a great number of prebuilt classes (over 250 classes and 40 UI components).

The **Swing API** has many classes for creating various types of user interface elements. The swing family tree is spilt in such a way that the component class is divided into two – the group of the classes derived from **Jcomponents** class and another that descends from the **Window classes**.



### The Swing Family Trees

#### Swings Features:-

- ❖ Borders
- ❖ Graphics debugging
- ❖ Easy mouseless operation
- ❖ Tooltips
- ❖ Easy Scrolling
- ❖ Pluggable look and feel
- ❖ New layout managers

## **Reporting Tool : HTML**

**HTML is a language for describing how pages of text, graphics and other information are organized, formatted and linked together.**

By 1993, almost 100 computers throughout the world were equipped to serve up HTML pages. Those interlinked pages were dubbed the World Wide Web (WWW), and several web browser programs had been written to allow people to view web pages. Because the popularity of the Web, a few programmers was Mare Andresen; he went on to become rich and famous, selling one of the world's most popular Web browser, Netscape Navigator.

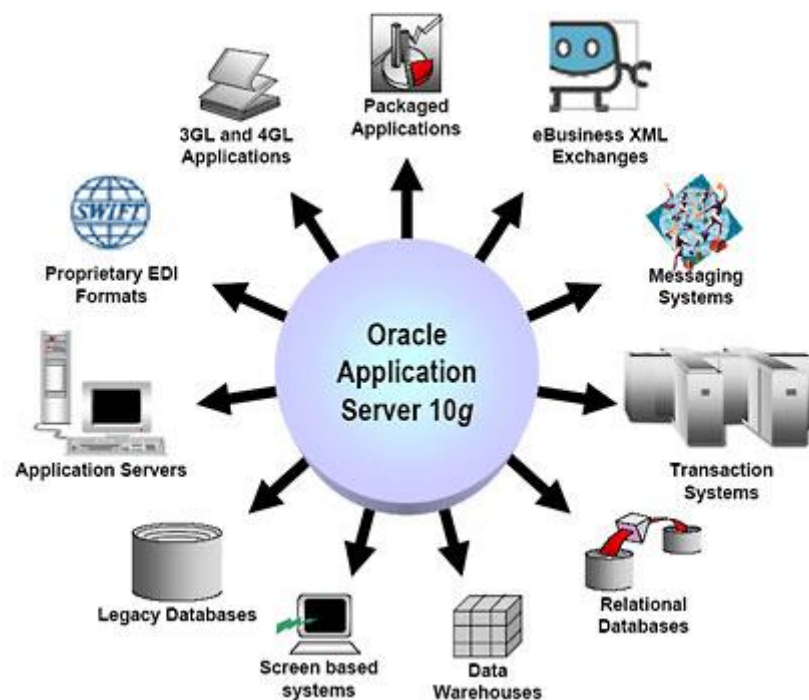
Today HTML pages are the standard interface to the internet. They can include animated graphics, sound and video complete interactive programs, and good old fashioned text. Millions of Web pages are retrieve each day from thousand of Web server computers around the world.

The Web is on the verge of becoming a mass-market medium, as high speed internet connection through TV cables, modernized phone lines and direct satellite feeds become common place.

## **ORACLE10g As a Back-end:**

Oracle, on the other hand, is the most prevalent Relational Database Management System (RDBMS).

The Oracle-8i database runs on every major machine and under every prominent operating system available. Versions of Oracle exist for platforms from modestly equipped portable computers up to powerful database servers accommodating thousands of concurrent users. Oracle-8i databases act as on-line transaction processing engines; decision-support systems etc. The prevalence and flexibility of ORACLE 10g have given rise to groups of IT professionals.



- ✚ Software houses took up the challenge of designing a system that would help users in managing database. These systems were called Database Management Systems (DBMS). One of the DBMS developed by software houses was Oracle-8i.
- ✚ Oracle is a company that produces server based, Multi-user RDBMS. It works in a client / server environment .It has a suite of products via which commercial application are created. Oracle provides server that is used essentially for data storage. It has a suite of tools that are being used to capture. Validate and display data according to user and systems needs. It also allows manipulation of data
- ✚ Oracle-8i provides extensive security features in order to safeguard information in its tables from an unauthorized viewing and damage. Depending on a user's status and responsibility appropriate rights on oracle's resources can be assigned to the user.
- ✚ Objects that are created by a user are owned and controlled by that user. If a user wishes to access any of the objects belonging to another user, the owner of the object will have to give permissions for such access.
- ✚ The owner of the object can take privileges once given. Oracle-8i provides an interactive SQL tool, which allows users to enter ANSI SQL sentences and pass them to the Oracle Engine for execution. These sentences allow the user to create access and maintain data structures like tables, indexes etc.
- ✚ Oracle-8i provides fast access to SQL even when there are thousands of data.



- ✚ Oracle-8i expands significantly the use of triggers to administer a database and publish information about events taking place within the database

**I preferred to use Oracle-8i backend because of the above stated advantages of Oracle and also because the Oracle-8i server's primary job is to manage data optimally among multiple users that concurrently request for the same data. Access to data, in the data storage system is always via the Oracle-8i server. This gives a single entry point to access of data, which in turn has the capacity of offering excellent data security. I also chose Oracle-8i as backend as I have worked on it.**

#### ✚ **Client/server environments (distributed processing):-**

To take full advantage of a given computer system or network, Oracle allows processing to be split between the database server and the client application programs. The computer running the database management system handles all of the **Oracle** server responsibilities while the workstations running the database application concentrate on the interpretation and display of data.

#### ✚ **Large databases and space management:-**

Oracle supports the largest of databases, which can contain terabytes of data. To make efficient use of expensive hardware devices, Oracle allows full control of space usage.

#### ✚ **Many concurrent database users:-**

Oracle supports large numbers of concurrent users executing a variety of database

applications operating on the same data. It minimizes data contention and guarantees data concurrency.

✚ **Connectability:-** Oracle software allows different types of computers and operating systems to share information across networks.

✚ **High transaction processing performance:-** Oracle maintains the preceding features with a high degree of overall system performance. Database users do not suffer from slow processing performance.

✚ **Controlled availability:-** Oracle can selectively control the availability of data, at the database level and sub-database level. For example, an administrator can disallow use of a specific application so that the application's data can be reloaded, without affecting other applications.

✚ **Manageable security:-** To protect against unauthorized database access and use, Oracle provides fail-safe security features to limit and monitor data access. These features make it easy to manage even the most complex design for data access.

✚ **Database enforced integrity:-** Oracle enforces data integrity, "business rules" that dictate the standards for acceptable data. This reduces the costs of coding and managing checks in many database applications

✚ **Portability:-** Oracle software works under different operating systems. Applications developed for Oracle can be ported to any operating system with little or no modification.

✚ **Compatibility:-** Oracle software is compatible with industry standards, including most industry standard operating systems.

Applications developed for Oracle can be used on virtually any system with little or no modification

➤ **Distributed systems:-** For networked, distributed environments, Oracle combines the data physically located on different computers into one logical database that can be accessed by all network users. Distributed systems have the same degree of user transparency and data consistency as non-distributed systems, yet receive the advantages of local database management.

➤ **Replicated environments:-** Oracle software lets you replicate groups of tables and their supporting objects to multiple sites. Oracle supports replication of both data- and schema-level changes to these sites. Oracle's flexible replication technology supports basic primary site replication as well as advanced dynamic and shared-ownership.

➤ I preferred to use Oracle backend because of the above stated advantages of Oracle and also because the Oracle server's primary job is to manage data optimally among multiple users that concurrently request for the same data. Access to data, in the data storage system is always via the Oracle server. This gives a single entry point to access of data, which in turn has the capacity of offering excellent data security. I also chose Oracle as backend as I have worked on it.

# **SYSTEM ANALYSIS**

# **SYSTEM ANALYSIS**

The main purpose of conducting system analysis is to study the various processes and to find out its requirements. These may include ways of capturing or processing data, producing information, controlling an activity or supporting management. The determination of requirements entails studying the existing details about it to find out what these requirements are.

System analysis has been conducted with the following objectives in mind.

- Identify the customer needs.
- Evaluate the system concept for feasibility.
- Perform economic and technical analysis.
- Allocate functions to hardware, software, people, data base, and other System elements.

System analysis includes requirements analysis. The requirement analysis task is a process of discovery, refinement, modeling, and specification. Requirement analysis allows the software engineer to refine the software allocation and build models of the data, functional and behavioral domains that will be treated by software. Requirement specification provides the developer and the customer with the means to assess quality once software

is built. While the analysis phase of development of this project following set of principles were considered:

- The information domain of a problem must be represented and understood.
- The function that the software is to be performed must be defined.
- The behavior of the software must be represented.
- The models that depict information function and behavior must be partitioned in a manner that uncovers detail in a layered fashion.
- The analysis process should move from essential information toward Implementation detail.

All the necessary details required in the analysis process of project were collected from record registers, customers, worker of the firm and the manager.

The requirement analysis played an important role in identifying what the firm wants from the proposed system. All the analysis was done in accordance to the user's requirements so that the proper design can be done and the firm requirements can be fulfilled.

# PROBLEM DEFINITION

The basis for a candidate system is recognition of a need for improving an information system or a procedure. The problem definition phase identifies the problem, studies the alternate solutions, and recommends the resources required for designing the system.

- ✚ The need for the proposed project has arisen because oracle is not a user friendly software.
- ✚ There are so many users who don't know how to make a query.
- ✚ User cannot remember large queries for longer time period.
- ✚ For getting any information user needs some help of programmers and the programmers are not available all time.
- ✚ Specially a lay man cannot operate oracle easily.
- ✚ If we have to make a database in oracle then we have to learn the oracle and all DBMS concepts.
- ✚ There is no GUI feature in oracle.
- ✚ So in many companies and organizations waste there time in teaching languages and queries.
- ✚ More money is wasted in teaching.

✍ On the basis of the above given problems the following conclusion is made:

A Proper software can solve the above given problems since it can provide the general user important information at correct time and can also help.

## **REQUIREMENT SPECIFICATION**

After collecting the requirements, requirement specification is created which specifies the need of the customer. It helps the developer and customer to access quality once software is built. If customer changes his requirements, the developer can demand for more time or additional cost or both.

The requirement specification for the proposed software is as follows:

Because of the needs specified above, it is necessary to build user friendly software, which would provide information about the database. Since it is not advisable to go through a lengthy procedure of making queries and learning languages for fetching information, deleting, updating and inserting data.

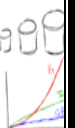
This software will help in above defined problems. User can easily do all task defined above.



The basis for a candidate system is recognition of a need for improving an information system or a procedure. The problem definition phase identifies the problem, studies the alternate solutions, and recommends the resources required for designing the system.

- ✚ The need for the proposed project has arisen because oracle is not a user friendly software.
- ✚ There are so many users who don't know how to make a query.
- ✚ User cannot remember large queries for longer time period.
- ✚ For getting any information user needs some help of programmers and the programmers are not available all time.
- ✚ Specially a lay man cannot operate oracle easily.
- ✚ If we have to make a database in oracle then we have to learn the oracle and all DBMS concepts.
- ✚ There is no GUI feature in oracle.
- ✚ So in many companies and organizations waste there time in teaching languages and queries.
- ✚ More money is wasted in teaching.

the basis of the above given problems the following conclusion is made:



Proper software can solve the above given problems since it can provide the general user important information at correct time and can also help.

### **PLANNING AND SCHEDULING**

<b>ACTION</b>	<b>DAYS</b>
<b>Synopsis Preparation</b>	<b>5</b>
<b>Tool Learning</b>	<b>10</b>
<b>Project(Analysis,Coding,Testing)</b>	<b>24</b>
<b>Documentation</b>	<b>6</b>
<b>TOTAL</b>	<b>45 days</b>

## **PRELIMINARY INVESTIGATION**

Very often when the customer put his demand for any system, the definition of problem or requirement remains incomplete. To collect the complete requirements so as to build a system according to the need of the customer, preliminary investigation is conducted. During investigation questions are asked to managers, person who will use the system. All the necessary details required in the analysis of project are collected from the record registers, customers.

Following questions are asked to the user to find out the major heading that he wants to give on the software.

- ④ Which type of information he wants to display?
- ④ What facilities he wants?
- ④ In which format he to display information?
- ④ What are the expectations from the software?
- ④ What is the information he wants to be viewed only by administrator and restricts to others?

The following fact-finding techniques are also used for preliminary investigation.

- ④ Existing documentation, forms, file and records
- ④ Observation of the work environment

## **FEASIBILITY STUDY**

## FEASIBILITY STUDY

- During the systems analysis phase, a feasibility study (also known as, a business case) is prepared and is updated during the remaining steps in the SDLC.
- The extent of the feasibility study depends on the size and nature of the system.
- Feasibility team should include:
  - Management
  - Accountants skilled in controls and auditing
  - Systems personnel
  - Users
- The feasibility study and its updates are used by the steering committee as the project proceeds to decide whether to:
  - Terminate the project
  - Proceed
  - Proceed if specific problems are resolved

Before getting started on the new system it was important to determine the feasibility involved in the developed system so that the new system development could be developed. Feasibility is the determination of whether or not a project is not worth doing. The project followed in making this determination is called a feasibility study. This type of study determines if a project can and should Be Taken. Once it has been determined that project is feasible, the analyst can go ahead and prepare the project specification, which finalizes project requirements.

Generally, feasibility studies are undertaken within tight time constraints and normally culminate in a written and feasibility report.

Five aspects need to be considered during a feasibility study:

➤ **Technical feasibility:** - Is the technology there to do it?

➤ **Operational feasibility:** -Do we have people who can do it, and will it get used?

➤ **Legal feasibility:** -Does it comply with legal, regulatory, and contractual obligations?

➤ **Scheduling feasibility:** -Can it be done in time?

➤ **Economic feasibility:** -Will the benefits exceed the costs?

## **1) Technical Feasibility: -**

**Definition:** - The process of proving that the concept is technically possible.

**Objective:** -The objective of the technical feasibility step is to confirm that the product will perform and to verify that there are no production barriers.

**Product:** - The product of this activity is a working model.

**Technical Activities:** - During the technical feasibility step the following must be completed.

- Test for technical feasibility;
- Examine the operational requirements;
- Identify potential safety and environmental hazards;
- Conduct a preliminary production feasibility assessment;
- Conduct a preliminary manufacturing assessment;
- Estimate engineering prototype costs

**Technical Information:** - The technical feasibility step generates knowledge about the product or process's design, performance, production requirements, and preliminary production costs.

### **Assessment:-**

- Do you have a working model of the product?
- Have you evaluated the safety factors of the model?
- Have you evaluated the environmental factors?
- Have you evaluated the feasibility of producing the product?
- Have you measured how the product will perform?
- Do you have a design for the product?
- Do you have a design for the production process?

As we know the technical feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirement. It is used to determine the requirements of technologies for the current system. The technical needs of the system may vary considerably, but might include:

- ✚ The facility to produce outputs in a given time.
- ✚ Response time under certain conditions.
- ✚ Ability to process a certain volume of transaction at a particular speed.
- ✚ Facility to communicate data to distinct location.



- ✚ Data should be retrieved fast and accurate.
- ✚ Technical feasibility assesses whether the current technical resources are sufficient for the new system.
- ✚ If they are not available, can they be upgraded to provide the level of technology necessary for the new system?

In examination technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system's requirements: How many workstations are required, how these units are interconnected so that they could operate and communicate smoothly.

## **2) Operational Feasibility: -**

Operational feasibility determines if the human resources are available to operate the system once it has been installed. Users that do not want a new system may prevent it from becoming operationally feasible. Proposed projects are beneficial only if they can be turned into information system that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it developed and installed. Are there major barriers to implementation? Some of the important questions that are useful to test the operational feasibility of a project are given below:

- Is there sufficient support for the project from the implementation? From user? If the present system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
- Are current business methods acceptable to the user? If they are not, user may welcome a change that will bring about a more operational and useful system.
- Have the user been involved in the planning and development of the project? If they are involved at the earliest stage of project development, the chances of resistance can be possibly reduced.
- Will the proposed system cause harm? Will it produce poorer result in any case or area? Will the performance of staff member fall down after implementation?

Issue that appears to be quite minor at the early stage can grow into major problem after implementation. Therefore, it is always advisable to consider operational aspects carefully. The system is developed in keeping mind that it should be user friendly and easy to operate hence the system is operational feasible

### **3) Economic Feasibility**

Economic feasibility is probably the most important and frequently analyzed aspect.

This examination requires a careful investigation of costs and benefits.

It typically uses a capital budgeting model that considers:

- Cost savings and other benefits
- Initial outlay costs
- Operating costs
- Other costs

An economic feasibility test/study focuses on returns and costs of a proposed plan in both the short and long-term. An economic feasibility study (EFS) should consider investment and operating costs, the time value of money, risk and uncertainty, quality of available data, and the sensitivity of assumptions. An economic feasibility study should demonstrate the net benefit of the proposed course of action in the context of direct and indirect benefits and costs to the organization and to the general public as a whole. An EFS makes a business case, prepares analytical worksheets and other necessary supporting

documentation. EFS should be required for both pilot and long-term activities, plans and projects.

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefits analysis the procedure is to determine the benefits and savings that are expected from a purposed system and compare with costs. If benefits out weigh cost, a decision is taken to design and implement the system otherwise further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an on going effort that improves in accuracy at each phase of the system life cycle. The analysis part also clears the doubt of economic problems could be possible in developing the system.

When possible, benefits and costs should be estimated and included even if they are not easily quantifiable.

If some costs and benefits cannot be accurately estimated, they should at least be listed, along with the likelihood of their occurrence and their expected impact .

# **PROJECT PLANNING AND SCHEDULING**

# **PROJECT PLANNING**

## **Objectives:-**

The objective of the system can be defined here as:

- The proposed system should be able to read the data from oracle files and store validated data in the database.
- easy to manage large amount of data.
- data can be managed effectively.

Specific products to be delivered

The product that will be delivered:

- The tested system and network.
- Client workstation.
- A robust data management server

Activities and milestones

The activities in the system, after including the provisions for security are:

- verifications of software
- migration of the software
- validation of the migrated data

**The milestones in the system are:-**

- synopsis preparation: 07-FEB-2013—18-FEB-2013
- tool learning : 19-FEB-2013—21-MAR-2013
- analysis : 22-MAR-2013—05-APR-2013
- design : 06-APR-2013—26-APR-2013
- coding : 27-APR-2013—03-MAY-2013
- testing : 04-MAY-2013—21-MAY-2013
- documentation : 22-MAY-2013—24-MAY-2013

## **SCHEDULING**

Scheduling of a software project can be correlated to prioritizing various tasks (jobs) with respect to their cost, time, and duration. Scheduling can be done with resource constraint or time constraint in mind. Depending upon the project, scheduling methods can be static or dynamic in implementation.

Scheduling techniques

There are various types of scheduling techniques in software engineering:

### **WORK BREAKDOWN STRUCTURE:-**

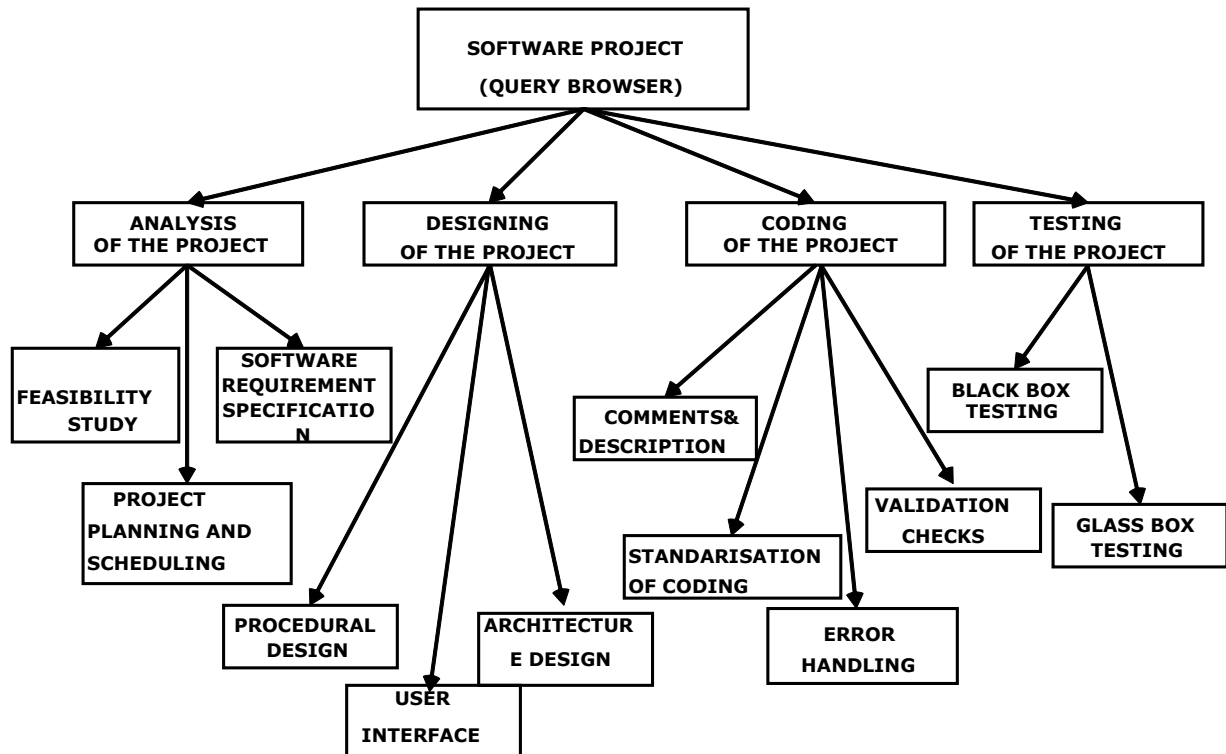
The project is scheduled in various phases following a top-down approach. A tree like structure is followed without any loops. At each phase or step, milestones and deliverables are mentioned with respect to requirements. The work breakdown structure shows the overall breakup flow of the project and does not indicate any parallel flow. Work breakdown structure of the website is given below-

### **WORK BREAKDOWN STRUCTURE OF THE PROJECT**

Here the project is split into Analysis, Design, Coding and Testing phase. Further analysis is divided into Feasibility Report, Cost Estimation, and Project Planning and scheduling, and software requirement specification; design is divided into procedural design, user interface



design, and architecture design; coding is divided into comments and description, standardization of coding, error handling and security; testing is divided into black box testing and white box testing.



## GANTT CHART FOR PROJECT:->

When creating a software project schedule, the planner begins with a set of tasks **(the work breakdown structure)**. If automated tools are used, the work breakdown is input as a task network. **Effort, duration and start dates are input are each task network**. As a consequence of this input, a timeline chart also called a **Gantt chart** is generated. A timeline chart is developed for entire project.

S.NO	ACTIVITY	TIME (days)	START DATE	END DATE	DURATION OF TASK
1.	SYNOPSIS PREPARATION	5	7/2/2013	18/2/2013	
2.	TOOL LEARNING	10	19/2/2013	21/3/2013	
3.	Analysis	2	22/3/2013	5/4/2013	
4.	Design	10	5/4/2013	26/4/2013	
5.	Coding	8	27/4/2013	3/5/2013	
6.	Testing	4	4/5/ 02013	21/5/ 2013	
7.	Documentation	6	22/5/2013	24/5/2013	

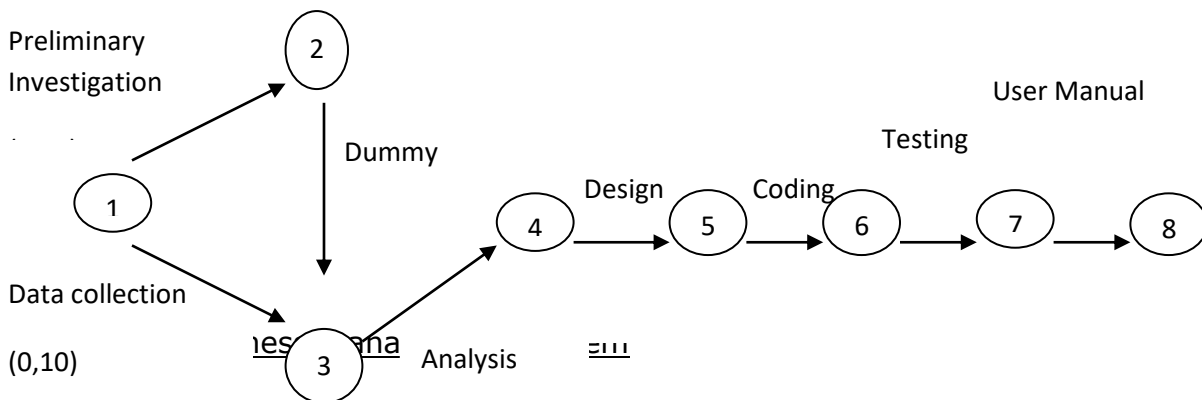
# Pert chart

Program evaluation and review technique (pert) is a project scheduling method that is applied to software development.

Pert provide quantitative tool that allow the software planner to-

- Determine the critical path-the chain of tasks that determines the duration of the project;
- Establish "most likely" time estimates for individual tasks by applying statistical models; and
- Calculate "boundary times" that defines a time "window" for a particular task.

## Pert chart for project-



# S/W REQUIREMENT SPECIFICATION

## **SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

This document is generated as output of requirement analysis. The requirement analysis involves obtaining a clear and throughout understanding of the product to be developed. Thus SRS should be consistent, correct unambiguous and complete document. The developer of the system can prepare SRS after detail communication with the customer.

### **An SRS can clearly define the following:**

✦ **External interface of a system:-** they identify the information which is to flow 'from and to the system'.

✦ **Functional and non:-**functional requirement of the system.

✦ **Design constraints:-**designing constraints of the system.

The SRS for the project is given below-

### **PURPOSE**

SRS contain details of the proposed software system, sufficient enough for designers to design the system. Thus SRS is means of communication the finding of analysis stage to the design stage. The SRS includes:

- Interface
- logical database
- Hardware
- Performance and other constraints

### **General information**

**Project titled:** "QUERY BROWSER" -A User friendly software for oracle

**Starting date:** 07-FEB-2013

**Expected end date:** 24-MAY-2013

## **Hardware and Software Requirements:**

### **Hardware**

- Pentium 4 with 1700 GHz speed or above
- Minimum of 256MB RAM
- Minimum of 40GB Hard Disk
- Dot Matrix/ Laser Printer/ DeskJet
- 15 'SVGA'colored monitor
- Serial plug and play Mouse

### **Software**

- Windows xp or above Operating system
- ORACLE 10g or above
- Virus scan program
- Crystal reports

### **Interface requirement**

The interface requirements include:

- ✚ Easy to follow interface
- ✚ No hidden buttons
- ✚ Relevant error message

### **Functional requirement**

- ✚ There should be detailed information about the division of the project
- ✚ There should be detailed information about user need.
- ✚ There should be detailed information about the database.
- ✚ There should be detailed information about the
- ✚ The administrator and the parties should be able to view all details.  
Administrator can also delete the records.
- ✚ Administrator should be able to add, update or delete all records  
updated by user.



### **Non functional requirements**

- Performance
- Reliability
- Efficiency
- Usability
- Portability
- Testability
- Understandability
- Modifiability

### **Basic processing actions of the system**

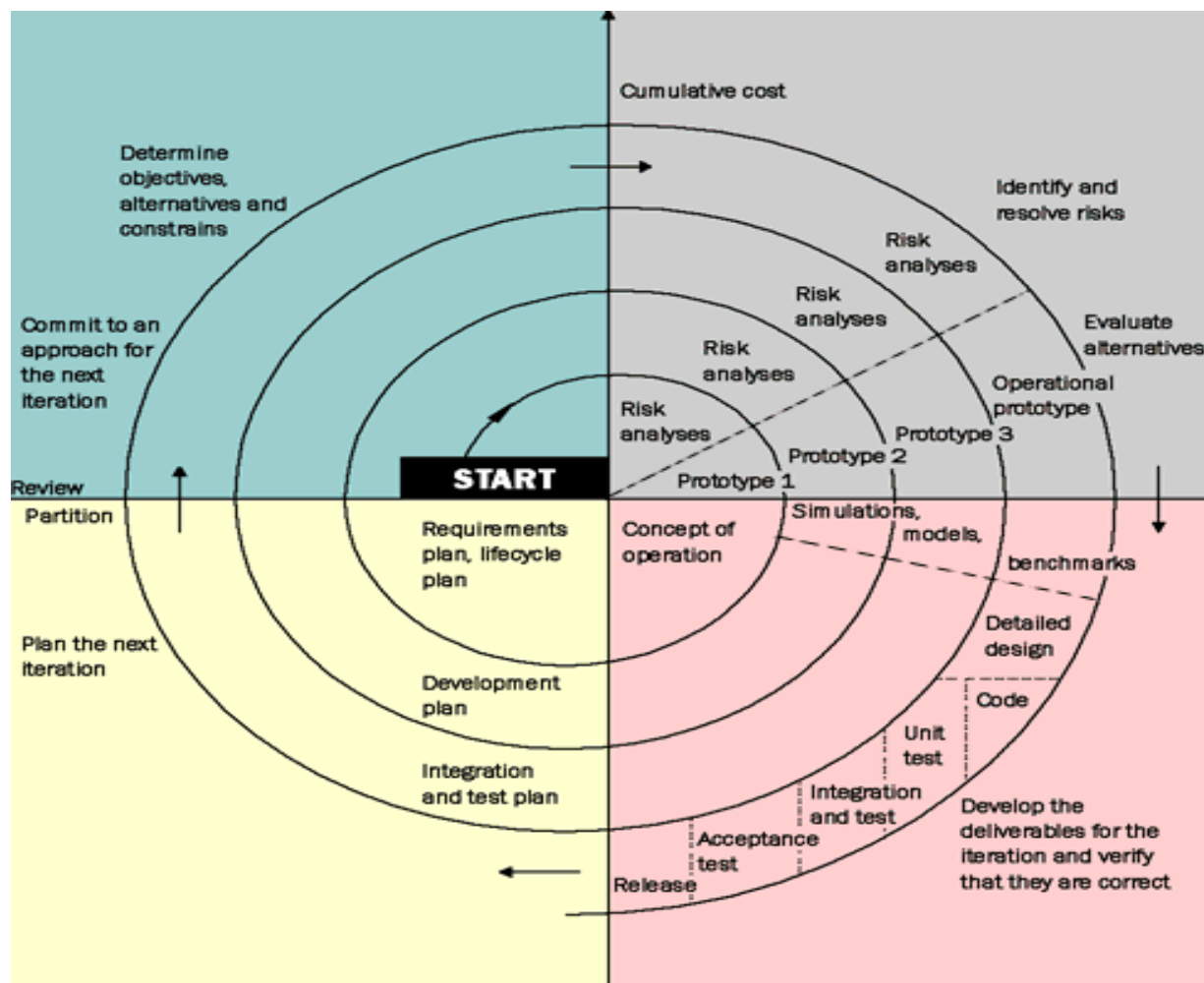
**The basic processing actions for the system are:**

- Verification of the user
- Uploads data

# **METHODOLOGY**

## Software Engineering Paradigm applied

In this project, Spiral model is used for the development. Spiral model has features of prototyping and linear sequential model. It helps to develop incremental versions of the software rapidly. Because software evolves as the process progresses, the developer and customer better understand and react to risks at each evolutionary level. This is a very powerful model to develop large-scale system and software.



This model is divided into a number of task regions. Each of the regions is populated by a set of work tasks called a task set that are done according to the requirement of the project to be developed. For small projects the number of tasks and their formality is low. Conversely, for large projects the reverse is true.

These task regions of spiral model are:

**Customer communication:** - This task is basically done to establish effective communication between developer and customer.

In this task region customers are asked about their requirements from the system, through a dialog between developer and the customer.

**Planning:-** This task is basically done to define resources, time lines and other project related information. In this task region resource, timelines and other project related information are considered and appropriate decision taken

**Risk analysis:-** This task is basically done to assess both technical and management risks In this region technical and management risks are identified and suitable measures taken wherever feasible.

**Engineering:-** This task is basically done to build one or more representations of the application. In this region one or more representations of application are built.

**Construction and Release:-** This task is basically done to construct, test, install and provide user support (e.g., documentation and training). In this region, application is constructed, tested, installed and user support is provided.

**Customer evaluation:-** This task is basically done to obtain customer feedback based on the evaluation of the software representation created during the engineering stage and implemented during the install stage in this region, feedback is taken from the customer about the overall performance of the application built during the engineering and implemented during installation stage.

The evolutionary process begins at the centre position and moves in a clockwise direction. Each traversal of the spiral typically results in a deliverable. For example, the first and second spiral traversals may result in the production of a product specification and a prototype, respectively. In this way a more stochastic version of the software is produced after some subsequent traversal. A key difference between the spiral model and other software models is that risk analysis is performed in former and not in later.

### **Advantages of the Spiral Model**

- The spiral model is a very good approach to develop large-scale software because of its evolutionary approach to develop software.

Besides this both customer and developer can identify and remove the risks that can occur during the software development.

- ✚ Prototyping is used to reduce the risk as it can be developed at any stage of the evolutionary level.
- ✚ Due to its demand of considering technical risks at all stages, it reduces risks that may become a big problem in future.
- ✚ It combines iterative and linear feature which helps in developing better software.

Thus according to the steps of spiral model first of all following steps are performed for the proposed project:

- The requirements are collected and partial software is developed fulfilling basic requirements of the user's (based on the requirement collected).
- As the information, the user wants to put on the software and the manner in which he wants to give the information has already been discussed, the technical and managerial risks were reduced.
- After designing and creation of some pages of site, a feedback is asked from the users. According to the given feedback, further refinement to the software was made, wherever required.

## **SYSTEM DESIGN**

## **SYSTEM DESIGN**

✚ The design of the system is most critical factor affecting the quality of the software; it has major impact on the phases particularly testing and maintenance. This document is similar to blueprint or plan for the solution.

The design activity is often divided into two phases:

✚ **System Design**

✚ **Detailed Design**

✚ **Database design**

### **SYSTEM DESIGN:-**

✚ System design aims to identify the modules there should be in the system, the specification of these modules and how they interact with each other to produce the desired result. At the end of system design all the major data structures, file formats and the major modules in the system and their specifications are decided.



### **DETAILED DESIGN:-**

- During detailed design, the internal logic of each of the modules specified in the system design is decided. During this phase further of the data structures and the algorithm design of the module is specification. The logic of a module is usually specification in a high-level design description language, which is independent of the language in which the software will eventually be implemented.

### **DATABASE DESIGN:-**

- A database may be thought of a set of related files. Related files mean that records in another file. The conventional file based systems emphasized that the application and the files were built around it. The database environment emphasizes the independently of application that use the data. The application are allowed evolve around a database design such that it can adopt the changing needs. Data becomes the central resource in the database environment.
- Information system is built around this central resource to give flexible access to data. Special software called Database Management System often referred the database operations.

## NUMBER OF MODULES

**1.** Login.

**2.** Table.

- ❖ Create Table
- ❖ Open Table
- ❖ Alter Table
- ❖ Rename Table
- ❖ Delete Table

**3.** Edit

- ❖ Insert Record.
- ❖ Delete Record
- ❖ Update Record

**4.** Advance

- ❖ Print

## **DETAILS OF MODULES**

➤ **Login:-** It is used for security reasons so that authorized persons can use the Software.

➤ **Table:-** It is used for operations on any Table.

❖ **Create Table:-** It is used to create a new table.

❖ **Open Table:-** It is used to select a table and view its record

❖ **Alter Table:-** It is used to modify data structure of existing table.

➤ **Add New Fields:-** It is used to add new fields in old table.

➤ **Modify Column:-** It is used to change datatype and size of existing column.

➤ **Add Primary Key:-** It is used to add primary key in any existing table.

❖ **Rename Table:-** It is used to give a new name to the existing table.

❖ **Delete Table:-** It is used to delete a selected table along with its structure.

🔗 **Edit:** Other operations on table like insert, update, Delete can be performed in this module.

❖ **Insert Record:-** It is used to insert a new record and modify the existing entries of a record.

❖ **Delete Record:-** It is used to delete record based on some criteria.

❖ **Find /Update Record:-** It is used find any record and update.

🔗 **Advanced**

❖ **Print:-** It is used to print report through printer.

### **What Is a Table?**

Tables are the basic structure where data is stored in the database. In most cases, there is no way for the database vendor to know ahead of time what our data storage needs are, chances are that we will need to create tables in the database. Many database tools allow us to create tables without

writing SQL, but given that tables are the container of all the data, it is important to include the syntaxes of create, drop, alter, rename table.

### **Advantages of Table?**

It is a good idea to understand what goes into a table. Tables are divided into rows and columns. Each row represents one piece of data, and each column can be thought of as representing a component of that piece of data. So, for example, if we have a table for recording customer information, then the columns may include information such as First Name, Last Name, Address, City, Country, Birth Date, and so on. As a result, when we specify a table, we include the column headers and the data types for that particular column.

### **Table options:-**

- **Create table:-**we It is used to store data by executing SQL CREATE TABLE statement. This statement is one of the data definition language(DDL) statements. DDL statements are a subset of SQL statements that are used to create, modify, or remove oracle database structures.

**Syntax:-**        CREATE        TABLE        "table\_name"        column        1"  
"data\_type\_for\_column\_1", "column 2" "data\_type\_for\_column\_2",...)

**In the above syntax:**

- ☞ Table is the name of the table
- ☞ Column is the column name
- ☞ Datatype is column's datatype and length

#### HELP:

- ☞ Table : select a table name from the list of tables.
- ☞ Field: select a column name from the list of column
- ☞ Type: select the datatype of column
- ☞ Size: specify the size of the column
- ☞ Add fields: used to add selected items in selected item box. You can add more fields after pressing add button.
- ☞ Create composite primary key: check the radiobutton if you want to make any column a primary key
- ☞ Build: To execute the query
- ☞ Close: to close the current form

**example:-** if we are to create the customer table specified as above, we would type in

CREATE	TABLE	customer
(First_Name		char(50),
Last_Name		char(50),
Address		char(50),
City		char(50),
Country		char(25),
Birth_Date	date)	

- **Alter table:-** The ALTER TABLE statement allows you to rename an existing table. It can also be used to add, modify, or drop a column from an existing table.

Use alter table statement to :

- ⌚ add a new column
- ⌚ modify an existing column
- ⌚ drop a column

**syntax:-** ALTER TABLE table\_name RENAME TO new\_table\_name;

ALTER TABLE table

ADD/MODIFY/DROP (column datatype, column dataype, ....)

**In above syntax:**

- ⌚ Table is the name of the table

- ④ ADD/MODIFY/DROP is the type of modification
- ④ Column is the name of the new column
- ④ Datatype is the datatype and length of new expression

**example:-** ALTER TABLE customer RENAME TO cus\_name;      ALTER  
TABLE dept\_90

ADD (job\_id VARCHAR2(9))

- ④ **Add a new column**
- ④ **Modify an existing column**
- ④ **Drop a column**

### **Add Column:**

You can add a new column by executing this query. First select the table in which you want to add column. Give the name of column that you want to add. Also specify the datatype of column.

### **Syntax:**

ALTER TABLE table

ADD (column datatype, column dataype, ....

### **In above syntax:**

- ④ Table is the name of the table



- ④ ADD is the type of modification
- ④ Column is the name of the new column
- ④ Datatype is the datatype and length of new expression

### **Example:**

```
ALTER TABLE dept_90
```

```
ADD (job_id VARCHAR2(9))
```

### **Modify Column:**

You can modify an existing column ( i.e you can change column's datatype, size, name) by executing this query. First select the table in which you want to modify column. Give the name of column that you want to modify. Also specify the datatype of column.

### **Syntax:**

```
ALTER TABLE table
```

```
MODIFY (column datatype, column datatype , ....)
```

### **In above syntax:**

- ④ Table is the name of the table

- ⌚ MODIFY is the type of modification
- ⌚ Column is the name of the new column
- ⌚ Datatype is the datatype and length of new expression

### **Example:**

```
ALTER TABLE dept_90
```

```
MODIFY (last_name VARCHAR2(30))
```

### **Drop Column:**

You can drop a column from a table by using ALTER TABLE statement with DROP COLUMN clause..using alter table only one column is dropped at a time. First select the table in which you want to drop a column. Give the name of column that you want to drop. Also specify the datatype of column.

### **Syntax:**

```
ALTER TABLE table
```

```
DROP COLUMN column
```

### **In above syntax:**

- ⌚ Table is the name of the table

⌚ DROP is the type of modification

⌚ Column is the name of the new column

### **Example:**

```
ALTER TABLE dept_90
```

```
DROP COLUMN job_id
```

### **Update Table:**

You can modify existing rows by using the update statement.

### **Syntax:**

```
UPDATE table
```

```
SET column = value [,column = value, ...
```

```
WHERE condition;
```

### **In above syntax:**

⌚ Table is the name of the table

⌚ Column is the name of the column to populate

⌚ Value is the corresponding value

⌚ Condition identifies the rows to be updated

### **Example:**

```
UPDATE employees  
  
SET department_id = 110  
  
WHERE emp_id = 113;
```

- **Rename table:-** It is used to rename a table. First select table name that you want to rename and then specify the new name that you want to give. Press rename button to execute the query. Refresh button changes the name in the list of table.

**syntax:-** RENAME old\_name TO new\_name

**In above syntax :**

- ⌚ Old\_name is the old name of the table
- ⌚ New\_name is the new name of the table

**example:-** RENAME dept\_80 TO dept\_90

- **Drop table:-** Sometimes we may decide that we need to get rid of a table in the database for some reason. In fact, it would be problematic if we cannot do so because this could create a maintenance nightmare for the DBA's. Fortunately, SQL allows us to do it, as we can use the DROP TABLE command. Drop table removes the definition of an oracle

table. When you drop a table, the database loses all the data in the table and all indexes associated with it.

**syntax:-** DROP TABLE tablename;

**example:-** we will drop the above table which is given in the previous example

DROP TABLE customer;

🔑 Password specifies that the user must log in with this password

**example:-** CREATE USER harry SET PASSWORD 'cat'

### **Options of Manipulate ?**

➤ **Insert Record:-** It is used to insert a new record and modify the existing entries of a record.

**syntax:-** INSERT INTO "table\_name" ("column1", column2,...)  
VALUES ("value1", "value2", ...)

**example:** INSERT INTO Store\_Information(store\_name,sales,Date)VALUES ('Los Angeles', 900, 'Jan-10-1999')

- **Delete Record:-** It is used to delete record based on some criteria. You can remove existing rows from the table by using the delete statement. First select the table From which you want to delete a record. Specific rows are deleted if you specify the where clause. All rows are deleted if where clause is omitted.

- **syntax:- DELETE FROM table WHERE condition**

**In the above syntax:**

⌚ Table is the table name from which you want to delete a record.


⌚ Condition identifies rows to be deleted


**example:-** DELETE FROM dept\_90 WHERE dept\_name = 'finance';


- **Find /Update Record:-** It is used to search and Update a record based on a given criteria.


**LOGIN:-** No user name will be displayed automatically to the user. User has to type the USERNAME and type the respective PASSWORD. If the USERNAME and PASSWORD matches then a WELCOME message will display and after 5 second "Main form" opens automatically else an error message will be flashed.


**TABLE:-**

 **CREATE TABLE:-** User has to give table name, fieldname, type and its size. If all the entries are valid then table is created else error message is flashed.


 **OPEN TABLE:-** All the table names in a specific "user" will be displayed automatically. User has to select a table name from combo box to see its content in the grid.


 **ALTER TABLE:-** All the table names in a specific "user" will be displayed automatically. User has to select the table name to be modified along with its fieldname and new details. Finally table is altered.


 **RENAME TABLE:-** All the table names of current user will be displayed automatically. User has to type new table name. Finally if all entries are valid then the table name is renamed.

 **DELETE TABLE:-** All the table names in a specific "user" will be displayed automatically. user has to select a table name which has to be deleted.


## **MANIPULATE:-**

 **INSERT RECORD:-** All the table names in a specific “user” will be displayed automatically. User has to select one of the table names and its content will be displayed in a grid and finally you can insert new record.

 **DELETE RECORD:-** All the table names in a specific “user” will be displayed automatically. User has to select one of the table names and its content will be displayed in a grid and user has to select condition to delete a record. According to given condition record will delete. If there is any problem then an error message will be displayed.

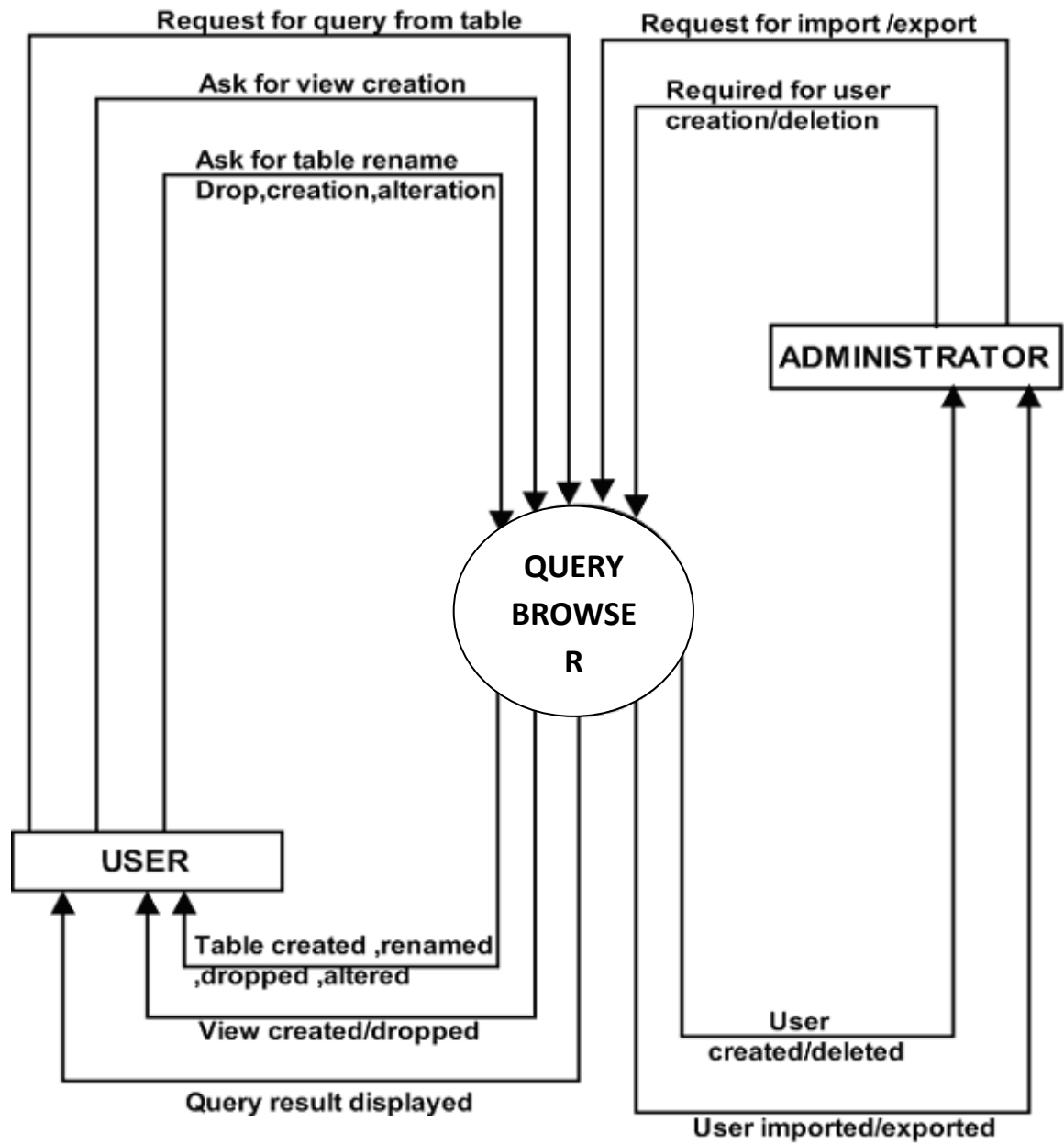
 **FIND/UPDATE RECORD: -** All the table names in a specific “user” will be displayed automatically. User has to select one of the table names and its content will be displayed in a grid and user has to select condition to find or update any record.

## **ADVANCED:-**

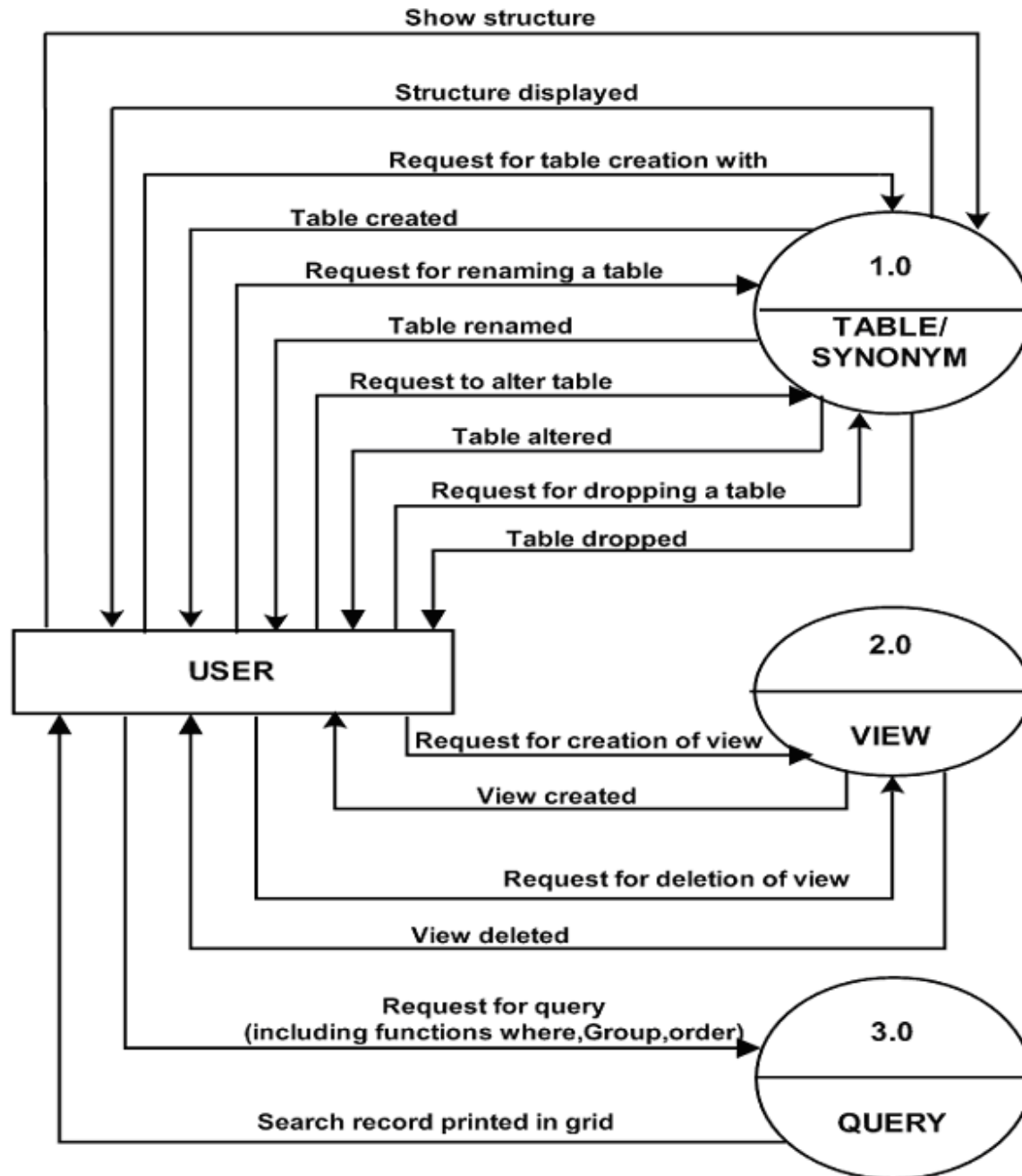
 **PRINT:-** All the table names in a specific “user” will be displayed automatically. User has to select table name then all fields are shown in list then user has to select the desired fields and click on print. If all entries are valid then crystal report will created.



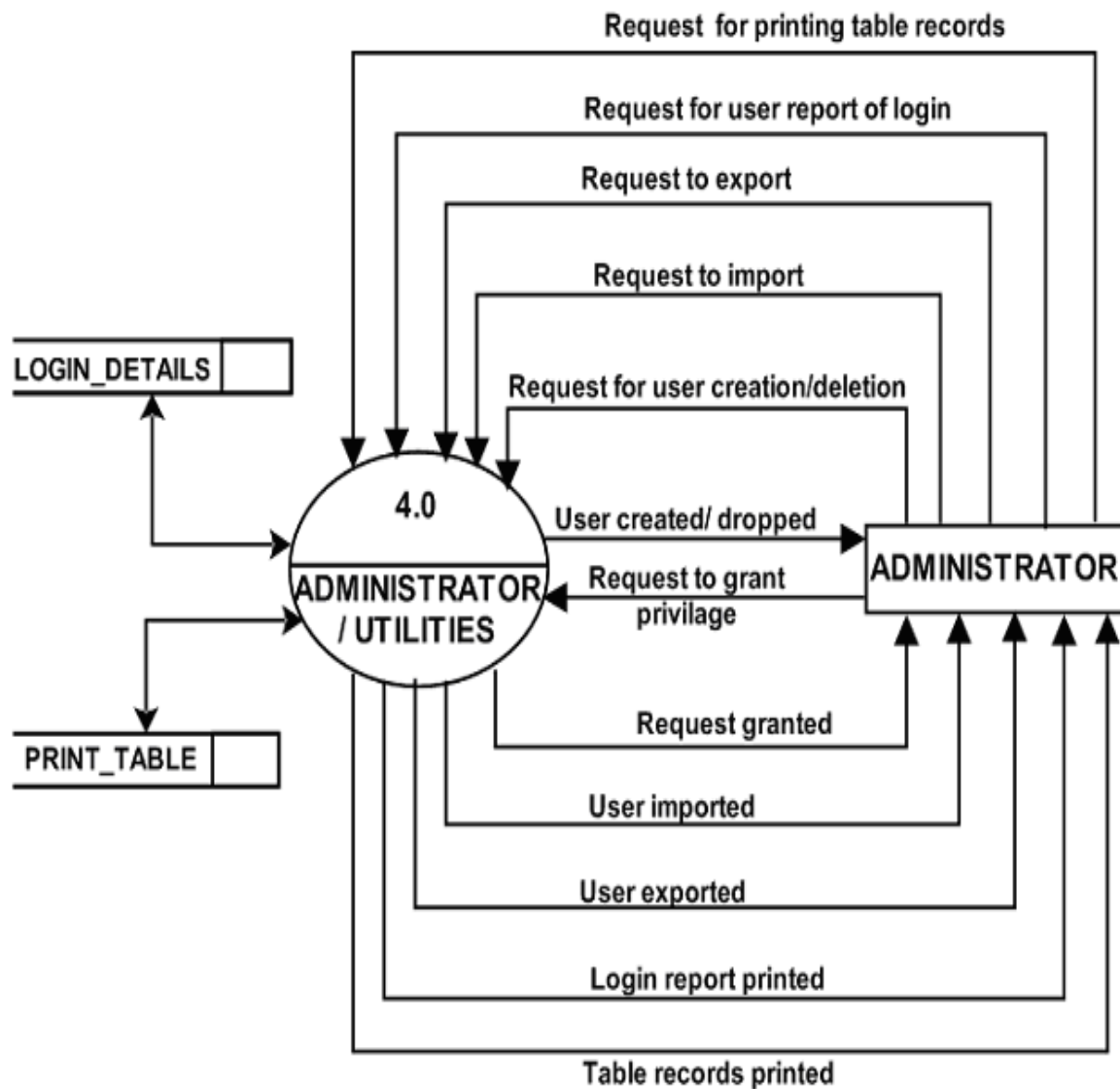
## **DFD AND ER DIAGRAM**



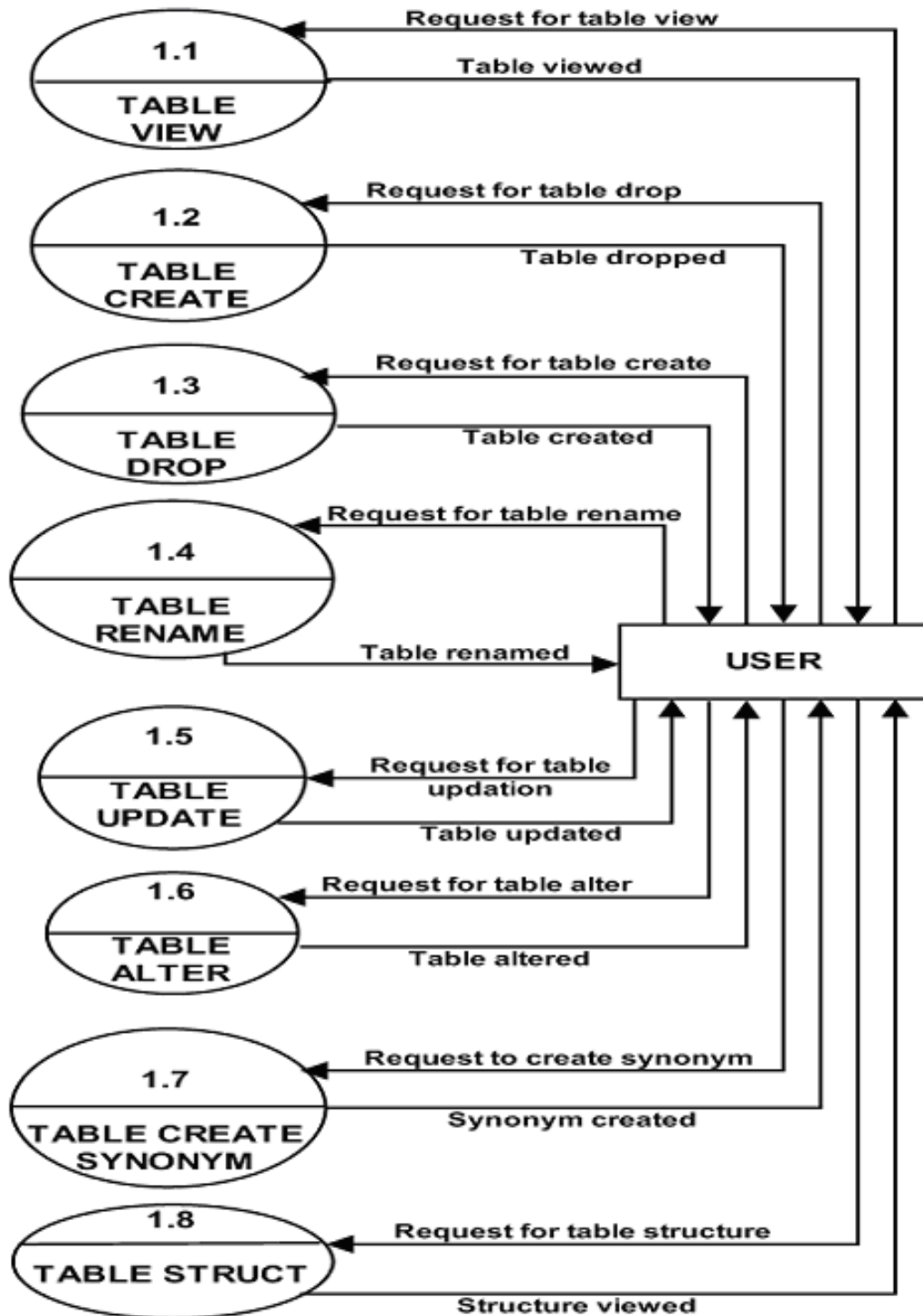
### CONTEXT LEVEL DFD



### FIRST LEVEL DFD



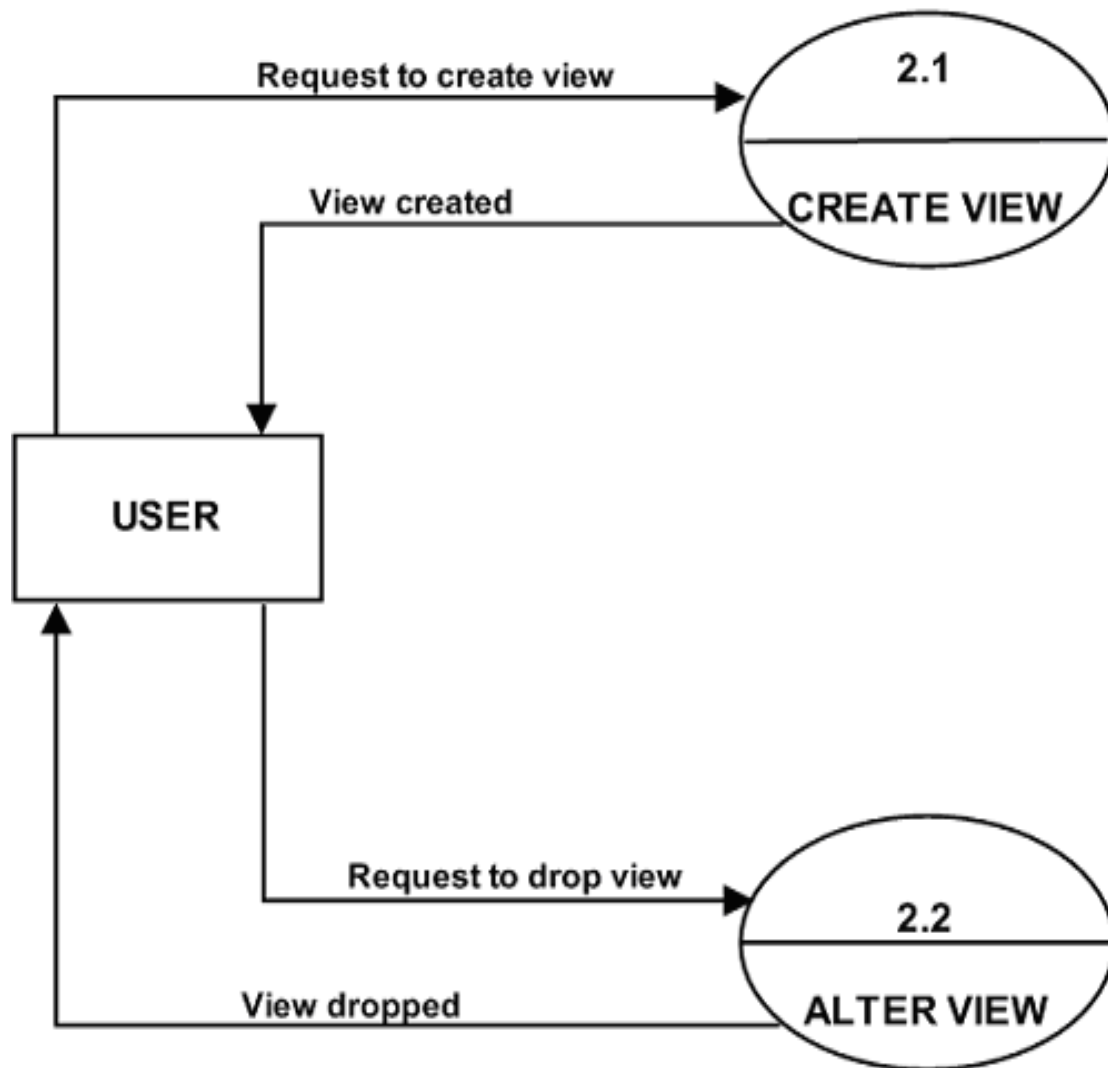
**FIRST LEVEL DFD(CONTINUED)**



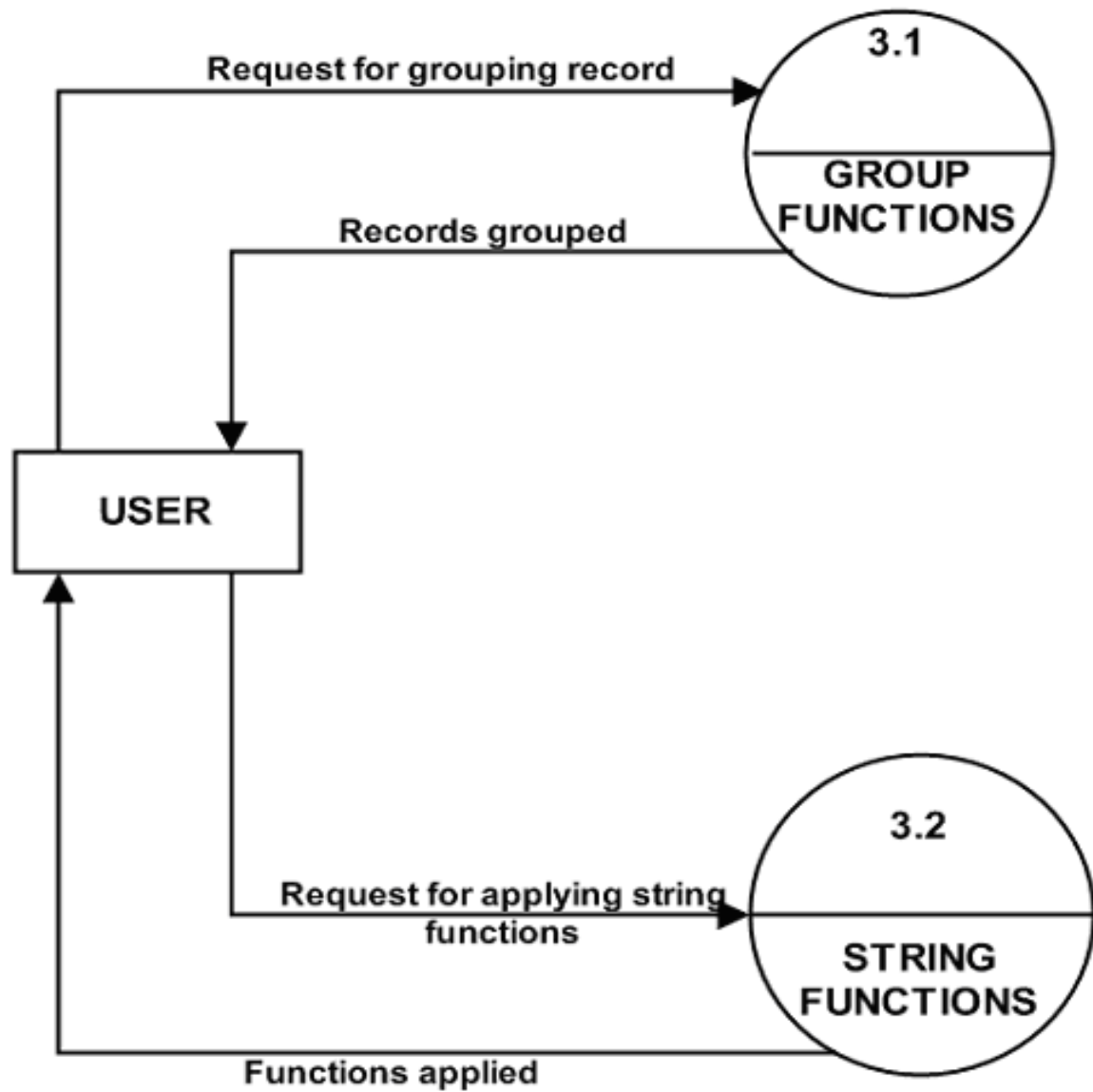
**SECOND**

**LEVEL**

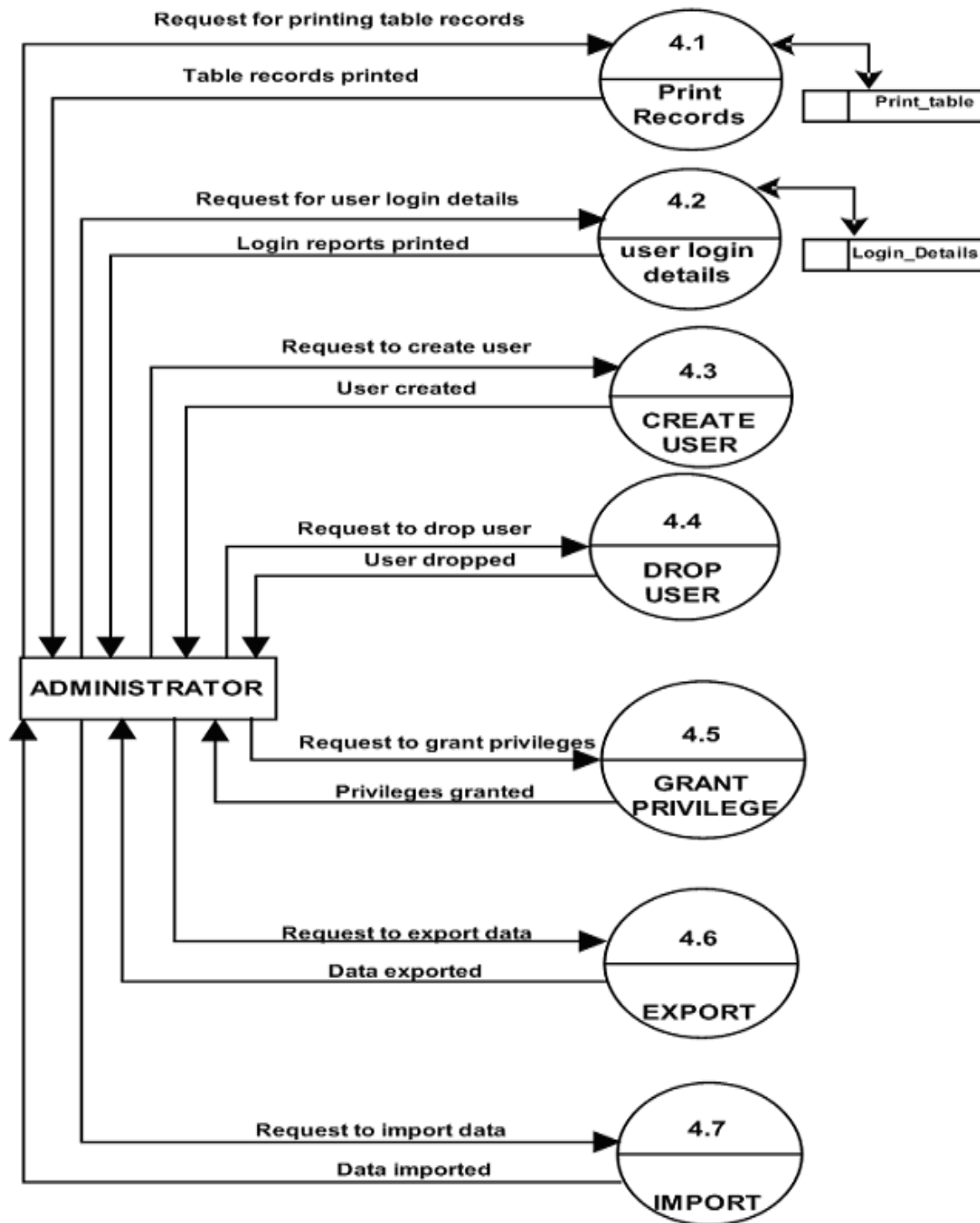
**DFD**



**SECOND LEVEL DFD**

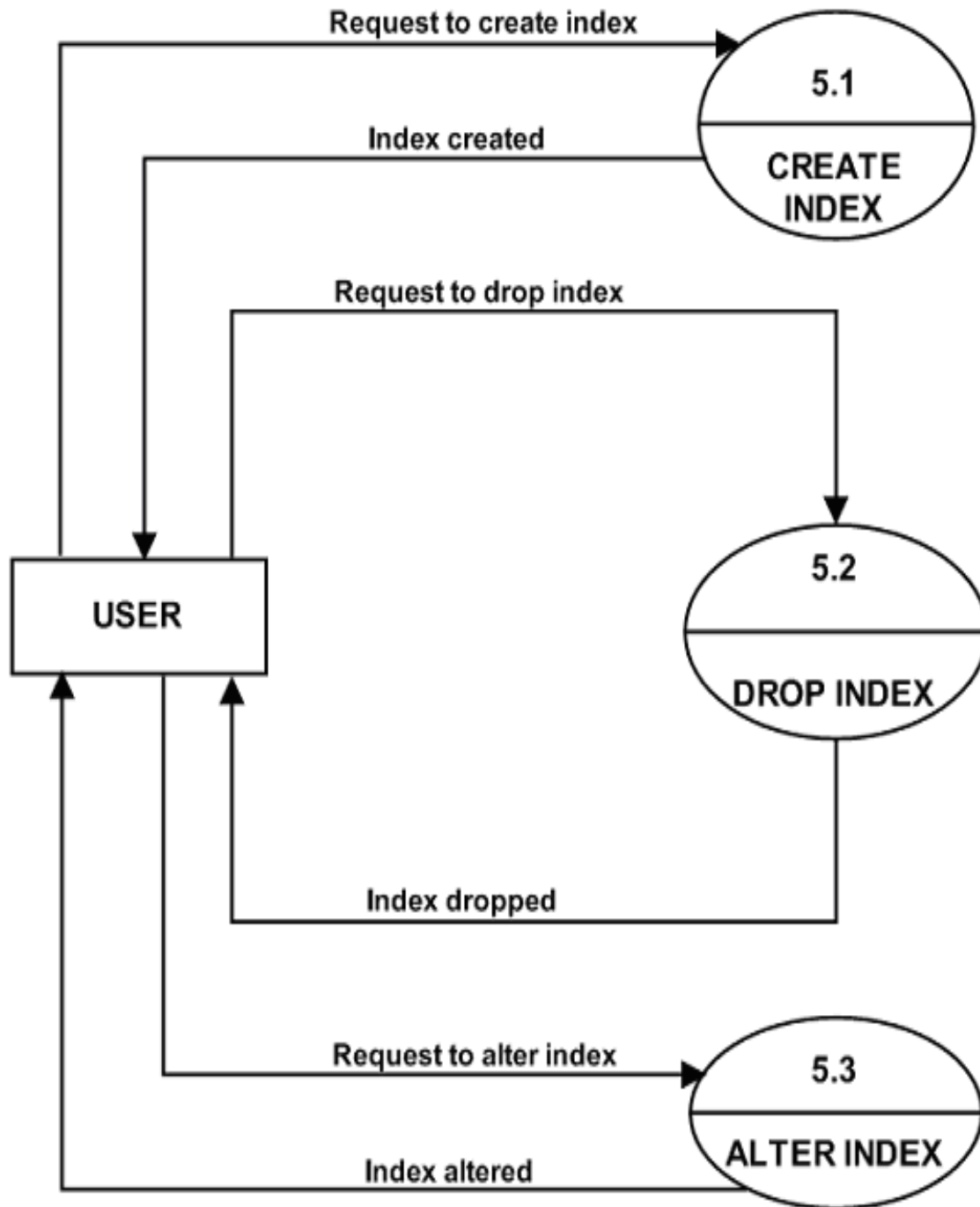


**SECOND LEVEL DFD**

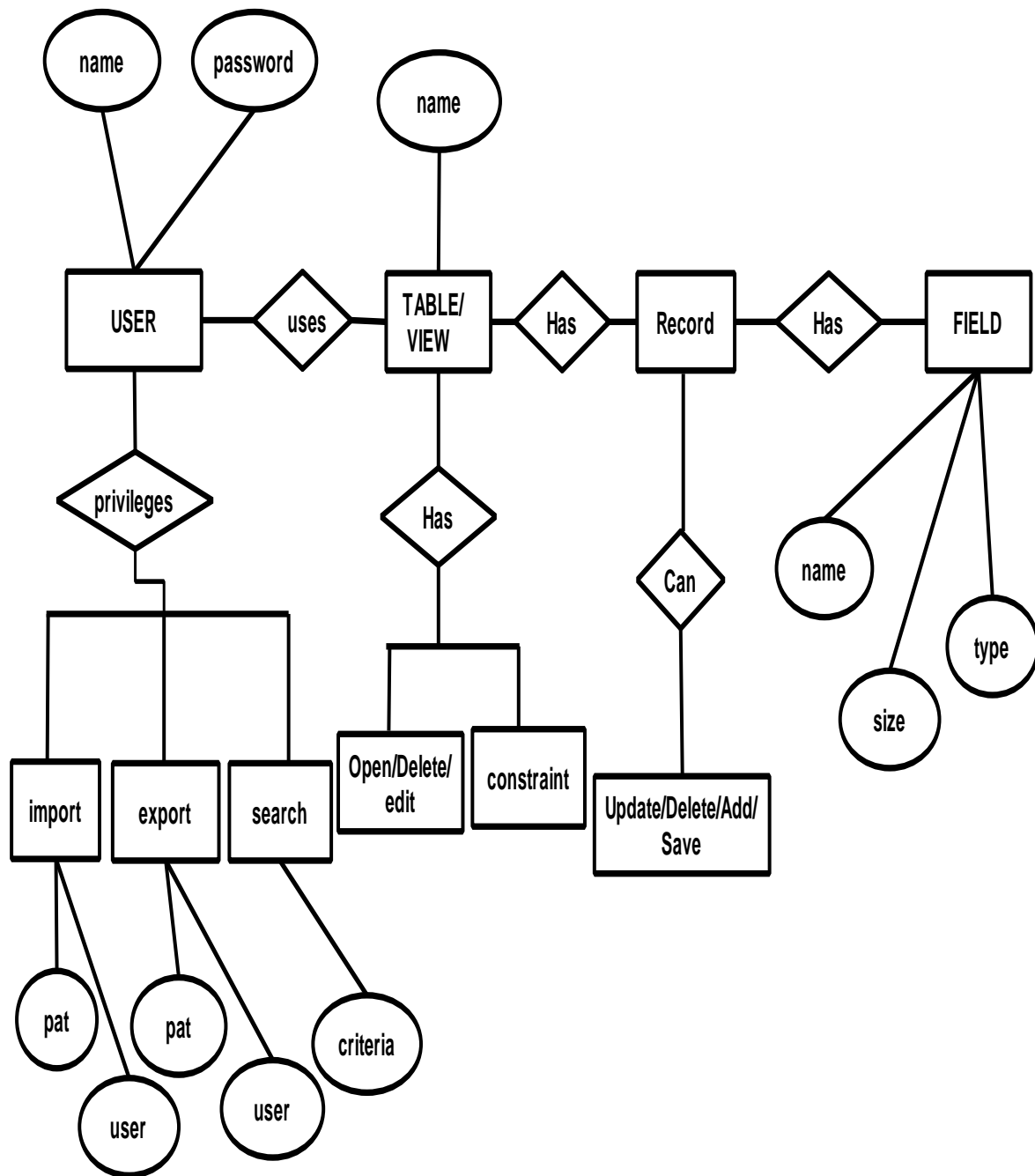


## SECOND LEVEL DFD



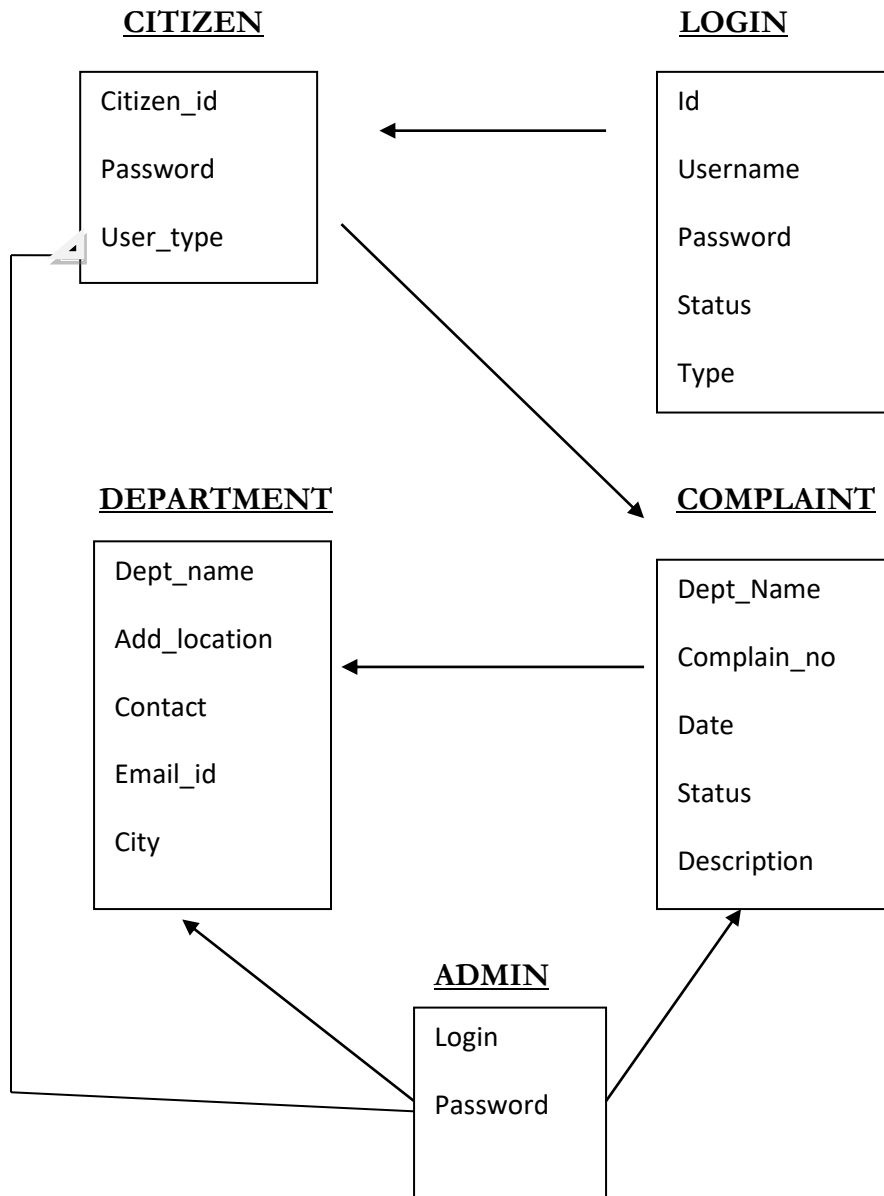


**SECOND LEVEL DFD**



## **E-R DIAGRAM**

# Database Schema



# Coding

# START UP FORM

```
package com.db;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.tree.*;

public class StartUpForm extends JWindow
{
    private JLabel l1;
    private JButton cmd1,cmd2;
    private JComboBox com1;
    private JPanel p1,p2;
    private JDesktopPane desktop;

    private void initLabel()
    {
        l1=new JLabel(new ImageIcon("images/db.jpg"));
    }

    private void initCombobox()
    {
        com1=new JComboBox();
        com1.addItem("NONE");
    }
}
```

```

        com1.addItem("ORACLE");
    }
    private void initButton()
    {

        cmd1=new JButton("CANCEL");
        cmd1.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent
ae)
            {

                StartUpForm.this.setVisible(false);
            }

        });

        cmd2=new JButton("PROCEED");
        this.getRootPane().
        setDefaultButton(cmd2);
        cmd2.addActionListener(new ActionListener()
        {

```

```

public void actionPerformed(ActionEvent ae)
{

Stringdb=com1.getSelectedItem().toString().toUpperCase(
);
if(db.equals("NONE"))
{
StartupForm.this.setVisible(false);
}
if(db.equals("DERBY"))
{
Frontlayout f=new Frontlayout();
p.pack();
StartupForm.this.setVisible(false);
}

if(db.equals("ORACLE"))
{

OracleBrowserForm f=new OracleBrowserForm();
f.pack();
StartupForm.this.setVisible(false);
}

```

```

        }

    });

}

private void initComponents()
{

    this.initLabel();
    this.initButton();
    this.initCombobox();
    p1=new JPanel(new GridLayout(1,4,5,5));

    p1.setBorder(BorderFactory.createTitledBorder(""));
    p1.add(new JLabel("SELECT DATABASE"));
    p1.add(com1);
    p1.add(cmd1);
    p1.add(cmd2);

    this.getContentPane().add(p1,BorderLayout.SOUTH);
    p2=new JPanel();

    p2.setBorder(BorderFactory.createTitledBorder(""));
    p2.add(l1);

    this.getContentPane().add(p2,BorderLayout.NORTH);

}

```



```

public StartUpForm()
{

    this.initComponent();
    this.setSize(350,350);
    //this.setLocation(250,250);
    this.setVisible(true);
    Helper.setPosition(this);
}

public static void main (String[] args)
{
try
{
StartUpForm p=new StartUpForm();
UIManager.setLookAndFeel(newcom.birosoft.liquid.LiquidL
ookAndFeel());
SwingUtilities.updateComponentTreeUI(p);
p.pack();
}
catch(Exception e){}

```

# ORACLE BROWSER FORM

```
package com.db;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.Connection;

public class OracleBrowserForm extends JFrame
{

    private JLabel l1,l3,l4,l5,l6,l7;
    private JTextField txt1,txt2,txt3,txt4;
    private JButton cmd1,cmd2,cmd3,cmd4;
    private JPasswordField jpf;

    private void cmdbutton()
    {
        cmd1=new JButton("Details >>");
        cmd2=new JButton("O.K");
        this.getRootPane().setDefaultButton(cmd2);
        cmd2.addActionListener(new ActionListener()
```

```

        {
            public void actionPerformed(ActionEvent
ae)
            {
                Oracle p=new Oracle();
                try
                {

                    p.setIP(txt1.getText());
                    p.setPort(txt2.getText());
                    p.setSid(txt3.getText());
                    p.setUser(txt4.getText());
                    p.setPassword(jpf.getText());

                    Connection
cn=p.getOracleConnection();
                    if(!cn.isClosed())
                    {

                        OracleQuerybrowserForm
f=new OracleQuerybrowserForm(cn);

                        OracleBrowserForm.this.setVisible(false);
                    }

                    else

```

```

        {

            JOptionPane.showMessageDialog(null, "Login
Failed...Retry\n\n"+p.ERROR);

        }

        catch (Exception e)

        {

            JOptionPane.showMessageDialog(null, "Login
Failed...Retry\n\n"+p.ERROR);

        }

    });

    cmd3=new JButton("Clear");
    cmd3.addActionListener(new ActionListener()

    {

        public void actionPerformed(ActionEvent
ae)

        {

            clear();

```

```

        }

    });

    cmd4=new JButton("Cancel");
    cmd4.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent
ae)
        {

            OracleBrowserForm.this.setVisible(false);
        }

    });
}

private void initLabel()
{
    l1=new JLabel(new
ImageIcon("images/Oracle.jpg"));

```

```
        l3=new JLabel("Server Host");
        l4=new JLabel("Port");
        l5=new JLabel("S.ID NO.");
        l6=new JLabel("Username");
        l7=new JLabel("Password");

    }

    private void textfield()
    {
        txt1=new JTextField(10);
        txt1.setText("127.0.0.1");
        txt2=new JTextField(2);
        txt2.setText("1521");
        txt3=new JTextField(10);
        txt3.setText("xe");
        txt4=new JTextField(10);

    }

    private void clear()
    {
        txt1.setText("");
    }
}
```

```

        txt2.setText("");
        txt3.setText("");
        txt4.setText("");
        jpf.setText("");

    }

    private void initComponents()
    {
        this.initLabel();
        this.textfield();
        this.cmdbutton();

        JPanel p1=new JPanel();

        p1.setBorder(BorderFactory.createTitledBorder(""));

        p1.add(l1);

        JPanel p2=new JPanel(new
GridLayout(6,2,5,5));

        p2.setBorder(BorderFactory.createTitledBorder("Conn
ect to Oracle server Instance"));

```

```
p2.add(l3);  
p2.add(txt1);  
p2.add(l4);  
p2.add(txt2);  
p2.add(l5);  
p2.add(txt3);  
p2.add(l6);  
p2.add(txt4);  
p2.add(l7);  
jpf=new JPasswordField(10);  
p2.add(jpf);
```

```
JPanel p3= new JPanel(new  
GridLayout(1,4,5,5));
```

```
p3.setBorder(BorderFactory.createTitledBorder(""));  
p3.add(cmd1);  
p3.add(cmd2);  
p3.add(cmd3);
```



```
        p3.add(cmd4);

this.getContentPane().add(p1, BorderLayout.NORTH);

this.getContentPane().add(p2, BorderLayout.CENTER);

this.getContentPane().add(p3, BorderLayout.SOUTH);

}

public OracleBrowserForm()
{
    this initComponents();
    this.setTitle("Oracle Query Browser");
    this.setSize(450, 450);
    this.setLocation(300, 300);
    this.setVisible(true);
    this.setResizable(false);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)
;

}
```

# ORACLE

```
package com.db;

import java.sql.Connection;
import java.sql.DriverManager;
public class Oracle
{
    public String ERROR;

    private String ip;
    private String port;
    private String sid;
    private String user;
    private String password;
    private String url;

    private void makeUrl()
    {

this.url="jdbc:oracle:thin:@"+this.ip+": "+this.port+": "+
+this.sid;

    }
```

```
public Oracle()  
{  
    this.ip="127.0.0.1";  
    this.port="1521";  
    this.sid="xe";  
    this.user="system";  
    this.password="";  
    this.makeUrl();  
}  
public Oracle(String ip)  
{  
    this.ip=ip;  
    this.port="1521";  
    this.sid="xe";  
    this.user="sytem";  
    this.password="";  
    this.makeUrl();  
}  
public Oracle(String ip,String port)  
{  
    this.ip=ip;  
    this.port=port;
```

```

        this.sid="xe";
        this.user="system";
        this.password="";
        this.makeUrl();
    }

    public Oracle(String ip,String port,String
database)
    {
        this.ip=ip;
        this.port=port;
        this.sid="xe";
        this.user="Oracle";
        this.password="";
        this.makeUrl();
    }

    public Oracle(String ip,String port,String
database,String user)
    {
        this.ip=ip;
        this.port=port;
        this.sid="xe";
        this.user=user;
        this.password="";
    }

```

```

        this.makeUrl();
    }

    public Oracle(String ip,String port,String
database,String user,String password)
    {
        this.ip=ip;
        this.port=port;
        this.sid=sid;
        this.user=user;
        this.password=password;
        this.makeUrl();
    }

    public Connection getOracleConnection()
    {
        Connection cn=null;
        try
        {
            Class.forName("oracle.jdbc.OracleDriver");

            cn=DriverManager.getConnection(this.url,this.user,this.
password);
        }
    }

```

```

        catch (Exception e)
        {
            this.ERROR=e.getMessage();
        }

        return cn;
    }

    public Connection getOracleConnection(boolean
commitMode)
    {
        Connection cn=null;
        try
        {
            Class.forName
("oracle.jdbc.driver.OracleDriver");

cn=DriverManager.getConnection(this.url,this.user,this.
password);

            if(!cn.isClosed())
            {
                cn.setAutoCommit(commitMode);
            }
        }
        catch (Exception e)

```

```

        {
            this.ERROR=e.getMessage();
        }

        return cn;
    }

    public void setIP(String ip)
    {
        this.ip=ip;
        this.makeUrl();
    }

    public void setPort(String port)
    {
        this.port=port;
        this.makeUrl();
    }

    public void setDatabase(String database)
    {

        this.makeUrl();
    }

    public void setUser(String user)

```

```
{  
    this.user=user;  
    this.makeUrl();  
}  
public void setPassword(String password)  
{  
    this.password=password;  
}  
public String getIP()  
{  
    return this.ip;  
}  
public String getPort()  
{  
    return this.port;  
}  
public String getSid()  
{  
    return this.sid;  
}  
public String getUser()  
{
```



```
        return this.user;
    }
    public String getPassword()
    {
        return this.password;
    }
    public String getUrl()
    {
        return this.url;
    }
    public void setSid(String s)
    {
        this.sid=s;
    }
}
```

## RENAME ORACLE TABLE

```
package com.db;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.*;
import java.sql.*;

public class RenameOracleTable extends JFrame
{
    private Connection cn;
    private JLabel l1,l2;
    private String tablename;
    private JTextField txt1;
    private JTextField txt2;
    private JButton cmd1,cmd2;
    // private DataComboBox cbo;
    private void initLabel()
    {

        l1=new JLabel(" OLD TABLE NAME");
        l2=new JLabel(" NEW TABLE NAME");
    }
    private void initTextBox()
```

```

{
    txt1=new JTextField(10);
    txt1.setText(this.tablename);
    txt2=new JTextField(10);
}

private void initCmdButton()
{
    cmd1=new JButton("Rename");
    cmd1.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ae)
        {
            initQuery();
        }
    });
    cmd2=new JButton("Cancel");
    cmd2.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent
ae)
        {
            clear();
        }
    }

```

```

        });
    }
    private void initQuery()
    {
        try
        {

            String str1="ALTER TABLE "+
txt1.getText().toString()+" RENAME TO
"+txt2.getText().toString();

            PreparedStatement
ps=cn.prepareStatement(str1);

            ps.executeUpdate();

            JOptionPane.showMessageDialog(null,"Successfully
Renamed");

        }
        catch(Exception e)
        {

            JOptionPane.showMessageDialog(null,e.getMessage());

        }

    }

```

```

private void clear()
{
    txt1.setText("");
    txt2.setText("");
}

private void initComponents()
{
    this.initLabel();

    this.initCmdButton();
    this.initTextBox();
    //this.initComboBox();

    JPanel p1=new JPanel(new GridLayout(2,2,5,5));
    p1.setBorder(BorderFactory.createTitledBorder(""));
    p1.add(l1);
    p1.add(txt1);
    p1.add(l2);
    p1.add(txt2);

    JPanel p2=new JPanel(new GridLayout(1,2,5,5));

    p2.setBorder(BorderFactory.createTitledBorder(""));
    p2.add(cmd1);
    p2.add(cmd2);
}

```

```

this.getContentPane().add(p1,BorderLayout.NORTH);

this.getContentPane().add(p2,BorderLayout.SOUTH);
}

public RenameOracleTable(Connection con,String str)
{
    cn=con;
    this.tablename=str;
    this.initComponent();
    this.setTitle("Rename Table");
    this.setVisible(true);
    this.setSize(300,300);
    Helper.setPosition(this);

    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
}

}

```

# ORACLE QUERY BROWSER FORM

```
package com.db;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.tree.*;
import java.sql.*;
import javax.swing.table.DefaultTableModel;
import java.util.Vector;
import java.sql.Connection;
import java.sql.PreparedStatement;

public class OracleQuerybrowserForm extends JFrame
{

    protected Connection cn;

    private JButton cmd1,cmd2,cmd3,cmd4,cmd5,b1,cmd6,cmd7;
    private JTextArea txtr1;
    private JToolBar tlb;
    private JTree tree1,tree2;
    private DefaultMutableTreeNode root1,root2;
    private JTabbedPane tab1;
```

```

private JPopupMenu Pmenu;

private JMenuItem menuItem;

private JMenuBar bar ;

private JMenu
Table,Advanced,Edit,View,Administrator;

private JMenuItem
table1,table2,table3,table4,table5;

private JMenuItem adv1,adv2,adv3;

private JMenuItem edit1,edit2,edit3;

private JMenuItem view1,view2;

private JMenuItem admin1,admin2;

private DataTable table;


private void PopUpMenu()
{

    Pmenu = new JPopupMenu();
    menuItem = new JMenuItem("Create new Table");
    Pmenu.add(menuItem);

    menuItem.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {

```



```

        CreateTable c=new CreateTable(cn);

        c.pack();

        setTree();

    }

});

menuItem = new JMenuItem("Insert Row");

Pmenu.add(menuItem);

menuItem.addActionListener(new ActionListener()

{

    public void actionPerformed(ActionEvent ae)

    {

        String

name=tree1.getLastSelectedPathComponent().toString();

        Helper.insertRow(cn,table,name);

    }

});

menuItem = new JMenuItem("Rename Table");

Pmenu.add(menuItem);

menuItem.addActionListener(new ActionListener()

{

```

```

        public void actionPerformed(ActionEvent e)
        {
            String
name=tree1.getLastSelectedPathComponent().toString().trim();

            RenameOracleTable r=new
RenameOracleTable(cn,name);

            r.pack();

        }

    });

menuItem = new JMenuItem("Drop Table");
Pmenu.add(menuItem);
menuItem.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        droptable();
        setTree();
    }
});

```

```

menuItem = new JMenuItem("Refresh");
Pmenu.add(menuItem);
menuItem.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {

        setTree();
        txtrl.setText("");
    }
});

}

public void MenuContainer()
{

    table1=new JMenuItem("Open Table");
    table1.addActionListener(new ActionListener()
    {

        public void
actionPerformed(ActionEvent ae)
        {

```

```

        String
name=tree1.getLastSelectedPathComponent().toString();

        TreeNode
node=(TreeNode) tree1.getLastSelectedPathComponent();

        if(!node.isLeaf())
        {

                JOptionPane.showMessageDialog(null,name);

                                table.setData(cn," SELECT *
FROM "+name);

        }

    });

        table2=new JMenuItem("Rename Table");

        table2.addActionListener(new
ActionListener()

        {

                public void
actionPerformed(ActionEvent ae)

        {

```

```

        RenameOracleTable r=new
RenameOracleTable(cn);

        r.pack();

    }

    });

    table3=new JMenuItem("Create Table");

    table3.addActionListener(new
ActionListener()

    {

        public void
actionPerformed(ActionEvent ae)

        {

            CreateTable c=new CreateTable(cn);

            c.pack();

        }

    });

    table4 =new JMenuItem("Drop Table");

    table4.addActionListener(new
ActionListener()

```

```

        {

                public void
actionPerformed(ActionEvent ae)

                {

                        droptable();

                        setTree();

                }

        });


        table5=new JMenuItem("Create using Another
Table");

        table5.addActionListener(new
ActionListener()

        {

                public void
actionPerformed(ActionEvent ae)

                {

                        CreateTable c=new
CreateTable(cn);

                        c.pack();

                }

        }

```

```

        });

        adv1=new JMenuItem("Print");
        adv1.addActionListener(new
        ActionListener()
        {

            public void
        actionPerformed(ActionEvent ae)
            {

                if(table.getRowCount()>0)
                {

                    Helper.printTable(table);

                }

            }

        });

        edit1=new JMenuItem("Insert Record");
        edit1.addActionListener(new
        ActionListener()
        {

```

```

        public void
actionPerformed(ActionEvent ae)
    {

        initInsertValues();

    }

});

edit2=new JMenuItem("Delete Record");
edit2.addActionListener(new
ActionListener()
{

    public void
actionPerformed(ActionEvent ae)
    {

        try
        {

            TreeNode
node=(TreeNode)tree1.getLastSelectedPathComponent();

            String sql="DELETE FROM
"+node.getParent().toString()+" WHERE
"+node+"="+JOptionPane.showInputDialog("Enter Value For
: "+node);

```



```

JOptionPane.showMessageDialog(null,sql);

        PreparedStatement
ps=cn.prepareStatement(sql);

        int a=ps.executeUpdate();
        if(a>0)
        {
            table.setData(cn,"Select *
from "+node.getParent().toString());
        }
        catch(Exception e){}

    }

});

view1=new JMenuItem("Create View");
view1.addActionListener(new
ActionListener()
{

```

```

        public void
actionPerformed(ActionEvent ae)
    {
        String
name=tree1.getLastSelectedPathComponent().toString().trim();

        CreateView e=new
CreateView(cn,name);

        e.pack();

    }

});

view2=new JMenuItem("Drop View");
view2.addActionListener(new
ActionListener()
{

    public void
actionPerformed(ActionEvent ae)
    {
        String
name=tree1.getLastSelectedPathComponent().toString().trim();

        DropView d=new DropView(cn,name);

```

```

        d.pack();

    }

});

    admin1=new JMenuItem("Create User");
    admin1.addActionListener(new
ActionListener()
    {

        public void
actionPerformed(ActionEvent ae)
        {

            CreateUser c=new CreateUser(cn);

            c.pack();

        }

    });

    admin2=new JMenuItem("Drop User");
    admin2.addActionListener(new
ActionListener()
    {

        public void
actionPerformed(ActionEvent ae)

```

```

        {

            String
name=tree1.getLastSelectedPathComponent().toString().trim();

            DropUser e=new DropUser(cn,name);

            e.pack();

        }

    });

    Table=new JMenu("Table");

        Advanced=new JMenu("Advanced");

        Edit=new JMenu("Edit");

        View=new JMenu("View");

        Administrator=new
JMenu("Administrator");

        Table.add(table1);

    Table.add(table2);

    Table.add(table3);

        Table.add(table4);

    Table.add(table5);

```

```
Advanced.add(adv1);  
    Edit.add(edit1);  
    Edit.add(edit2);  
    View.add(view1);  
    View.add(view2);  
    Administrator.add(admin1);  
    Administrator.add(admin2);
```

```
bar=new JMenuBar();  
    bar.add(Table);  
    bar.add(Advanced);  
    bar.add(Edit);  
    bar.add(View);  
    bar.add(Administrator);  
    this.setJMenuBar(bar);  
}
```

```
private void initTree()  
{  
    try  
{
```

```

        root1=new DefaultMutableTreeNode("-----
"+cn.getMetaData()+"-----");

        this.setTree();

        root2=new DefaultMutableTreeNode("History");
    }

    catch(Exception e){}


    tree1=new JTree(root1);
    tree1.addMouseListener(new MouseAdapter()
    {
        public void mouseReleased(MouseEvent me)
        {
            if(me.isPopupTrigger())
            {

Pmenu.show(me.getComponent(),me.getX(),me.getY());

            }

        }
    });

    tree2=new JTree(root2);

```

```

        tab1=new JTabbedPane();
        tab1.add("Schemata",new JScrollPane(tree1));
        tab1.add("History",new JScrollPane(tree2));

        JPanel p1 =new JPanel(new GridLayout(1,1));
        p1.add(tab1);

        this.getContentPane().add(p1,BorderLayout.EAST);
    }
    private void initToolBar()
    {

        tlb=new JToolBar();

        txtr1=new JTextArea();

        cmd1=new JButton(new
ImageIcon("images/goback.jpg"));

        cmd1.addActionListener(new ActionListener()
        {

```

```

        public void actionPerformed(ActionEvent
ae)

        {

            OracleBrowserForm q=new
OracleBrowserForm();

            q.pack();

OracleQuerybrowserForm.this.setVisible(false);

        }

    });

    cmd2=new JButton(new
ImageIcon("images/next.jpg"));

    cmd3=new JButton(new
ImageIcon("images/refresh1.jpg"));

    cmd4=new JButton(new
ImageIcon("images/execute.jpg"));

    cmd5=new JButton(new
ImageIcon("images/Stop.jpg"));

    cmd4.addActionListener(new ActionListener()

    {

        public void actionPerformed(ActionEvent
ae)

        {

            try

```



```

        {

            if(txtr1.getText().length()>0)

                {

                    String
sql=txtr1.getText().toUpperCase();

                    if(sql.startsWith("SELECT")
|| sql.startsWith("DESC") || sql.startsWith("SHOW"))

                        {

                            table.setData(cn,sql);

                        }

                        else

                            {

                                PreparedStatement
ps=cn.prepareStatement(sql);

                                int a=ps.executeUpdate();

                            }

                        }

                    }

                catch(Exception e)

                    {

                        JOptionPane.showMessageDialog(null,e.getMessage());

```

```

        }

    }

    });

    tlb.add(cmd1);
    tlb.add(cmd2);
    tlb.add(cmd3);
    tlb.add(txtr1);
    tlb.add(cmd4);
    tlb.add(cmd5);

    this.getContentPane().add(tlb, BorderLayout.NORTH);
}

private void initTable()
{

    table=new DataTable();

    JPanel p1=new JPanel(new GridLayout(1,1,5,5));

    p1.setBorder(BorderFactory.createTitledBorder("Result
    Set"));

    p1.add(new JScrollPane(table));

    this.getContentPane().add(p1, BorderLayout.CENTER);

```

```

    }

    private void initComponents()
    {
        this.MenuContainer();
        this.initTree();
        PopUpMenu();
        this.initTable();
        cmd6=new JButton("APPLY INSERT");
        cmd6.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent
ae)
            {

                initInsertValues();
            }

        });

```

```

        cmd7=new JButton("APPLY DELETE");

```

```

cmd7.addActionListener(new ActionListener()
{

    public void actionPerformed(ActionEvent ae)
    {

        try
        {

            TreeNode
node=(TreeNode)tree1.getLastSelectedPathComponent();

            String sql="DELETE FROM
"+node.getParent().toString()+" WHERE
"+node+"="+JOptionPane.showInputDialog("Enter Value For
: "+node);

            JOptionPane.showMessageDialog(null,sql);

            PreparedStatement
ps=cn.prepareStatement(sql);

            int a=ps.executeUpdate();

            if(a>0)
            {

                table.setData(cn,"Select *
from "+node.getParent().toString());

            }

```

```

        }

        catch(Exception e){}

    }

    });

    JPanel p2=new JPanel(new GridLayout(1,2,5,5));
    p2.setBorder(BorderFactory.createTitledBorder(""));
    p2.add(cmd6);
    p2.add(cmd7);

    this.getContentPane().add(p2, BorderLayout.SOUTH);
    }
    private void droptable()
    {

        try
        {

            String
name=tree1.getLastSelectedPathComponent().toString();

            TreeNode
node=(TreeNode) tree1.getLastSelectedPathComponent();

```

```

        if(!node.isLeaf())
        {
            if(table.getRowCount()==0)
            {

JOptionPane.showMessageDialog(null,name);

                String sql=" DROP TABLE
"+name;

                PreparedStatement ps=
cn.prepareStatement(sql);

                int a=ps.executeUpdate();

            }

        }

        catch(Exception e)
        {

JOptionPane.showMessageDialog(null,e.getMessage());

        }

    }

    public OracleQuerybrowserForm(Connection con)
    {

        cn=con;

```

```

        this.initToolBar();

        this.initcomponents();

        this.setTitle("Oracle Query
Browser.....");

        this.setVisible(true);

this.setSize(Toolkit.getDefaultToolkit().getScreenSize(
));

        this.setResizable(false);

this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }

private void initInsertValues()
{
    try
    {

        Vector v=new Vector();

        String
name=tree1.getLastSelectedPathComponent().toString();

```

```

        int count=table.getColumnCount();

        String str="INSERT INTO "+name+ " VALUES
(";

        for(int i=0;i<table.getColumnCount()-
1;i++)

        {

            str+="?, ";

            v.addElement(table.getValueAt(table.getSelectedRow(
),i).toString());

        }

        str+="?) ";

v.addElement(table.getValueAt(table.getSelectedRow(),co
unt-1).toString());

        JOptionPane.showMessageDialog(null,str);

        PreparedStatement
ps=cn.prepareStatement(str);

        for(int k=1;k<=count;k++)

        {

            ps.setString(k, (String)v.get(k-1));

        }

```



```

        int a=ps.executeUpdate();
    }

    catch(Exception e)
    {

        JOptionPane.showMessageDialog(null,e.getMessage());
    }

}

private void setTree()
{

    root1.removeAllChildren();

    java.util.Iterator
itr=Helper.getOracleTables(cn);

    while(itr.hasNext())
    {

        String n=(String)itr.next();

        DefaultMutableTreeNode temp=new
DefaultMutableTreeNode(n);

        setFields(temp,n);

        root1.add(temp);

    }

    tree1.updateUI();

```

```

    }

    private void setFields(DefaultMutableTreeNode
parent,String tableName)
    {

        try
        {

            PreparedStatement
ps=cn.prepareStatement("SELECT * FROM TAB
"+tableName);

            ResultSet rs=ps.executeQuery();

            while(rs.next())
            {

                parent.add(new
DefaultMutableTreeNode(rs.getString(1)));

            }

        }
        catch(Exception e){}

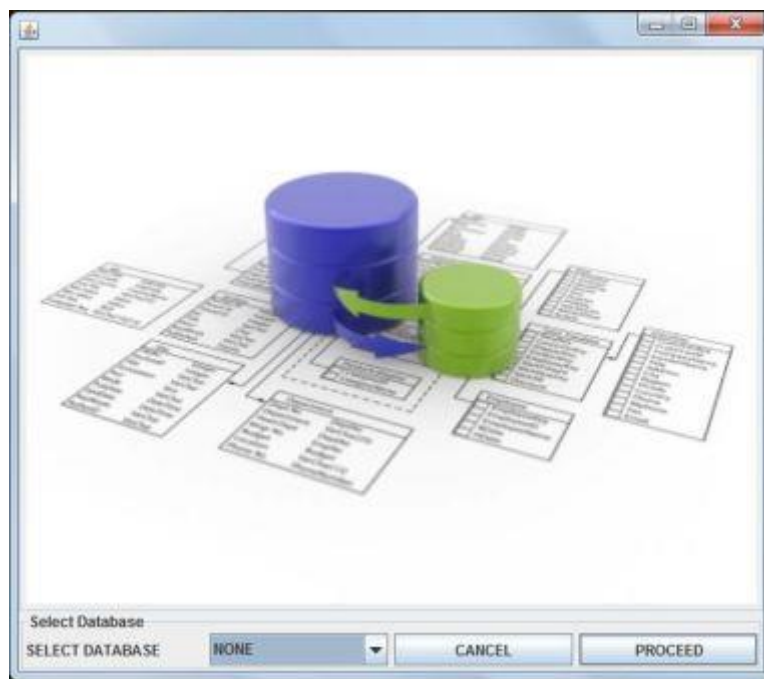
    }

}

```

## INPUT/OUTPUT SHOTS

# **ORACLE FORM SHOTS**



**PROCEED WITH ORACLE MODULE**



The image shows a screenshot of the 'Oracle Query Browser' window. At the top, there is a red banner with the 'ORACLE' logo in white. Below the banner, the text 'Connect to Oracle server instance' is displayed. The form contains five input fields: 'Server Host' with the value '127.0.0.1', 'Port' with the value '1521', 'SID NO.' with the value 'xe', 'Username' (empty), and 'Password' (empty). At the bottom of the window, there are four buttons: 'Details >>', 'O.K.', 'Clear', and 'Cancel'.

Connect to Oracle server instance	
Server Host	127.0.0.1
Port	1521
SID NO.	xe
Username	
Password	

Details >> O.K. Clear Cancel

## ORACLE LOGIN FORM



CREATE TABLE

create table

TABLE NAME

table

FIELD	TYPE	SIZE	CONSTRAINT

ADD ROW REMOVE ROW GENERATE

CREATE TABLE

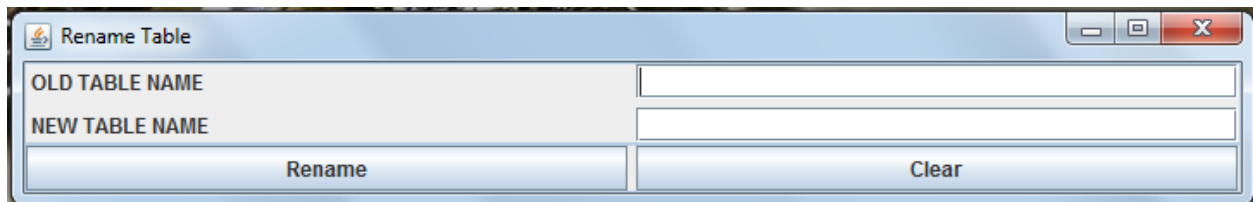






**ALTER TABLE**

# RENAME TABLE



A screenshot of a 'Rename Table' dialog box. The dialog has a title bar with a small icon and the text 'Rename Table'. It contains two input fields: 'OLD TABLE NAME' and 'NEW TABLE NAME'. Below these fields are two buttons: 'Rename' and 'Clear'.

Rename Table	
OLD TABLE NAME	<input type="text"/>
NEW TABLE NAME	<input type="text"/>
<input type="button" value="Rename"/>	<input type="button" value="Clear"/>

## **TESTING**

# **TESTING TECHNIQUES**

**Introduction:** Testing present an interesting anomaly. A series of the test are creating that are intended to “demolish” the software that has been created. Testing required that the developer discard preconceived notions of the “correctness” of the software just developed overcome a conflict of interest that accure when errors are uncovered.

**Testing Objective:** These are as follows-

- Testing is a process of the executing a program with the intent of the finding an error.
- A good test case is one that has a high probability of the finding an as-yet undiscovered error.
- A successful test is one that is one uncovers as-yet undiscovered error. If the testing is conducted successfully it will uncover errors in the software. Testing can not show the absence of the defects, it can only show that software errors are present.

**Testing principles-**

All tests should be traceable to customer requirements.

Tests should be planned long before testing begins.

The pareto principle applies to testing ( this principle implies that 80% of the all errors uncovered during testing likely be traceable to 20% of the of all program modules)".

Testing should begin "in the small" and progress toward testing "in the large".

Exhaustive testing is not possible.

## **TESTING-**

Since the faults can occur during any phase in the software development life cycle, different levels of testing are used in the testing process. Each level of testing aims to test different aspects of the system. Thus the system was put through the following tests:

### **Unit Testing**

In unit testing the programs that make up the system were tested. This level focuses first on the independent module to locate the error.

### **Integration testing**

In integration testing many modules, which passed the Unit, testing was combined into subsystems and was then tested.

## **System Testing**

In System Testing the entire Software system is tested to find any type of discrepancy that still exist between the designed system and the software requirements document.

## **Performance Testing**

In the test of performance time of inquiry was tested when the designed system was fully loaded with operating data.

Each and every input and output program is executed separately and tested for bugs. Whole of the system is executed and test data is parted and is fixing into the system and the output reports are compared with the parameters. Testing forms a very important part of the system development life cycle. It is basically this stage that helps the programmer to know all the possible flaws related with the working of his software and also helps to check the functioning of the various checks that have been used in the programs.

The complete project was tested in top down approach. For modular testing of programs the program are executed one by one and there inputs were taken out. The outputs thus formed were taken in the form of reports of the project, which is given in the later part of the documentation. The

reports thus formed were found out to be correct when compared with the data entered in the files.

### **Objective of System Testing:**

Once a system has been designed it is necessary to undergo an exhaustive to before installing the system. This is important because in some cases a small error not detected and corrected early before installation, may explode into a much large problem later on. Testing is being performed when users are asked to assist in identifying all possible situations. That might arise as regards the factor that efforts were put to tackle the problem under consideration. A plan was decided to be followed for testing the system. The complete testing procedure was divided into several steps, to be performed at different stages. Test was to be done as follows.

### **TESTING CRITERIA**

#### **White Box Testing :**

This testing is also known as “glass-box testing”. Using white-box testing, test cases are derived that

- Guarantee that all independent path with in a module have been exercised at last once.



- Exercise all logical decisions on their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.

### **Reasons for preferring White-box Testing:**

Logic errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed.

We often believe that a logical path is not likely to be executed when in fact it may be executed on a regular basis.

Topographical errors are random.

### **Transaction path Testing**

In this phase each and every condition within a unit program were tested. As and when a loop or condition statement were incorporated into a unit the loops were tested for correctness, for faulty condition and for not getting into infinite execution cycle. The data used was whatever necessary at the instance. The path of each transaction from origin to destination was tested for reliable results.

### **Module Testing**

This was carried out during the programming stage itself. Individual programs were tested at the time of coding and necessary changes are made there on to make sure that the tested at the time of coding and necessary changes are made there on to make sure that the modules in the form program is working satisfactory as regards the expected output from the module. All aspects of the program viz. all choices available are properly tested.

### **String Testing**

After loading all individual program string was performed for each one of programs where the output generated by one program is used as input by another program. This step was completed after making necessary changes wherever required.

### **Black Box Testing :**

This testing focuses on the functional requirements of the software i.e. it enables to derive set of the input conditions that will folly exercise all functional requirements for a program. It is not an alternative to white-box techniques rather it is a complementary approach i.e. likely to uncover a different class of than white-box Testing methods.

### **Black-box attempts to find errors the following categories:**

- ⌚ Incorrect or missing function,
- ⌚ Interface errors,
- ⌚ Error in data structure or external database access
- ⌚ Performance errors
- ⌚ Initialization and termination errors

### **System Testing**

After module and string testing, the systems were tested as whole system testes were undertaken to check bundled modules for errors. The errors founded in the couple system as a whole was corrected. A testing on the actual data of the company followed this. During this phase the existing system and this package was running in parallel to enable us to verify and compare the result sets. The following criteria were to be used while testing the system.

### **Output Testing**

No systems could be useful if it does not produced the required operation for that matter operation in the required format the output generated or displayed by the system under consider was tested by asking the format required by them.

## **User Acceptance Testing**

User acceptance of a system is a key factor for the success of any system. The system under consideration was tested for user acceptance by constantly keeping in touch with the prospected system user at time of developing and making changes.

## **Testing Procedure**

Different type of checks like duplicate checks, completeness check, validity checks etc. are incorporated in this system, as the data has to be entered in different cards.

The user is not familiar with new system the data entry screens are designed in such a way that they are:

- Consistent
- Compatible
- Easy to use
- Had quick response
- The following conventions are used while designing of the various screens to make the system user friendly.
- All the items that are logically related are together.
- For a particular card, query has been provided.
- Error and validation messages are provided wherever required.

- System testing is against its initial objectives, it is done either in a simulated environment.

### **Test Review**

Test review is the process, which ensures that testing is carried out, as planned test review decides whether or not the program is ready to ship out for the implementation. For each data entry screen, we prepared test data with extreme values and under all relevant data screen against real this process helps in rectifying the modules time.

### **Functional testing**

For this reason this testing is sometimes refers to as Black box testing in which the content of a black box is not known and function of black box is understood completely in terms of its inputs and outputs. We have used functional testing in our project. Functional testing refers to testing, which involves only observation of output for certain input values, and there is no attempt to analyze the code, which produces the output.

## **IMPLEMENTATION**

## **IMPLEMENTATION**

This section will be discussing the plan, which is adopted in implementation of the software. The description of the implementation plan in the organization is as follows:

In conversion from manual to computerize, the objective is to put the tested system into operation while holding cost, risks and personal irritation to a minimum. It involves creating:

- Computer capable files
- Training operation staff
- Installing terminals and hardware
- A critical aspect of conversion is not disrupting the functioning of organization. Conversion should be existing, because it is the last step before the candidate system begins to show result. Unfortunately the results of the conversion have been chaotic and traumatic for many firms. Unforeseen difficulties crop up as the system break down, data files are damaged and tempers grow short. The training package is frequently not complete and people are trying to figure out what to do. Much of stem from poor planning or no planning at all. Therefore to avoid such problems we will be using the "parallel run approach".

➤ The basic steps involved in the conversion are as follows:

- ✚ Conversion begins with a review of the project plan, the system test documentation and the plan.
- ✚ The conversion portion of the implementation plan is finalized and approach.
- ✚ Files are converted.
- ✚ Parallel processing between the existing system and the new system is initiated.
- ✚ Result of computer runs and operation for the new system are logged on a special form.
- ✚ Assuming no problem, parallel procession is discontinued. Implementation result are documented for reference Conversion is completed. Plan for the post implementation review are prepared.
- ✚ Following the review, the new system is officially operational.

The reason behind the use of this approach is that the software has been prepared and tested on a stand alone system. So, this software will be used along with the manual system in the organization for sometimes to test its functioning under real life conditions if there will be any problem with the software then necessary changes will be made in the software and tem it will be implemented for use.



## **MAINTENANCE**

## **MAINTENANCE**

Maintenance is a set of the software engineering activities that occur after software has been delivered to the customer and put into operation. Changes will occur because the software must be adapted to accommodate changes in its external environment (e.g. a change required because of operating system or peripheral devices) or because the customer requires functional or performance enhancements.

Software maintenance reapplies each of the preceding phases to an existing program rather than a new one.

All the four categories of the maintenance have been guaranteed to the system of the Annual Maintenance Contract:

### **Corrective Maintenance:**

Corrective maintenance means repairing, processing of performance failure or making changes because of previously incorrect or false assumptions.

### **Perfective Maintenance:**

Perfective Maintenance means enhancing the performance or modification the programmers to respond to users additional or changing needs. More money is spending on perfective than on corrective maintenance together.

Preventive maintenances are the process by which we prevent our system from becoming obsolete. Preventive maintenance involves the concept of re-engineering in which and system with an old technology is re-engineering into using new technology.

## **COST ESTIMATION**

## **COST ESTIMATION OF THE PROJECT**

Cost/Benefit analysis is performed to ascertain whether the costs in the system match with the benefits it will provide. The various kinds of costs incurred in the development and implementation of the system include:

✦ **Cost of Hardware:-** Cost incurred in purchase of hardware, i.e., computer and its peripherals.

✦ **Personnel Costs:-** Include The salaries of the EDP staff as well as perks of personnel involved in the development of the system.

✦ **Costs of Facilities:-** Expenses incurred in the preparation of the physical site. This includes flooring, wiring, lighting, etc.

✦ **Operating Costs:-** Include all the costs associated with the day to day operation of the system.

✦ **Supply Costs:-** Variable costs that increase with increased use of paper, ribbon, disks etc.

Under Cost/Benefit analysis of the **"SQL GENERATOR"** it was analyzed that the firm is ready to do one time investment but doesn't want high

## **SCOPE**

The idea of the project is to develop software for carrying out the following activities like:

- Easy to use Oracle on click event.
- Providing facility of the content as well as structure of a table.
- Generating report and Creating backup.
- It works as a navigator for Oracle -8i as it doesn't have navigator facility
- This s/w package occupies less storage space in comparison of Oracle Navigator.

## **FUTURE SCOPE AND FURTHER**

## **ENHANCEMENT OF THE PROJECT**

The candidate system developed is itself very flexible to modify itself with the needs of users and customers in the future. Yet the software is capable of enhancement. Few features, which can be later, introduce in the software for the enhancing it.

- More authentication responsibilities can be added in existing system.
- The system could be implemented that the user can create sequences.
- Programming facilities (PL/SQL) can be introduced such that a user can develop own program.

## **LIMITATIONS OF THE PROJECT**

The candidate system developed is itself very flexible to modify itself with the needs of users and customers in the future. Yet the software is capable of enhancement. Few features, which can be later, introduce in the software for the enhancing it.

- More authentication responsibilities can be added in existing system.
- The system could be implemented that the user can create sequences .
- Programming facilities can be introduced such that a user can develop own program.
- Using sub queries and joins.

The idea of the project is to develop software for caring out the following activities like:

- Easy to use Oracle on click event.
- Providing facility of the content as well as structure of a table.
- Delete or more field name from a table.
- Generating report and Creating backup.



## **Bibliography**

Here I would like to mention the name of the books and URLs used for reference while designing, testing, implementation, and coding of the system:

### **DESIGNING, TESTING, IMPLEMENTATION PHASE**

- System Analysis & Design, by Elias M. Ewad.
- An Integrated Approach to S/W Engineering, by Pankaj Jalote.
- Software Engineering, by Roger S. Pressman.
- Database System Concepts, by Kurt.
- Oracle SQL developer handbook by Mc Grow Hill.

### **CODING PHASE**

- Complete Reference-Java
- Black Book -Java
- [www.sunjava.com](http://www.sunjava.com)
- [www.w3schools.com](http://www.w3schools.com)