Unit 2 Database System Architecture

Structure:

2.1 Introduction

The architecture of a database system is greatly influenced by the underlying computer system on which the database system runs. Database systems can be centralized, or client–server, where one server machine executes work on behalf of multiple client machines. Database systems can also be designed to exploit parallel computer architectures. Distributed

databases span multiple geographically separated machines. Distributed query processing and directory systems are also described in this unit.

The architecture of a database system is greatly influenced by the underlying computer system on which it runs, in particular by such aspects of computer architecture as networking, parallelism, and distribution:

- Networking of computers allows some tasks to be executed on a server system, and some tasks to be executed on client systems.
- Parallel processing within a computer system allows database-system activities to be speeded up, allowing faster response to transactions, as well as more transactions per second.
- Distributing data across sites or departments in an organization allows those data to reside where they are generated or most needed, but still to be accessible from other sites and from other departments.

Also the architecture of DBMS packages has evolved form the early monolithic systems, where the whole DBMS software package was one tightly integrated system, to the modern DBMS packages that are modular in design, with a client/server system architecture. A DBMS can be considered as a buffer between application programs, end-users and a database designed to fulfill features of data independence. In 1975 the American National Standards Institute Standards Planning and Requirements Committee (ANSI-SPARC) proposed a **three-level architecture** for this buffer. This architecture identified three levels of abstraction. These levels are sometimes referred to as schemas or views.

Objectives:

By the end of Unit 2, the learners are able to understand

- Three views of data i.e., three levels of architecture of DBMS
- Architectures of MySQL, SQL server and Oracle architecture
- Database management systems - facilities and structure
- Database manager and database administrator
- Distributed processing and client server architecture.

2.2 The Three Level Architecture of DBMS

In the previous section we defined three levels of abstraction at which the database may be viewed. A database management system that provides these three levels of data is said to follow three-level architecture as shown in figure 2.1. These three levels are the external level, the conceptual level and the internal level.

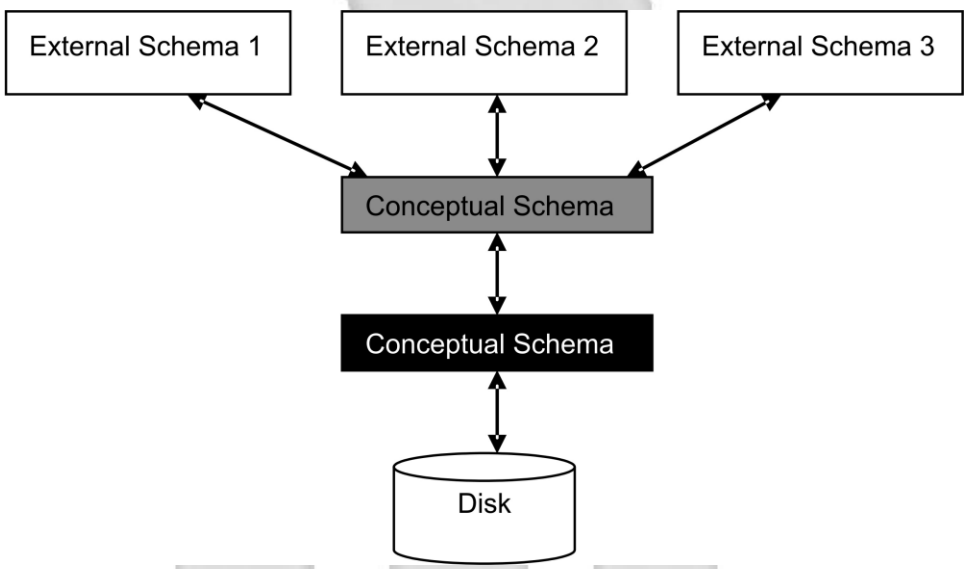

Figure 2.1: The three level architecture for a DBMS

The view at each of these levels is described by a schema. A schema as mentioned earlier is an outline or a plan that describes the records and relationships existing in the view. The schema also describes the way in which entities at one level of abstraction can be mapped to the next level. The overall design of the database is called the database schema. A database schema includes such information as:

- Characteristics of data items such as entities and attributes
- Logical structure and relationship among those data items
- Format for storage representation
- Integrity parameters such as physical authorization and backup politics.

The concept of a database schema corresponds to programming language notion of type definition. A variable of a given type has a particular value at a given instant in time. The concept of the value of a variable in Programming languages corresponds to the concept of an instance of a database schema.

Since each view is defined by a schema, there exists, several schema in the database, and these schema are partitioned following three levels of data abstraction or views. At the lower level we have the physical schema; at the intermediate level we have the conceptual schema, while at the higher level we have a subschema. In general, database system supports one physical schema, one conceptual schema and several **subschema**.

2.2.1 External Level or Subschema

The external level is at the highest level of database abstraction where only those portions of the database of concern to a user or application program are included. Any number of user views (some of which may be identical) may exist for a given global or conceptual view.

Each external view is described by means of a schema called an external schema or subschema. The external schema consists of the definition of the logical records and the relationships in the external view. The external schema also contains the method of deriving the objects in the external view from the objects in the conceptual view. The object includes entities, attributes, and relationships.

2.2.2 Conceptual Level or Conceptual Schema

At this level of database abstraction all the database entities and the relationships among them are included. One conceptual view represents the entire database. This conceptual view is defined by the conceptual schema. It describes all the records and relationships included in the conceptual view and, therefore, in the database. There is only one conceptual schema per

database. This schema also contains the method of deriving the objects in the conceptual view from the objects in the internal view.

The description of data at this level is in a format independent of its physical representation. It also includes features that specify the checks to retain data consistency and integrity.

2.2.3 Internal Level or Physical Schema

We find this view at the lowest level of abstraction, closest to the physical storage method used. It indicates how the data will be stored and describes the data structures and access methods to be used by the database. The internal view is expressed by the internal schema, which contains the definition of the stored record, the method of representing the data fields, and the access aids used.

2.2.4 Mapping

Two mappings are required in a database system with three different views as shown in figure 2.1. A mapping between the external and conceptual level gives the correspondence among the records and the relationships of the external and conceptual levels.

Similarly, there is a mapping from a conceptual record to an internal one. An internal record is a record at the internal level, not necessarily a stored record on a physical storage device. The internal record of figure 2.2 may be split up into two or more physical records. The physical database is the data that is stored on secondary storage devices. It is made up of records with certain data structures and organized in files. Consequently, there is an additional mapping from the internal record to one or more stored records on secondary storage devices.

Self Assessment Questions 2.2

1. The overall design of the database is called the _____ schema.
2. In general, database system supports one physical schema, one conceptual schema and several _____.
3. The highest level of abstraction as seen by a user is called as _____view.
4. _____ level describes what data are actually stored in the database.
5. There is only _____ conceptual schema per database.
6. The physical database is the data that is stored on _____ storage devices.

2.3 MySQL Architecture

MySQL is based on a tiered architecture, consisting of both primary subsystems and support components, that interact with each other to read, parse, and execute queries, and to cache and return query results.

Primay Subsystems

The MySQL architecture consists of five primary subsystems that work together to respond to a request made to the MySQL database server:

- The Query Engine
- The Storage Manager
- The Buffer Manager
- The Transaction Manager
- The Recovery Manager

The organization of these features is shown in Figure 2.2. We'll explain each one briefly to help you gain a better understanding of how the parts fit together.

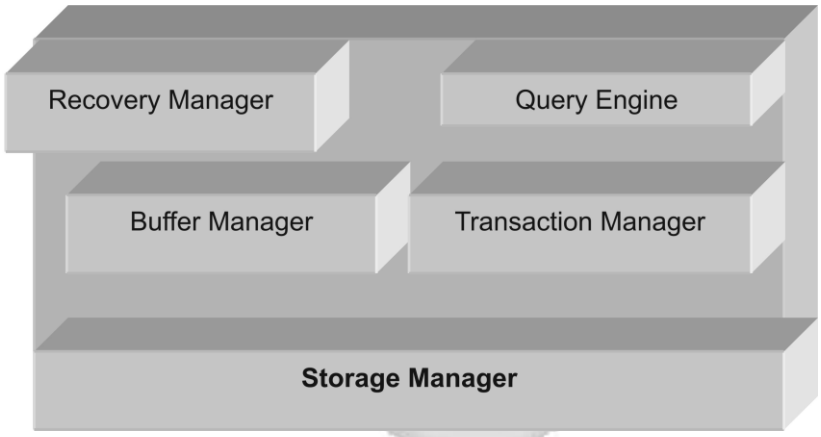

Figure 2.2: Architecture of MySQL

The Query Engine: This subsystem contains three interrelated components:

- The Syntax Parser
- The Query Optimizer
- The Execution Component

The Syntax Parser decomposes the SQL commands it receives from calling programs into a form that can be understood by the MySQL engine. The objects that will be used are identified, along with the correctness of the syntax. The Syntax Parser also checks the objects being referenced to ensure that the privilege level of the calling program allows it to use them.

The Query Optimizer then streamlines the syntax for use by the Execution Component, which then prepares the most efficient plan of query execution. The Query Optimizer checks to see which index should be used to retrieve the data as quickly and efficiently as possible. It chooses one from among the several ways it has found to execute the query, and then creates a plan of execution that can be understood by the Execution Component.

The Execution Component then interprets the execution plan and, based on the information it has received, makes requests of the other components to retrieve the records.

The Storage Manager: The Storage Manager interfaces with the operating system (OS) to write data to the disk efficiently. Because the storage functions reside in a separate subsystem, the MySQL engine operates at a level of abstraction away from the operating system. This means that if you port to a different operating system that uses a different storage mechanism, for example, you can rewrite only the storage portion of the code while leaving the rest of the engine as it is. With the help of MySQL's Function Libraries, the Storage Manager writes to disk all of the data in the user tables, indexes, and logs as well as the internal system data.

The Buffer Manager: This subsystem handles all memory management issues between requests for data by the Query Engine and the Storage Manager. MySQL makes aggressive use of memory to cache result sets that can be returned as-is rather than making duplicate requests to the Storage Manager; this cache is maintained in the Buffer Manager.

This is also the area where new records can be cached while waiting for availability of targeted tables and indexes. If any new data is needed, it's requested from the Storage Manager and placed in the buffer, before then being sent to the Query Engine.

The Transaction Manager: The function of the Transaction Manager is to facilitate concurrency in data access. This subsystem provides a locking facility to ensure that multiple simultaneous users access the data in a consistent way, without corrupting or damaging the data in any way. Transaction control takes place via the Lock Manager subcomponent, which places and releases locks on various objects being used in transactions.

Each transactional table handler implements its own Transaction Manager to handle all locking and concurrency needs.

The Recovery Manager: The Recovery Manager's job is to keep copies of data for retrieval later, in case of a loss of data. It also logs commands that modify the data and other significant events inside the database.

Self Assessment Questions 2.3

1. The MySQL architecture consists of _____ primary subsystems.
2. The _____ decomposes the SQL commands it receives from calling programs into a form that can be understood by the MySQL engine.
3. The _____ interfaces with the operating system (OS) to write data to the disk efficiently.
4. Buffer manager subsystem handles all memory management issues between requests for data by the _____ and the Storage Manager.
5. Each transactional table handler implements its own _____ to handle all locking and concurrency needs.

2.4 SQL Server 2000 Architecture

Microsoft® SQL Server 2000 is a family of products that meet the data storage requirements of the largest data processing systems and commercial Web sites, yet at the same time can provide easy-to-use data storage services to an individual or small business.

The data storage needs of a modern corporation or government organization are very complex. Some examples are:

- Online Transaction Processing (OLTP) systems must be capable of handling thousands of orders placed at the same time.
- Increasing numbers of corporations are implementing large Web sites as a mechanism for their customers to enter orders, contact the service department, get information about products, and for many other tasks

that previously required contact with employees. These sites require data storage that is secure, yet tightly integrated with the Web.

- Organizations are implementing off-the-shelf software packages for critical services such as human resources planning, manufacturing resources planning, and inventory control. These systems require databases capable of storing large amounts of data and supporting large numbers of users.
- Organizations have many users who must continue working when they do not have access to the network. Examples are mobile disconnected users, such as traveling sales representatives or regional inspectors. These users must synchronize the data on a notebook or laptop with the current data in the corporate system, disconnect from the network, record the results of their work while in the field, and then finally reconnect with the corporate network and merge the results of their fieldwork into the corporate data store.
- Managers and marketing personnel need increasingly sophisticated analysis of trends recorded in corporate data. They need robust Online Analytical Processing (OLAP) systems easily built from OLTP data and support sophisticated data analysis.
- Independent Software Vendors (ISVs) must be able to distribute data storage capabilities with applications targeted at individuals or small workgroups. This means the data storage mechanism must be transparent to the users who purchase the application. This requires a data storage system that can be configured by the application and then tune itself automatically so that the users do not need to dedicate database administrators to constantly monitor and tune the application.

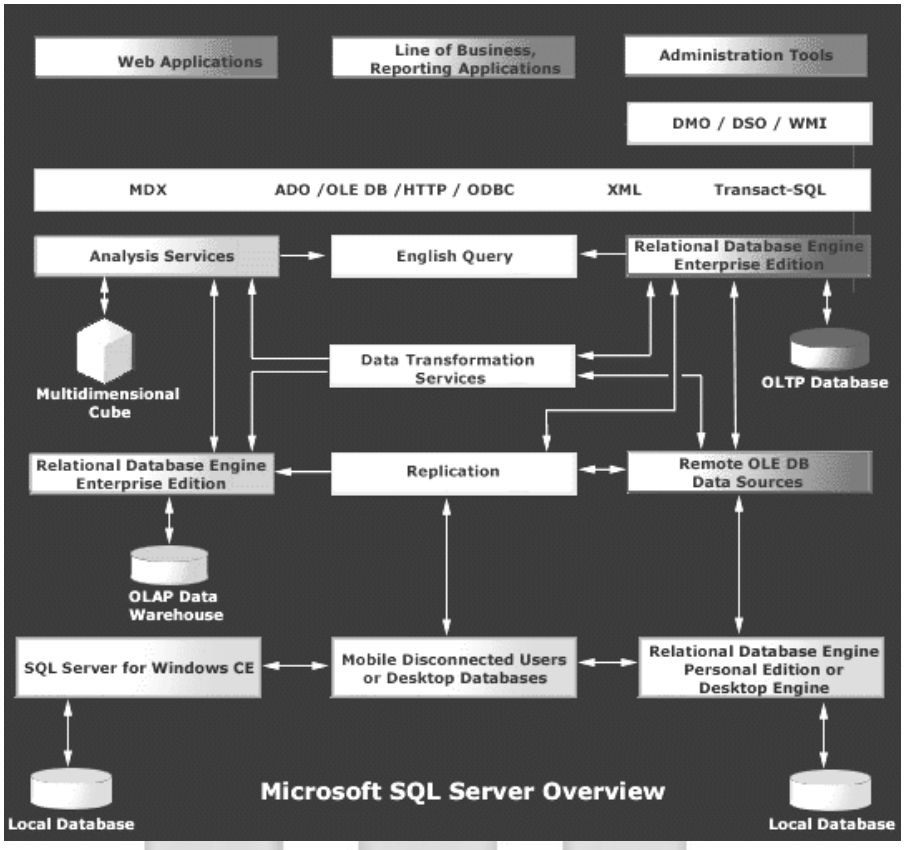

Figure 2.3: SQL Server Architecture

Self Assessment Questions 2.4

1. _____ systems must be capable of handling thousands of orders placed at the same time.
2. In _____ the data storage mechanism must be transparent to the users who purchase the application.

2.5 Oracle Architecture

The Oracle server consists of physical files and memory components. The Oracle 9i Database product is made up of three main components namely:

- **The Oracle Server** – This is the Oracle database management system that is able to store, manage and manipulate data. It consists of all the

files, structures, processes that form Oracle Database 9i. The Oracle server is made up of an Oracle instance and an Oracle database.

- **The Oracle Instance** – Consists of the memory components of Oracle and various background processes.
- **The Oracle Database** – This is the centralized repository where the data is stored. It has a physical structure that is visible to the Operating system made up of operating system files and a logical structure that is recognized only by the Oracle Server.

Figure 2.4 displays the architecture of the Oracle Database 9i. It is broadly divided into memory components which form the Oracle instance and the physical database components, where different kinds of data are stored.

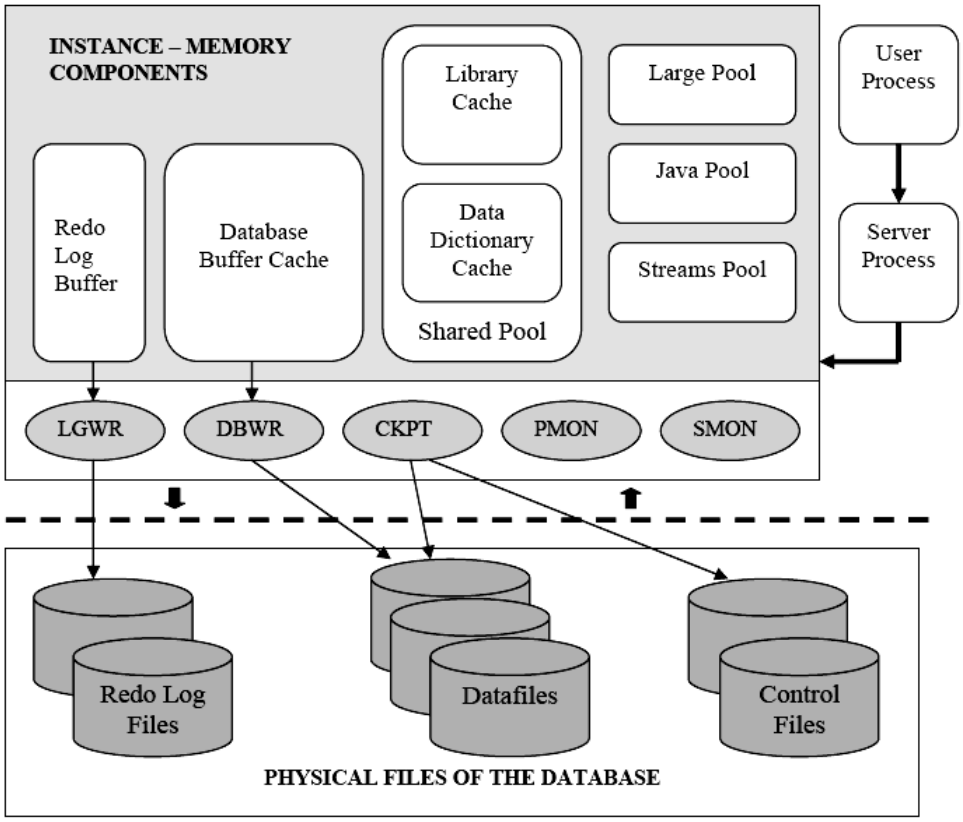

Figure 2.4: Oracle Architecture

Self Assessment Questions 2.5

1. Oracle 9i Database product is made up ———————— main components.
2. The ———————— consists of physical files and memory components.
3. The ———————— consists of the memory components of Oracle and various background processes.

2.6 Database Management System Facilities

Two main types of facilities are supported by the DBMS: the data definition facility or data definition language (DDL), the data manipulation facility or data manipulation language (DML)

2.6.1 Data Definition Language

Database management systems provide a facility known as the data definition language (DDL), which can be used to define the conceptual schema and also give some details about how to implement this schema in the physical devices used to store the data. This definition includes all the entity sets and their associated attributes, as well as the relationships among the entity sets. The definition also includes any constraints that have to be maintained, including the constraints on the value that can be assigned to a given attribute, and the constraints on the values assigned to different attributes in the same or different records. These definitions, which can be described as metadata about the data in the database, are expressed in the DDL of the DBMS and maintained in a compiled form (usually as a set of tables). The compiled form of the definitions is known as a data dictionary, directory, or system catalogue. The data dictionary contains information on the data stored in the Database Base Concepts and is consulted by the DBMS before any data manipulation operation.

The database management system maintains the information on the file structure, the method used to efficiently access the relevant data (i.e., the access method). It also provides a method whereby the application

programs indicate their data requirements. The application program could use a subset of the conceptual data definition language or a separate language. The database system also contains mapping functions that allow it to interpret the stored data for the application program.

The internal schema is specified in a somewhat similar data definition language called data storage definition language. The definition of the internal view is compiled and maintained by the DBMS. The compiled internal schema specifies the implementation details of the internal database, including the access methods employed. This information is handled by the DBMS; the user need not be aware of these details.

2.6.2 Data Manipulation Language

DML is a language that enables users to access or manipulate as organized by the appropriate data model. Data manipulation involves retrieval of data from the database, insertion of new data into the database, and deletion or modification of existing data. The first of these data manipulation operations is called a query. A query is a statement in the DML that requests the retrieval of data from the database. The subset of the DML used to pose a query is known as a query language; however, we use the terms DML and query language synonymously.

The DML provides commands to select and retrieve data from the database. Commands are also provided to insert, update, and delete records. They could be used in an interactive mode or embedded in conventional programming languages such as Assembler, COBOL, FORTRAN, Pascal, or PL/I. The data manipulation functions provided by the DBMS can be invoked in application programs directly by procedure calls or by preprocessor statements. The latter would be replaced by appropriate procedure calls by either a preprocessor or the compiler.

There are basically **two types** of DML:

Procedural: which requires a user to specify what data is needed and how to get it.

Nonprocedural: which requires a user to specify what data is needed without specifying how to get it.

Data definition of the external view in most current DBMSs is done outside the application program or interactive session. Data manipulation is done by procedure calls to subroutines provided by a DBMS or via preprocessor statements. In an integrated environment, data definition and manipulation are achieved using a uniform set of constructs, that forms part of the user's programming environment.

Self Assessment Questions 2.6

1. Database management systems provide a facility known as the _____, which can be used to define the conceptual schema.
2. DML is a language that enables users to access or manipulate as organized by the appropriate _____.
3. _____ DML requires a user to specify what data is needed and how to get it.

2.7 Database Management System Structure

The major components of a DBMS are explained below:

DML Precompiler: It converts DML statement embedded in an application program to normal procedure calls in the host language. The precompiler must interact with the query processor in order to generate the appropriate code.

DDL Compiler: The DDL compiler converts the data definition statements into a set of tables. These tables contain information concerning the

database and are in a form that can be used by other components of the DBMS.

File Manager: File manager manages the allocation of space on disk storage and the data structure used to represent information stored on disk. The file manager can be implemented using an interface to the existing file subsystem provided by the operating system of the host computer, or it can include a file subsystem written especially for the DBMS.

2.7.1 Database Manager

Databases typically require a large amount of storage space. Corporate databases are usually measured in terms of gigabytes of data. Since the main memory of computers cannot store this information, it is stored on disks. Data is moved between disk storage and main memory as needed. Since the movement of data to and from disk is slow relative to the speed of control processing unit of computers, it is imperative that database system structure data so as to minimize the need to move data between disk and main memory. A database manager is a program module which provides the interface between the low level data stored in the database and the application programs and queries submitted to the system. It is responsible for interfacing with file system.

One of the functions of database manager is to convert user's queries coming directly via the query processor or indirectly via an application program from the user's logical view to the physical file system. In addition, the tasks of enforcing constraints to maintain the consistency and integrity of the data as well as its security are also performed by database manager. Synchronizing the simultaneous operations performed by concurrent users is under the control of the data manager. It also performs backup and recovery operations. Let us now summarize the important responsibilities of Database manager:

- **Interaction with file manager:** The raw data is stored on the disk using the file system which is usually provided by a conventional operating system. The database manager translates the various DML statements into low-level file system commands. Thus the database manager is responsible for the actual storing, retrieving and updating of data in the database.
- **Integrity enforcement:** The data values stored in the database must satisfy certain types of consistency constraints. For example, the balance of a bank account may never fall below a prescribed amount (for example Rs. 200). Similarly the number of holidays per year an employee may be having should not exceed 25 days. These constraints must specified explicitly by the DBA. If such constraints are specified, then the database manager can check whether updates to the database result in the violation of any of these constraints and if so appropriate action may be imposed.
- **Security enforcement:** As discussed above, not every user of the database needs to have access to the entire content of the database. It is the job of the database manager to enforce these security requirements.
- **Backup and recovery:** A computer system like any other mechanical or electrical device is subject to failure. There are a variety of causes of such failure, including disk crash, power failure and s/w errors. In each of these cases, information concerning the database is lost. It is the responsibility of the database manager to detect such failures, and restore the database to a state that existed prior to the occurrence of the failure. This is usually accomplished through the backup and recovery procedures.
- **Concurrency Control:** When several users update the database concurrently, the consistency of data may no longer be preserved. It is

necessary for the system to control the interaction among the concurrent users, and achieving such a control is one of the responsibilities of the database manager.

Query Processor: The database user retrieves data by formulating a query in the data manipulation language provided with the database. The query processor is used to interpret the online user's query and convert it into an efficient series of operations, in a form capable of being sent to the data manager for execution. The query processor uses the data dictionary to find the structure of the relevant portion of the database, and uses this information in modifying the query, and preparing an optimal plan to access the database.

2.7.2 Database Administrator

One of the main reasons for having a database management system is to have control of both the data and programs accessing that data. The person having such control over the system is called the database administrator (DBA). The DBA administers the three levels of the database and, in consultation with the overall user community, sets up the definition of the global view or conceptual level of the database. The DBA further specifies the external view of the various users and applications, and is responsible for the definition and implementation of the internal level, including the storage structure and access methods to be used for the optimum performance of the DBMS. Changes to any of the three levels necessitated by changes or growth in the organization and/or emerging technology are under the control of the DBA.

Mappings between the internal and the conceptual levels, as well as between the internal and the conceptual levels, as well as between the conceptual and external levels, are also defined by the DBA. Ensuring that appropriate measures are in place to maintain the integrity of the database,

and that the database is not accessible to unauthorized users is another responsibility. The DBA is responsible for granting permission to the users of the database, and stores the profile of each user in the database. This profile describes the permissible activities of a user on that portion of the database accessible to the user via one or more user views. The user profile can be used by the database system to verify that a particular user can perform a given operation on the database.

The DBA is also responsible for defining procedures to recover the database from failures due to human, natural, or hardware causes with minimal loss of data. This recovery procedure should enable the organization to continue to function, and the intact portion of the database should continue to be available.

Let us summarize the functions of the DBA:

- **Schema definition:** The creation of the original database schema. This is accomplished by writing a set of definitions which is translated by the DDL compiler to a set of tables that is permanently stored in the data dictionary.
- **Storage Structure and access method definition:** The creation of appropriate storage structure and access method. This is accomplished by writing a set of definitions which are translated by the data storage and definition language compiler.
- **Schema and Physical organization modification:** Either the modification of the database schema or the description of the physical storage organization. These changes, although relatively rare, are accomplished by writing a set of definitions which is used by either the DDL compiler or the data storage, and definition language compiler to generate modification to the appropriate internal system tables (for example, the data dictionary).

- **Granting of authorization for data access:** The granting of different types of authorization for data access to the various users of the database.
- **Integrity constraint specification:** These constraints are kept in a special system structure that is consulted by the database manager whenever one of the valuable tools that the DBA uses to carry out data administration in data dictionary is required.

2.7.3 Data Dictionary

It is seen that when a program becomes somewhat large in size, keeping track of all the available names that are used, and the purpose for which they were used becomes more and more difficult. Of course it is possible for a programmer who has coined the available names to bear them in mind, but should the same author come back to his program after a significant time, or should another programmer have to modify the program, it would be found that it is extremely difficult to make a reliable account of the purpose the data files were used for.

The problem becomes even more difficult when the number of data types that an organization has in its database increases. It has also now perceived that the data of an organization is a valuable corporate resource, and therefore some kind of an inventory and catalogue of it must be maintained, so as to assist in both the utilization and management of the resource.

It is for this purpose that a data dictionary or dictionary/directory is emerging as a major tool. An inventory provides definitions of things. A directory tells you where to find them. A data dictionary/directory contains information (or data) about the data.

A comprehensive data dictionary would provide the definition of data items, how they fit into the data structure and how they relate to other entities in the database. With the comprehensive base of information the data

dictionary can serve several useful purposes connecting across the whole spectrum of planning, determining information requirement, designing and implementation operation and revision. There is now a greater emphasis on having an integrated system in which the data dictionary is part of the DBMS. In such a case the data dictionary would store the information concerning the external, conceptual and internal levels of the databases. It would combine the source of each data field value that is from where the authenticate value is obtained, the frequency of its use, and audit trail regarding the updates, including user identification with the time of each update.

The DBA uses the data dictionary in every phase of a database life cycle, starting from the embryonic data gathering phase to the design, implementation and maintenance phases. Documentation provided by a data dictionary is as valuable to end users and managers as it are essential to the programmers. Users can plan their applications with the database only if they know exactly what is stored in it. For example, the description of a data item in a data dictionary may include its origin and other text description in plain English, in addition to its data format. Thus users and managers will be able to see exactly what is available in the database. You could consider a data dictionary to be a road map which guides users to access information within a large database.

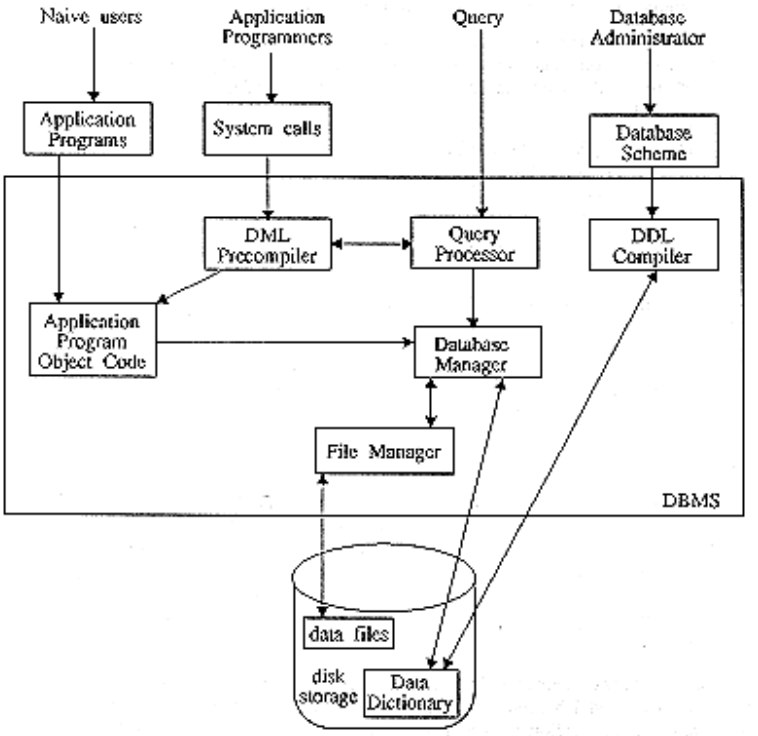

Figure 2.5: DBMS Structure

An ideal data dictionary should include everything a DBA wants to know about the database.

1. External, conceptual and internal database descriptions
2. Descriptions of entities (record types), attributes (fields), as well as cross-references, origin and meaning of data elements
3. Synonyms, authorization and security codes
4. Which external schemas are used by which programs, who the users are, and what their authorizations are.

A data dictionary is implemented as a database so that users can query its content by either interactive or batch processing. Whether or not the cost of acquiring a data dictionary system is justifiable depends on the size and complexity of the information system. The cost effectiveness of a data

dictionary increases as the complexity of an information system increases. A data dictionary can be a great asset not only to the DBA for database design, implementation and maintenance, but also to managers or end users in their project planning. Figure 2.6 shows these components and the connection among them.

Self Assessment Questions 2.7

1. _____ converts DML statement embedded in an application program to normal procedure calls in the host language.
2. The DDL compiler converts the data definition statements into a set of _____.
3. The _____ translates the various DML statements into low-level file system commands.
4. The _____ is used to interpret the online user's query and convert it into an efficient series of operations.
5. The _____ is also responsible for defining procedures to recover the database from failures.
6. The DBA grants different types of _____for data access to the various users of the database.
7. The DBA uses the _____ in every phase of a database life cycle.
8. A data dictionary is implemented as a database so that users can query its content by either interactive or _____ processing.

2.8 Distributed Processing

It is possible to make the distinction between two dimensions of distributed database systems: distributed data and distributed processing. In this Unit we provide an overview of the topic of distributed processing. Distribution can also be discussed in terms of distribution of functions or processing. Figure 2.6 illustrates a system characterized by distributed processing. Here, we also have four sites connected by a communications network.

However, in this configuration only site 1 stores any data. Sites 2, 3 and 4 act as clients to this database, perhaps running particular information system applications. It is for this reason we include client – server database systems within our discussion of distributed database systems. Although most current client – server database systems do not effectively distribute data, they do distribute functionality.

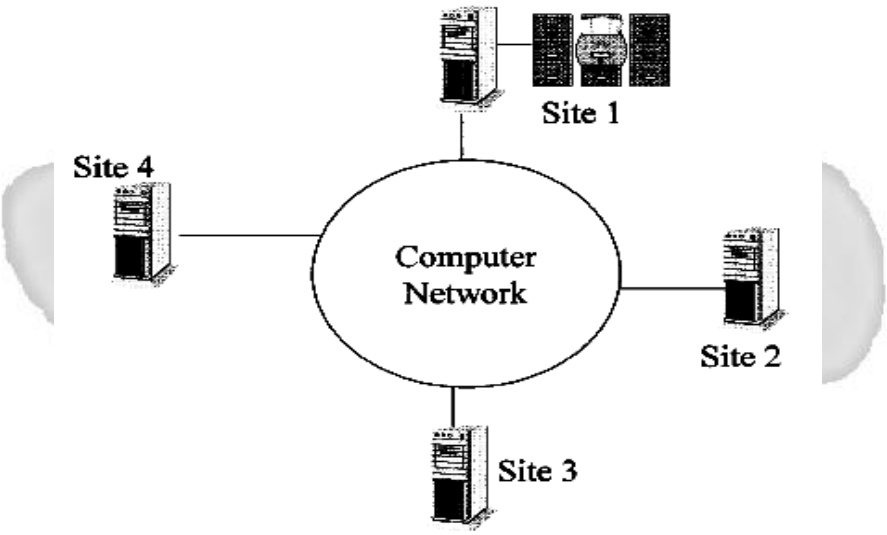

Figure 2.6: Example of Distributed Processing

2.8.1 Information and Communications Technology System (ICT)

In terms of software it is useful to consider an ICT system as being made up of a number of subsystems or horizontal layers (Figure 2.7). The layers define the major component parts of processing in an ICT system:

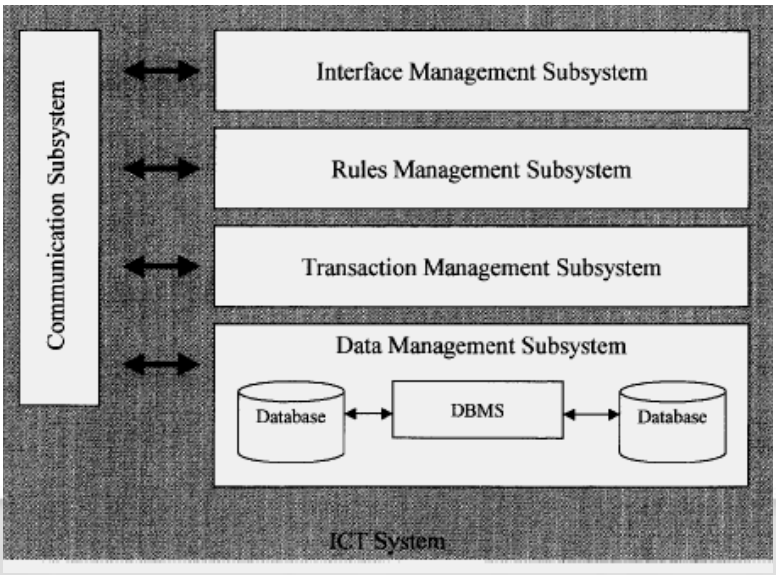

Figure 2.7: Components of ICT

- *Interface subsystem*: This subsystem is responsible for managing interaction with the end-user. It is generally referred to as the user interface
- *Rules subsystem*: This subsystem manages the application logic in terms of a defined model of business rules. In a database application these will comprise functions, which primarily enforce both inherent and additional constraints on data
- *Transaction subsystem*: This subsystem acts as a link between the data subsystem and the rules and interface subsystems. Querying, insertion and update activity are triggered at the interface, validated by the rules subsystem and packaged as units (transactions) that will initiate actions (responses or changes) in the data subsystem
- *Data subsystem*: This subsystem is responsible for managing the underlying data needed by an application.

2.8.2 Client – Server Architecture

Client – server is a software architecture in which two processes interact as superior and subordinate. The client process always initiates requests and the server process always responds to requests. Theoretically, client and server can reside on the same machine. Typically however they run on separate machines linked by some form of communications network, usually a local area network.

Many different types of servers exist, for example:

- Mail servers
- Print servers
- File servers
- Database (SQL) server

Most people tend to equate the term client–server with a network of PCs or workstations connected by a network to a remote machine running a database server.

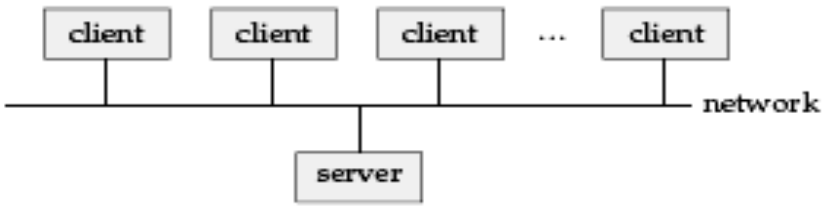

Figure 2.8: Client/Server Architecture

In practice, a client – server database system generally refers to a local area network of personal computers (PCs). At least one of these PCs is dedicated to serve the database needs of the others, which act in a client capacity. The database is held on the server. The user interface and application development tools are held on the client machines.

The server in this configuration is either set up as a file server or SQL server. In a file server situation, an SQL query expressed by a client will issue a request to the server for the appropriate files needed by the query. The client will perform the query and extract the relevant data. In an SQL server situation, the SQL statement will travel down the communication line from client to server.

The server then executes the query and sends back only the extracted data. Clearly, because of the reduced communication traffic, most DBMS now offer SQL server facilities.

Self Assessment Questions 2.8

1. In ICT, the _____ subsystem is responsible for managing interaction with the end-user.
2. Transaction subsystem acts as the link between the _____and the rules and interface subsystems.
3. The _____ subsystem manages the application logic in terms of a defined model of business rules.
4. Client–server is a software architecture in which two processes interact as superior and _____.
5. In practice, a client–server database system generally refers to a _____ of personal computers (PCs).

2.9 Summary

This Unit gives the descriptions and different components of MySQL, SQL Server 2000 and Oracle 9i architecture. The DBMS structure is defined by three views of data. A DBMS is a major software system consisting of a number of elements. It provides users DDL for defining the external and conceptual view of the data and DML for manipulating the data stored in the database. The database manager is the component of DBMS that provide the interface between the user and the file system. The database

administration defines and maintains the three levels of the database as well as the mapping between levels to insulate die higher levels from changes that take place in the lower levels. The DBA is responsible for implementing measures for ensuring the security, integrity and recovery of the database.

2.10 Terminal Questions

1. Explain the three level architecture of DBMS.
2. List and explain the MySQL architectural primary subsystems.
3. What are the applications of SQL Server 2000 architecture?
4. Explain the Oracle architecture with a neat diagram.
5. What are the two main types of facilities that are supported by the DBMS? Discuss their applications.
6. Discuss the responsibilities of the database manager and database administrator.
7. Describe the features of distributed processing.
8. Give the advantages of the client/server architecture.

2.11 Answers to Self Assessment Questions

Answers to Self Assessment Questions 2.2

1. Database
2. Subschema
3. External
4. Internal
5. One
6. Secondary

Answers to Self Assessment Questions 2.3

1. Five
2. Syntax Parser

3. Storage Manager
4. Query Engine
5. Transaction Manager

Answers to Self Assessment Questions 2.4

1. Online Transaction Processing (OLTP)
2. Independent Software Vendors (ISVs)

Answers to Self Assessment Questions 2.5

1. Three
2. Oracle server
3. Oracle Instance

Answers to Self Assessment Questions 2.6

1. data definition language (DDL)
2. data model
3. Procedural

Answers to Self Assessment Questions 2.7

1. DML Precompiler
2. Tables
3. database manager
4. query processor
5. DBA
6. Authorization
7. data dictionary
8. batch processing

Answers to Self Assessment Questions 2.8

1. Interface subsystem
2. data subsystem
3. rules
4. subordinate
5. local area network