SCSX1056 - ORACLE & SQL

UNIT - I

DBMS-Database Management System

RDBMS-Relational Database Management System.

DBMS-
A Database Management System (DBMS) is a set of computer programs that controls thecreation, maintenance, and the use of the database in a computer platform or of an organization and its end users.A DBMS is a system software package that helps the use ofintegrated collection of data records and files known as databases. It allows different userapplication programs to easily access the same database.

Example-
Perl, Python,IMS.

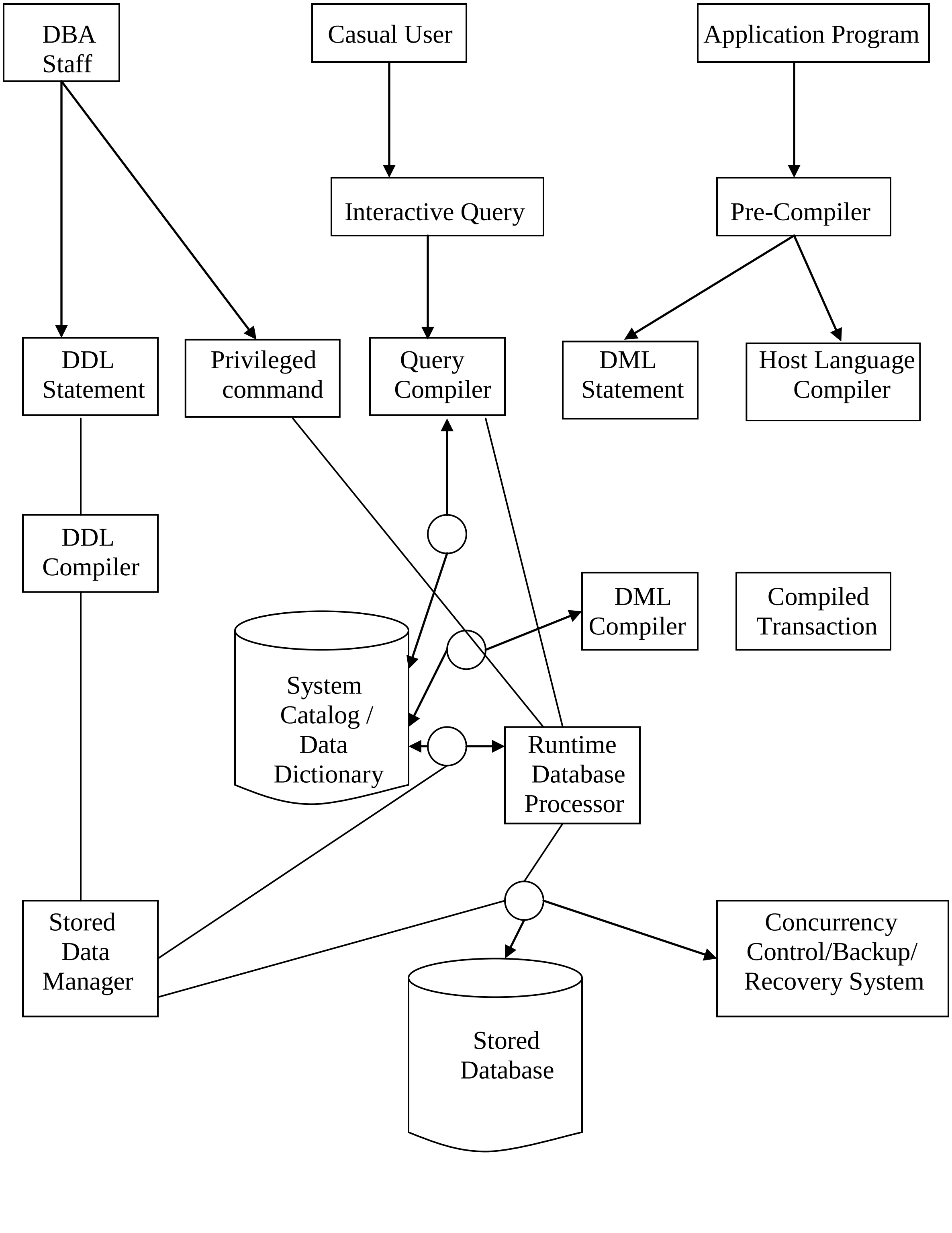
RDBMS-
A Relational Database Management System (RDBMS) is a Database Management System
(DBMS) that is based on the relational model .

RDBMS is a DBMS in which data is stored in the form of tables and the relationship among thedata is also stored in the form of tables.

Difference between DBMS and RDBMS-

1. Main difference between the DBMS and RDBMS is we can establish the relation between the tables in RDBMS.in case of DBMS we cannot establish relation between tables.
2. Another difference between dbms and rdbms is that RDBMS follows relational data model where as dbms can follow even network, hierarchical or other data model.
3. The relational model is based on set theory ,its accuracy and usefulness have a basis in mathematics.
4. In RDBMS, if we open the Database, the whole tables can be accessable, where in DBMS, we need to open each and every table whenever we want to use.
5. RDBMS allows the data to be queried based on any column in any table not need to create an index or keys in order to query data.So, relational data is easier to query than hierarchical, CODASYL, or some other model.

ORACLE ENVIRONMENT:



INTRODUCTION TO ORACLE

Mr.E.F.T Codds applied the principle of relationship in statistics to data management and came up with twelve laws using mathematics. He proved that if all these 12 laws were incorporated into database core technology, there would be a revolution in the speed of any database management system while managing data even when used on network operating system. Oracle implements around seven of Codds laws. Since, E.F.T Codd was a satisfaction and in statistic relationships play a key role. The newer product that came into existence in which several Codds laws were used was called “Relational Database Management Systems” (RDBMS) for short E.F.T ted Codds law for a fully functional database management system.

Information Representation:

All information stored in relational database is represented only by the data item values which are stores in the table that make up the database. Association between data items are not logically represented in any other way such as by the user of pointer from one table to the other.

Logically Accessibility:

Every data item value stored is a relational database is accessible by sharing the name of the table. It stored in the name of the column under which it is stored and the values of the primary key that defines the rows in which it is stored.

The Systematic Treatment Of Null Values:

The database management system has a method for representing null values for example small values for numeric data must be distinction zero and any other numeric value and for character data, it must be different from a string of blank or any other character value.

The Dynamic Online Catalog Facilities:

The logical description of a relation database is represented in the same ordinary data. This is done so that the relation management system itself can be used to maintain database description.

Comprehensive Data Sublanguage Rule:

A relational database management system may support many types of language for describing data and accessible the database. However, there must be atleast one language that we use ordinary character string to support the definition of the data, the definition of views, the manipulation of data constraint on data integrity, information concerning authorization and the boundaries for recovery of units.

Views Updateability:

Any view that can be defined using combination of base table that is theoretically updateable is capable of being updated by the Relational Database Management System.

Insert / Update and Delete:

Any operand that describes the result of a single retrieval operation is capable of being applied to as insert, update, or delete operation as null.

Physical data independence:

Changes made to physical storage representation or access methods do not require changes to be made to application programs.

Logical Data Independence:

Changes made to the table that do not modify any stored in that table do not require changes to be made to application programs.

Integrity Constraint:

Constraints that apply to entity integrity and referential integrity are specificable by the data implemented by the Database Management System and not by the statement coded into the application program.

Database Distribution:

The data language implement by the Retrieval Database Management System supports the ability to distribution that the database without requiring changes to be made to application program. These facilities must be provided in the data language whether or not the Database Management System itself support distributed database. If the relational management system supports facilities that allow application program to operate on the table as row at a time an application program using this type of database access is prevented from bypassing entity integrity on referential integrity constraint that are defined for the database.

ORACLE INTRODUCTION:

Oracle Corporation was the first company to offer a true relational database management system. Commercially led innovation in the field of RDBMS. The oracle corporation strategy of offering a RDBMS that is portable, compactable and connectable resulted in a very powerful tool for user.

When we speak of Oracle, we are speaking of a single product on-program Oracle RDBMS let you manipulation on Oracle database. Many of these products are fourth generation operational language tool that let your interactive screen to create application programs.

Overview Oracle Architecture:

To understand about RDBMS, oracle let you first study the architecture. The Oracle is organized four levels. The Kernel provides communication with the database and connection with other kernels in or distributed environment.

The database dictionary provides details of the database objects such as tables, columns, views etc. the oracle uses the right and privilege of the object. The dictionary is made up of tables and views. They created at the time of oracle installation and are manipulated through SQL server. The tables in the dictionary are not accessible to the user to maintain the integrity of the database.

Oracle provides the following advantage over the relational database:

- Direct SQL Interface to the database through SQL plus lets developers and user to interact with the database and manipulate.

- The forms developers SQL form let us to provide the prototype application quickly.

- The probability of application developed using SQL form to any platform requires running Oracle gives on us flexibility to develop application on a PC for production oracle is available for many interfaces minicomputer, mainframe and microcomputer platforms.

- The transferability of data files from many files and flat file format into table structure of an oracle database using SQL loader reduces the problem in database.

- The oracle capability to interface with the variety of third generation language such as COBOL, PASCAL, C helps to create Oracle application based on any of those language.

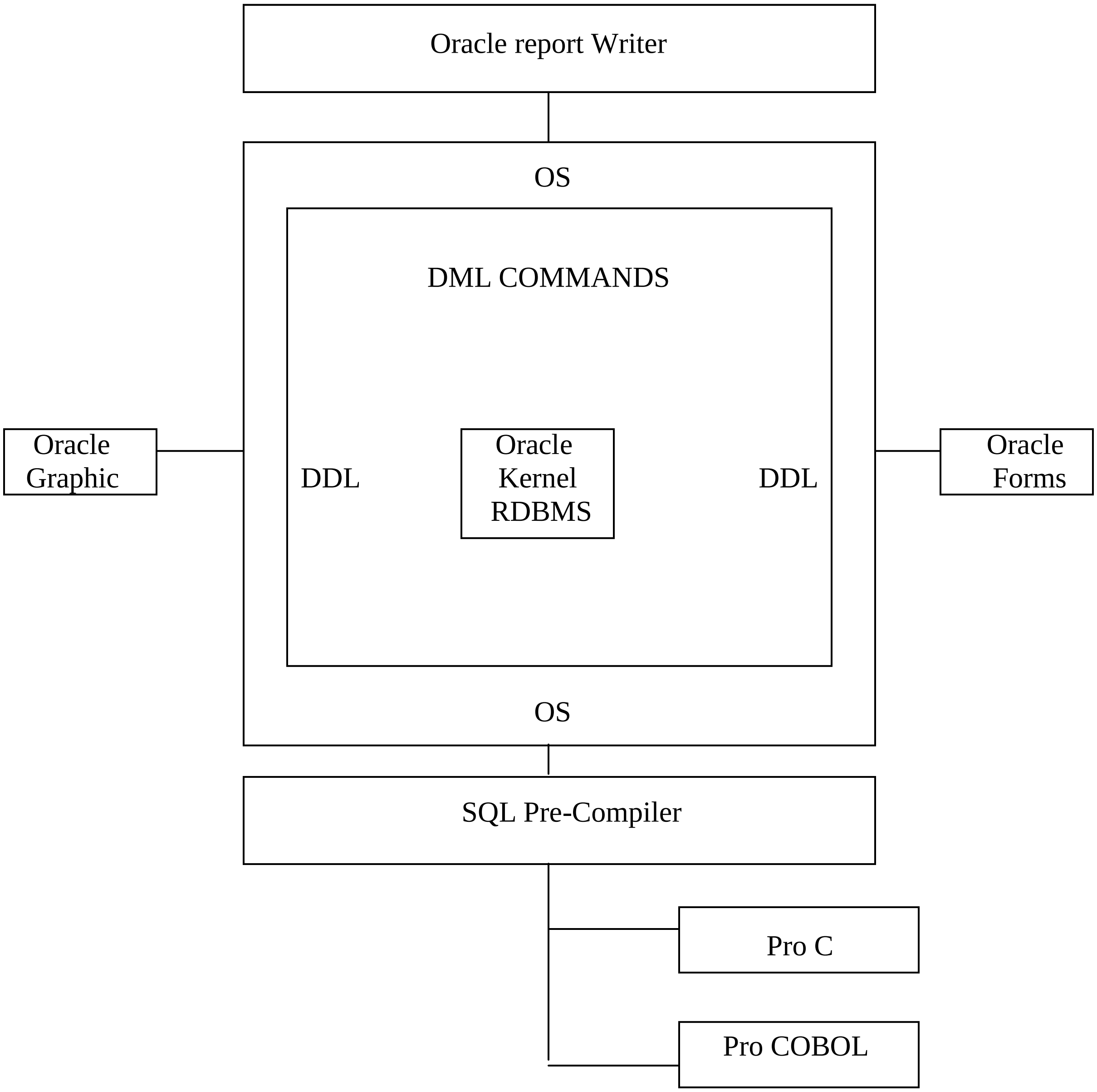
Oracle database allows the utilities:

- The database structure that is easy to visualize and understand.

- The ability to create any number of temporary relationships between tables.

- Freedom from concern about how to query the database through the use of SQL.

- Tables easy to visualize.



DATA MODELS

A database model is a theory or specification describing how a database is structured and used. Several such models have been suggested.

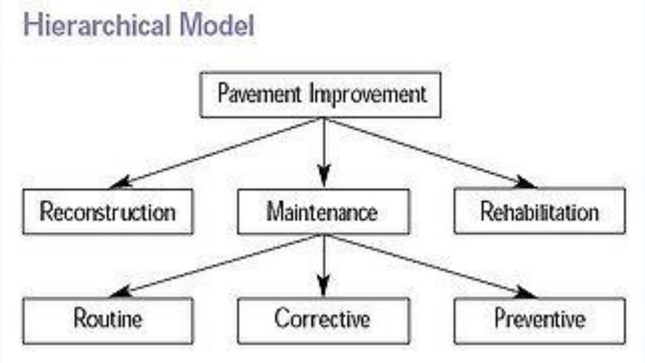
Common models include:

- Hierarchical model
- Network model

- Relational model
- A data model is not just a way of structuring data: it also defines a set of operations that can be performed on the data. The relational model, for example, defines operations such as select, project, and join. Although these operations may not be explicit in a particular query language, they provide the foundation on which a query language is built.

Hierarchical model

- A **hierarchical data model** is a data model in which the data is organized into a tree-like structure. The structure allows repeating information using parent/child relationships: each parent can have many children but each child only has one parent. All attributes of a specific record are listed under an entity type.



- Example of a Hierarchical Model.
- In a database, an entity type is the equivalent of a table; each individual record is represented as a row and an attribute as a column. Entity types are related to each other using *1: N* mapping, also known as one-to-many relationships.
- The most recognized and used hierarchical database is IMS developed by IBM.

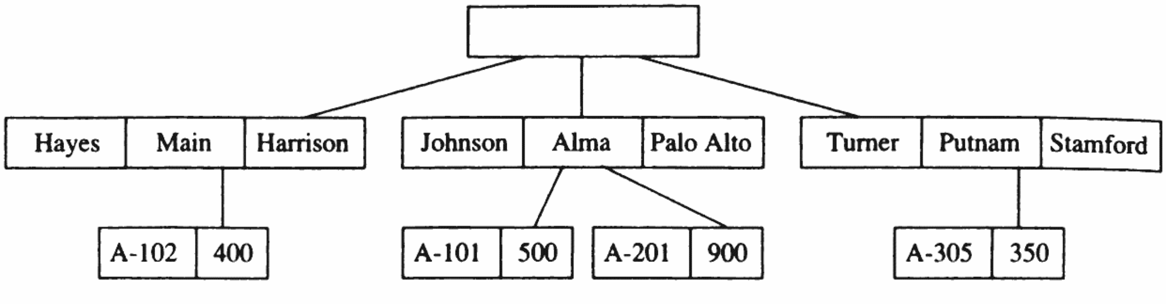
Example

- An example of a hierarchical data model would be if an organization had records of employees in a table (entity type) called "Employees". In the table there would be attributes/columns such as First Name, Last Name, Job Name and Wage. The company also has data about the employee’s children in a separate table called "Children" with attributes such as First Name, Last Name, and date of birth. The Employee table represents a parent segment and the Children table represents a Child segment. These two segments form a hierarchy where an employee may have many children, but each child may only have one parent.

- Consider the following structure:

EmpNo	Designation	ReportsTo
10	Director	
20	Senior Manager	10
30	Typist	20
40	Programmer	20

- In this, the "child" is the same type as the "parent". The hierarchy stating EmpNo 10 is boss of 20, and 30 and 40 each report to 20 is represented by the "ReportsTo" column. In Relational database terms, the ReportsTo column is a foreign key referencing the EmpNo column. If the "child" data type were different, it would be in a different table, but there would still be a foreign key referencing the EmpNo column of the employees table.



Sample database

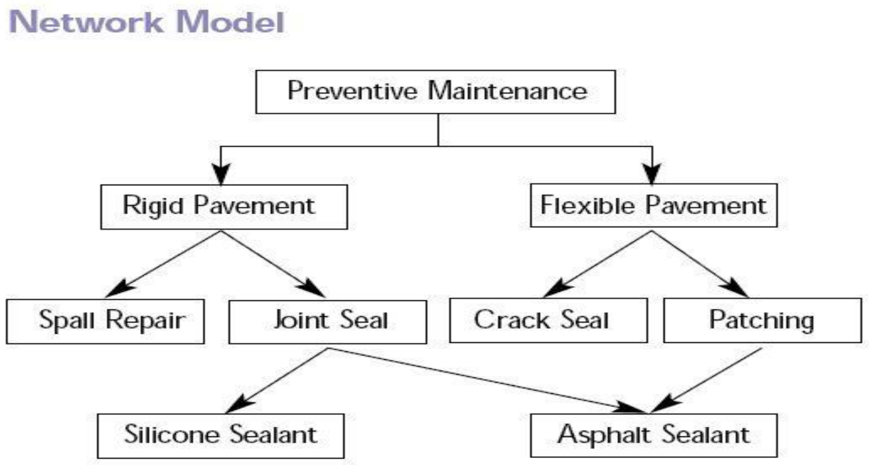
Network Model

The network model is a database model conceived as a flexible way of representing objects and their relationships.

Network Model – Basic Concepts

o **Data are represented as *collection of records***

o **Relationships are represented as *links***
o **Each record is a collection of fields:**



DDL (DATA DEFINITION LANGUAGE) COMMANDS

DDL OPERATIONS EXAMPLES:

1. To create an employee table with ename, eid, doj, basic pay, age, dept.

Create table employee (ename varchar2 (15), eid number (5), doj date, basicpay number (8,2), age number (3));

2. To include one more field address (varchar2 (15)).
Alter table employee add (address varchar2 (15));

3. Describe the table.
Desc employee;

4. Modify the width of the address field.
Alter table employee modify (address varchar2 (20));

5. Drop the table employee.
Drop table employee;

2. DML (DATA MANIPULATION LANGUAGE) COMMANDS.

DML OPERATIONS EXAMPLES:
1. Insert five records to the employee database.
Insert into employee (ename, eid, doj, basicpay, age, address) values (‘&ename’, &eid,‘&doj’, &basicpay, &age, ‘&address’);

2. Increment the basic pay of all the employees by 5% of their basic pay.
Update employee set basicpay = basicpay + basicpay * 5 / 100;

3. Delete the third employee record from the table.

```
Delete from employee where eid=3;
```

3. **DQL (DATA QUERY LANGUAGE) COMMANDS**.

DQL OPERATIONS EXAMPLES:

Create a student detail database with roll number, name, date of birth, course, department,semester, percentage.

```
Create table student (rno number (3), sname varchar2 (15), dob date, course varchar2
(6),dept varchar2 (4), sem varchar2 (3), per number (5,2));
```

Insert 5 records into the database and perform the following operations.

```
Insert into student (rno, sname, dob, course, dept, sem, per) values (&rno, ‘&sname’,
‘&dob’,‘&course’, ‘&dept’, ‘&sem’, &per);
```

1. Select all Records.

```
Select * from student;
```

2. Select name, percentage from the records.

```
Select sname, per from student;
```

3. Select distinct course.

```
Select distinct course from student;
```

4. Select details of all students with percentage greater than 75.

```
Select * from student where per > 75;
```

5. Select roll number and name of all records sorted in order by percentage.

```
Select rno, sname from student order by per;
```

6. Select name of students belonging to 3rd semester MBA degree.

```
Select sname from student where sem = ‘III’ and course = ‘mba’;
```

7. Select name, course and semester of all students belonging to 2 different departments.

```
Select sname, course, sem from student where dept in (‘be’, ‘mca’);
```

8. Select name, course and semester of all students not in MBA department.

```
Select sname, course, sem from student where dept not in (‘mba’);
```

9. Select name, percentage of all students with percentage between 85 & 90.

```
Select sname, per from student where per between 85 and 90;
```

10. Select name, course of all students whose department is in MBA.

```
Select sname, course from student where dept in (‘mba’);
```

11. Select name, course for all students whose department is not in IT, CSE, ECE.

```
Select sname, course from student where dept not in (‘it’, ‘cse’, ‘ece’);
```

12. Select all records where name is start with alphabet “A”.

```
Select * from student where sname like ‘A%’;
```

4. TCL (TRANSACTION CONTROL LANGUAGE) COMMANDS.

TCL OPERATIONS EXAMPLES:

1. In the student database make the transaction as permanent using Commit command.

```
Select * from student;

Commit;
```

2. Undo the transaction performed in the database using Rollback.

```
Rollback;
```

3. Select a point and implement Savepoint command which is later rollbacked to.

```
Select * from student;

Savepoint sp1;

Delete from student where rno = 101;

Rollback to spl;
```