

Database management System

Module 1

Data

Data means raw facts that can be recorded and have fixed meaning. We need to store the data somewhere. Thus database is used

Database

A database is a collection of related data. It has the following Properties

- Database represent some aspects of real world and is called miniworld or universe of disclosure
- Data base is a collection of data with fixed meaning
- Database is designed .built and populated with data for a specific purpose. It is used by a group of users

DBMS

It is a collection of programs that enables the users to create and maintain a database

Definition of DBMS:

DBMS is a general purpose software system that facilitated the process of defining, constructing, manipulating and sharing databases among various users and applications

- **Defining a database:** it involves specifying the data types, constraints and structures of the data to be stored in the database
Database definition is stored in database catalog and the definition is called **Meta data**
- **Constructing a database:** It is the process of storing data on some storage medium that is controlled by the DBMS
- **Manipulating a database:** It include querying the database to retrieve specific data, updating database, and removing any values from database

- **Sharing a database:** It allows multiple users and programs to access the database simultaneously

Characteristics of database approach

1. Self describing nature of database system
2. Insulation between programs, data and data independence
3. Support of multiple views of data
4. Sharing of data & multiuser transaction processing

1. Self describing nature of database system

- The database system contains not only the data itself but also the complete description of the database structure and constraints. This information is stored in DBMS catalog.
- Catalog contains information such as data types of each data item , structure and storage format of each data item and various constraints on data .These information's are called meta data. It describe the structure of the database

2. Insulation between programs, data and data independence

- In DBMS, the structure of the data file is stored in DBMS catalog separately from the access programs. So that the change in database structure do not require change in the access programs. This property is called program-data independence
- An operation in a database contain two parts
 - Interface(Signature)
 - Implementation(method)
- The interface of an operation includes the operation name and data type of the arguments. The description of the operation is specified in the implementation. We can change the implementation without affecting the interface.

The application programs can invoke these operations through their name and arguments regardless of how the operations are implemented. This property is called program-operation independence

- The characteristics which allows program-data independence and program-operation independence is called data abstraction

3. **Support of multiple views of data**

A database typically has many users. Each of whom may require different view of the database. A view may be subset of the database or it may contain virtual data which is derived from the database.

4. **Sharing of data & multiuser transaction processing**

A multiuser DBMS allows multiple users to access the database at the same time. The DBMS must include concurrency control software to ensure that several users trying to update the same data, do so in a controlled manner. So that the result of the update is correct

Advantages of DBMS

1. Controlling redundancy
2. Restricting unauthorized access
3. Providing persistent storage for program objects
4. Providing storage structure and search techniques for efficient query processing
5. Providing backup and recovery
6. Providing multiple user interfaces
7. Representing complex relationship among data
8. Enforcing integrity constraints
9. Permitting inferencing and action using rules

1. Controlling redundancy

Redundancy means storing the data multiple times in different location. This leads to several problems

- a. Duplication of effort
- b. Storage space is wasted
- c. Data may become inconsistent

Dbms could not allow the storage of redundant data

2. Restricting unauthorized access

When multiple users share a large database, some users will not be authorized to access all the information in the database. Some users only permitted to retrieve the stored data and others are allowed to retrieve and update the database

Hence this type of access operations must be controlled. Sign in methods are used by the dbms for this purpose

3. Providing persistent storage for program objects

DBMS is used to provide Providing persistent storage for program objects and data structures. In traditional system, there exists a problem that the data structure provided by the DBMS is incompatible with the data structure of the programming language. This problem could not exist in DBMS

4. Providing storage structure and search techniques for efficient query processing

Data base has the capabilities for efficiently executing queries and updates. The database is typically stored on the disk, so that the DBMS must provide specialized data structures and search techniques to speed up the disk search

5. Providing backup and recovery

DBMS must provide facilities for recovering from hardware and software failures. The backup and recovery subsystem of DBMS handle this function. The recovery system is responsible for making sure that the database is restored to the safe state

6. Providing multiple user interfaces

Because many types of users with varying levels of technical knowledge use the database, a DBMS should provide a variety of user interfaces

7. Representing complex relationship among data

DBMS has efficient facilities to represent complex relationship among the data. It also the facilities to define new relationships as they arise and to retrieve and update related data easily and efficiently

8. Enforcing integrity constraints

These constraints are derived from the meaning or semantics of the data. It is the responsibility of the database designers to identify integrity constraints during the database design

9. Permitting inferencing and action using rules

Some database system provides capabilities for defining deduction rules for inferencing new information from the stored database facts. Such systems are called deductive database systems.

Database users

There are 3 types of users

1. Data base administrators
2. Database designers
3. End users

1. Database administrators

In a database environment, the primary resource is the database itself and the secondary resource is the DBMS and related software's. Administrating these resources is the responsibility of the database administrator (DBA).

The DBA is responsible for authorizing access to the database, co-ordinating and monitoring its use, and acquiring software and hardware resources etc

2. Database designers

Database designers are responsible for identifying the data to be stored in the database. They are also responsible for choosing appropriate structures to represent and store this data.

These tasks are mostly undertaken before the database is implemented and populated with data.

Database designers interact with the users and develop the requirements to create and design the database

3. End users

End users are the people, whose jobs require access to the database for querying, updating and generating reports. The database primarily exists for their use.

Several types of end users

- ❖ Casual end users: They are occasionally access the database, but they may need different information each time
- ❖ Native or parametric end users: They make up a sizable portion of the database. Their job is to constantly querying and updating the database. Their transactions are called canned transactions.
- ❖ Sophisticated end users: They include engineers, scientists, and business analysts. They are using the DBMs in order to implement their own applications to meet their complex requirements
- ❖ Standalone end users: this category maintain personal databases by using ready-made program packages

Data models

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

Data model is used to achieve data abstraction. Data abstraction generally refers to the suppression of details of data organization and storage and highlighting only the essential features

By structure of a database we mean the data type, relationships and constraints that apply to the data

Categories of data model

We can categorize the data model according to the types of concepts they used to describe the data structure

1. High level or conceptual data model
2. Low level or physical data model
3. Representational or Implementation data model

High level or conceptual data model

- It provides concepts that are close to the way many users perceive data.
- Conceptual data models use concepts such as entities, attributes and relationships
 - An entity represents a real world object or concepts such as student, employee etc
 - An attribute represents some property that describes the entity such as student-id, student name, age etc
 - Relationship represents the association among two or more entities.
Example : study relationship between student and a department

Low level or physical data model

- It provides the concepts that describe the details of how data is stored on the computer storage media such as magnetic disks.
- The concepts provide by this model are generally meant for computer specialists not for end users
- It also specify the access path for the data
 - An access path is a structure that makes the search for particular database more efficient. Index file is an example of access path

Representational or Implementation data model

- It exists between conceptual and physical data models. It provide the concepts that may be easily understood by the end users
- This data model hide the details of data storage on disk
- Representational data model represent data by using record structures and hence it is called record-based data models
- Examples of representational data model include network and hierarchical models and they are called legacy models \

Schemas and Instances

In any datamodel, it's important to distinguish between the description of the database and the database itself. The description of the database is called database schema which is specified during database design and is not expected to change frequently

A displayed schema is called schema diagram. The diagram displays the structure of each record type but not the actual instances of records

We call each object in the schema as a schema construct

The data in the database at a particular moment in time is called a database state or snapshot. It is also called the current set of occurrences or instances in the database

The schema is sometime called the intension and a database state is called an extension of the database

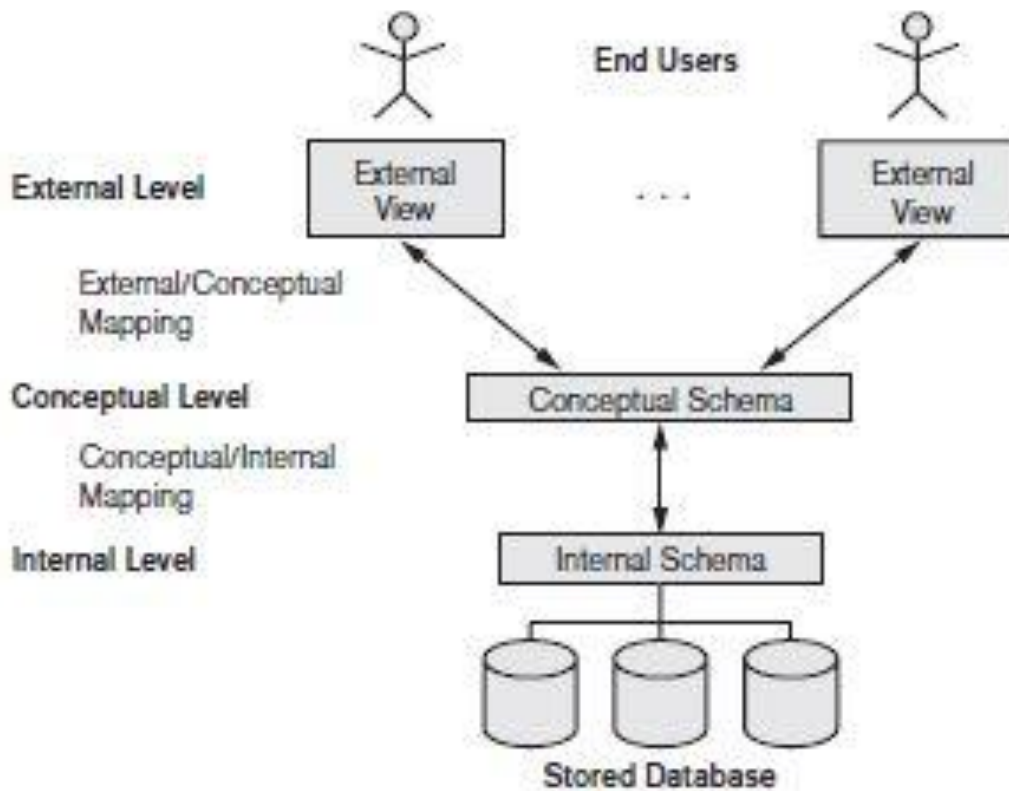
Schema Evolution

It is the ability of the database systems to respond to the changes in the real world

Valid state

A database state is said to be a valid state if it satisfies the structure and constraints which are specified in the schema

Three - Schema Architecture



The goal of the three-schema architecture is to separate the user application from the physical database

Schemas can be defined at the following three levels

1. Internal Level

The internal level has an internal schema, which describes the physical storage structure of the database. It uses the physical data model and describes the complete details data storage and access path for the database

2. Conceptual Level

The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users.

It hides the details of physical data storage structures and concentrates on describing entities, data types, relationships and user operations. Representational data model is used to describe the conceptual schema

3. External or View Level

It includes a number of external schemas. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group

External schema is implemented using representational data model

Data Independence

- The three schema architecture can be used to explain the concept of data independence.
- It is defined as the capacity to change the schema at one level of database system without changing the schema at the next higher level

There are two types of data independence

- ❖ Logical data independence
- ❖ Physical data independence

Logical Data Independence

It is the capacity to change the conceptual schema without changing the external schema or application programs

We may change the conceptual schema to expand the database [by adding data item], to change constraints or to reduce the database [by removing the data items]

Physical Data Independence

It is the capacity to change the internal schema without changing the conceptual schema and hence the external schema

We change the internal schema, if we want to reorganize some physical files

Database Languages

It include

- ❑ **Data Definition Languages(DDL)**

- ❑ **Storage Definition Languages(SDL)**

- ❑ **View Definition Languages(VDL)**

- ❑ **Data Manipulation Languages(DML)**

- DDL is used by the DBA and database designers to define both conceptual and internal schemas. The DBMS will have a DDL Compiler whose function is to process DDL statements in order to identify description of the schema and to store the schema description in the DBMS catalog
- In some DBMS, there is a clear separation is maintained between the conceptual and internal levels. In such DBMS, DDL is used to specify the conceptual schema only. So that, in order to specify the internal schema, storage definition language (SDL) is used

- The view definition language (VDL) is used to specify user views and their mappings to the conceptual schema. In relational DBMS, SQL is used in the role of VDL to define the user views
- Once the database schemas are compiled and populated with data, users can specify some manipulation on the database. The typical manipulations include insertion, deletion, retrieval and modification of the data. For this purpose database manipulation language is used

Two types of DML

- 1. High level or Non-procedural DML**
- 2. Low level or Procedural DML**

High level or non-procedural DML can be used to specify complex database operations concisely. They are entered interactively from a display monitor.

High level DML such as SQL can specify and retrieve many records in a single DML statement. Therefore, they are called set-at-a-time or set-oriented DML's

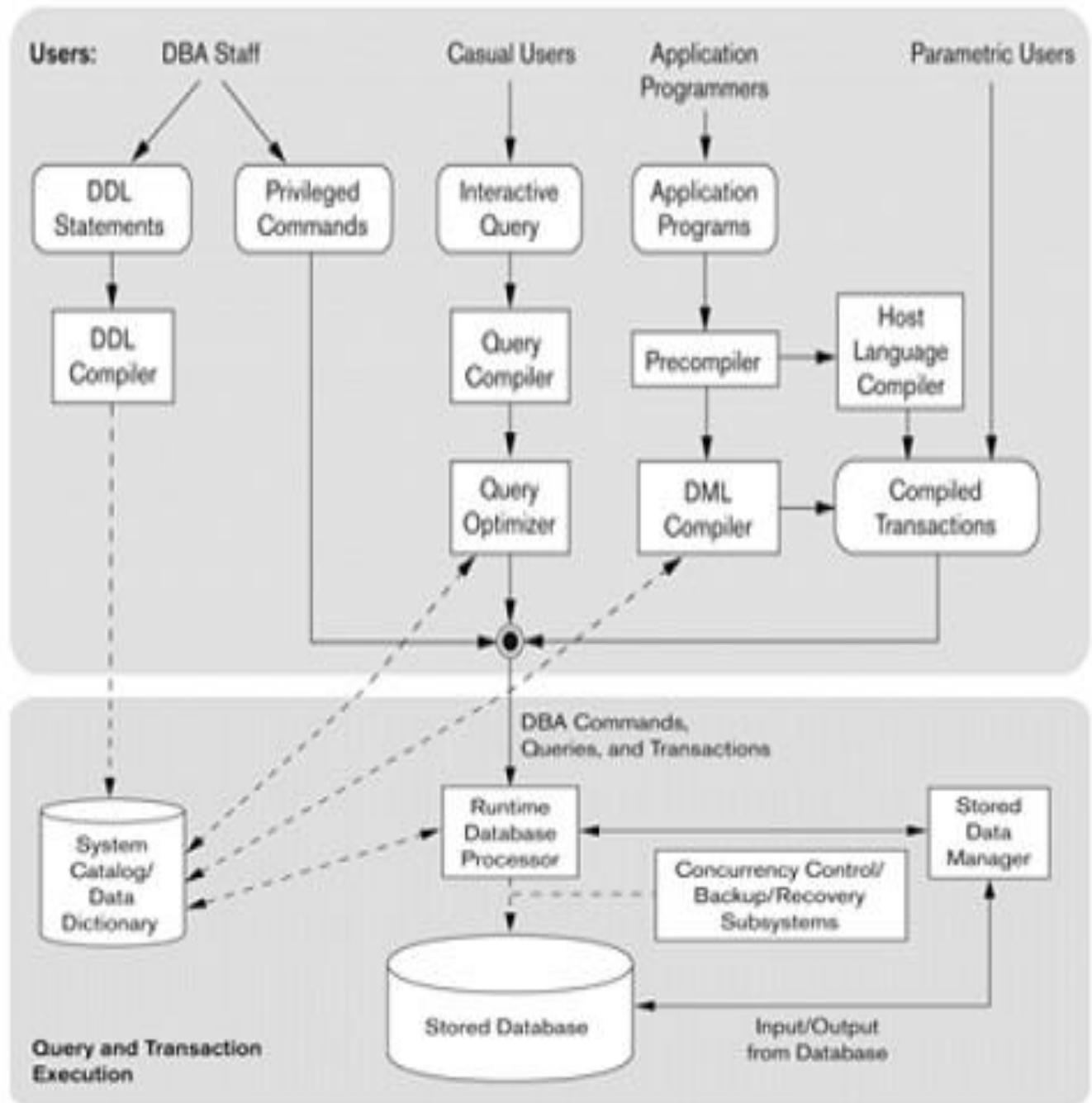
Low level DML must be embedded in a programming language. It retrieves individual record from the database and process each separately. So that low level DML's are also called record-at-a-time DML's

The Database System Environment

DBMS Component Modules

Following figure illustrate the DBMS component modules .The figure is divided into two parts

- Top part refers to the various users of database environment and their interfaces
- Lower part shows the storage of data and processing of transactions



- The DBA staff works on defining the database and making changes to its definition using DDL. The DDL compiler process schema definitions specified in the DDL and stores it in DBMS catalog

- Casual users occasionally need information from the database and they are interacting with the database using interactive query interface. It is a menu based or form based interaction.

The queries specified by the casual users are validated for correctness of the query syntax, the name of the files and data elements and so on by a query compiler. It compiles them into internal form. It is then used by the query optimizer for possible re-arrangement and re-ordering of operations, elimination of redundancies etc

- Application programmers write programs in host languages like C, C++, Java etc and they are submitted to a precompiler. The precompiler extract DML commands from an application program written in host lanagange I.e. [it separates back end from front end]. Then DML statements are passed on the DML compiler and host language programs are passed on to the host language compiler. The result of these two compilations together forms the canned transactions[compiled transactions]. Canned transactions are executed repeatedly by parametric users

- The run time database processor executes
 - Privileged commands
 - Executable query from query optimizer
 - Canned transaction

From the stored database. It is also work with stored data manager which uses basic operating system services for carrying out read/write operations from the database. It also has concurrency control subsystem, backup and recovery subsystem to co-ordinate the functions of DBMS