1.0 Introduction

1.0 Database Management System

Data are the raw facts that can be found after some experiment, observation or experience. Data itself do not provide any meaning but after processing it becomes information. The collection of related data organized in some specific manner is known as database. The database, its processing methods and the set of rules and conditions to be followed; collectively known as database management system (DBMS). Here, related data refers logically consistent facts of the real world. Random collection of data can not consider database. The primary goal of DBMS is to store and manage data both conveniently and efficiently. Database systems are generally designed to manage large volume of information. Management of data involves defining structure for storage of information and providing mechanisms for manipulation of information.

DBMS can also define as a general purpose software system that enables user to create, maintain and manipulate database. It provides fast and convenient access to information from data stored in database. DBMS interfaces with application programs so data contained in database can be accessed by multiple applications and users. Some popular DBMS softwares are: Oracle, SQL – Server, IBM-DB2, MySQL, MS Access, Sybase etc.
Some application areas of database system are:

- Banking: customer and their account info
- Airlines: reservations and schedules info
- Universities: student info, grades etc.
- Credit card transactions: for purchases on credit cards and generation of statements.
- Telecommunications: record of calls made
- Finance: for storing information about holding, sales and purchases etc.
- Sales: for customer, product and purchase information.
- Manufacturing: for management of supply chain.
- Human resources: for information about employee

1.2 Purpose of Database System

Traditionally, file processing system was used to manage information. It stores data in various files of different application programs to extract or insert data to appropriate file.
File processing system has several drawbacks due to which database management system is required. Database management system removes problems found in file processing system. Some major problems of file processing systems are:

1. Data redundancy and inconsistency

In file processing system, different programmer creates files and writes application programs to access it. After a long period of time files may exist with different formats and application programs may written in many different programming languages. Moreover, same information may be duplicated in several files. We have to pay for higher storage and access cost for such redundancy. It may leads database in inconsistent state because update made in one file may reflected in one file but it may not reflected in another files where same information exist in another files.

2. Difficulty in accessing data

In file processing system, we can not easily access required data stored in particular file. For each new task we have to write a new application program. File processing system can not allow data to be retrieve in convenient and efficient manner.

3. Data isolation

Since data are scatter in different files and data may stored in different format, so it is difficult to write program to retrieve appropriate data.

4. Integrity problem

In database, we required to enforce certain type consistency constraints to ensure the database correctness or to enforce certain business rules. It is in fact called integrity constraints (e.g. account balance > 0), integrity of database need not to be violated. In file processing system, integrity constraint becomes the part of application program. Programmer need to write appropriate code to enforce it. When new constraints are required to add or change existing one, it is difficult to change program to enforce it.

5. Atomicity problem

Failures may lead database in an inconsistent state with partial updates. For example, failure occurs while transferring fund from account A to B. There would be the case that certain amount from account A is retrieved and it is updated but failure occurs just before it is deposited to account B, such case may lead database in inconsistent state.

6. Concurrent access problem

Concurrent accessed increase the overall performance of system providing fast response time but uncontrolled concurrent accesses can lead inconsistencies in system. File processing system allow concurrent access but it is unable to coordinate different application programs so database may lead in inconsistent state. E.g. two people reading a balance and updating it at the same time

7. Security problems

Since file processing system consist large no. of application programs and it is added in ad hoc manner. So it is difficult to enforce security to each application to allow accessing only part of data/database for individual database users.

1.3 Data Abstraction

Data abstraction in database system is a mechanism to hide complexity of database. It allow database system to provides abstract view to database user. It hides how data are actually stored and maintain in database. Data abstraction simplifies users' interactions with the system.
Three are three level of abstraction

Physical level

It is a lowest level of abstraction. It describes how data are actually stored in database. It describes complex low level data structures in detail.

Logical Level

This is a next highest level of abstraction. It describes what data are stored in database and what relationship exists among them. It describes entire database relatively in a simple structure. The user in logical level needs not to aware the complexity of physical level structure.

View Level

It is the highest level of abstraction. It describes only part of the entire database. It simplifies interaction with the system. It allows database system to provide many views for the same database. That is it allows each user/application to get different perspective of the database.

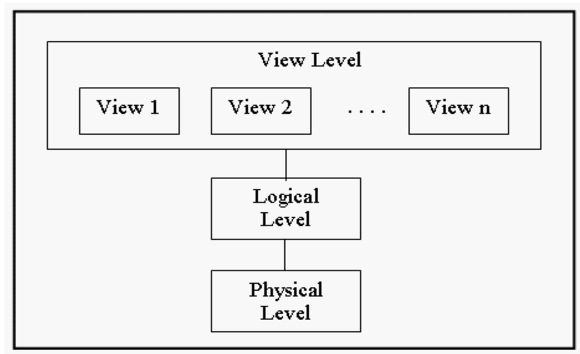

Figure1.3 Three level of abstraction

Example:

view level

- CS Majors
- Math Majors

logical level: entire database schema

- Courses (CourseNo,CourseName,Credits,Dept)
- Student (StudentID,Lname,Fname,Level,Major)
- Grade (StudentID,CourseNo,mark)

physical level:

- how these tables are stored, how many bytes it required etc.

1.4 Data Models

Data models describe the underlying structure of database. It is a conceptual tool for describing data, relationship among data, data semantics and consistency constraints. There are several data models which can be group into three categories.

(a) Object-based Logical Models.
(b) Record-based Logical Models.
(c) Physical Data Models.

1.4.1 Object-based Logical Models

Object based logical model describe data at the logical and view levels. It has flexible structuring capabilities. It allows to specify data constraints explicitly. Under object-based logical model there are sever data models

- Entity-relationship model
- Object-oriented model

1.4.1.1 Entity Relationship Model

E-R model describes the design of database in terms of entities and relationship among them. An entity is a “thing” or “object” in real world that are distinguishable from other objects. An entity is describes by a set of attributes.

For example

- Attributes account_number and balance may describe entity “account”.
- Attributes customer_id, customer_name, customer_city may describe entity “customer”.

A relationship is an association among several entities. For example, a depositor relationship associates a customer with each account he or she has.

The set of all entities of same type called entity set and similarly set of all relationship of the same type called relationship set.

E-R model graphically express overall logical structure of a database by an E-R diagram. Components of E-R diagram are as follows

rectangles: represent entity sets
ellipses: represent attributes
diamonds: represent relationships among entity sets
lines: link attributes to entity sets and entity sets to relationships

Example:

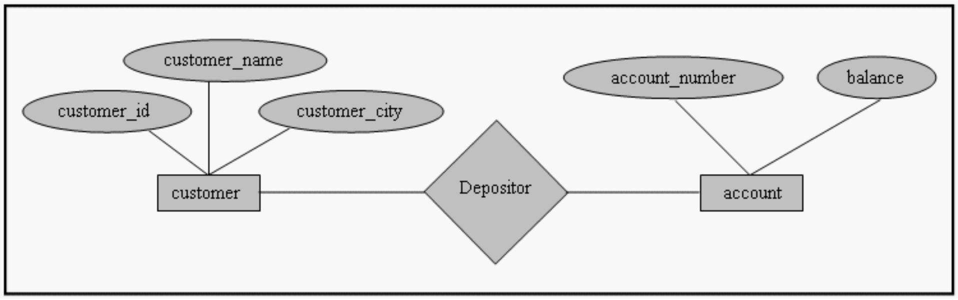

Figure 1.4.1.1.1 Sample E-R Diagram

Beside entities and relationship among them, E-R model has a capability to enforce constraints, mapping cardinalities which tell no. of entities to which another entity can be associated via relationship set. If each account must belong to only one customer, E-R model can express it. We discuss mapping cardinalities in detail in next chapter.

1.4.1.2 Object oriented model

Object oriented data model is extension to E-R model with the notion of encapsulation, methods (functions) and object identity. It is based on collection of objects, like the E-R model. An object contains values stored in instance variables within the object. These values are themselves objects. That is, objects can contain objects to an arbitrarily deep level of nesting. An object also contains bodies of code that operate on the object. These bodies of code are

called methods. Objects that contain the same types of values and the same methods are grouped into classes.
The only way in which one object can access the data of another object is by invoking the method of that other object. This is called sending a message to the object. Internal parts of the object, the instance variables and method code, are not visible externally.

Example:
Consider an object representing a bank account.

- The object may contain instance variables account_number and balance.
- The object may contain a method pay-interest which adds interest to the balance.

Unlike entities in the E-R model, each object has its own unique identity. It is independent to the values it contains. Two objects containing the same values are distinct. Distinction is maintained in physical level by assigning distinct object identifier.

1.4.2 Record-based Logical Models

As object based base logical model, record-based logical model also describes data at logical and view level. But it describes logical structure of database in more detail for implementation point of view. It describes database structure in terms of fixed-format records of different types. Each table contains records of a particular type. And each record type defines fixed number of fields or attributes. Each field is usually of a fixed length.

There are several languages which are used to express database queries and updates.

The three most widely-accepted models under record-based logical models are:

- Relational model
- Network model
- Hierarchical

1.4.2.1 The Relational Model

Relational model describes database design by a collection of tables (relations). It represents both data and their relationships among those data. Each table consist number of columns (attributes) with unique names. It is a most widely used data model. Relational model is lower level abstraction than E-R model. Database model are often carried out in E-R model and then translated into relational mode.

Example:
Previous describe E-R model can be express in relational model as follows

customer_id	customer_name	customer_city
C01	X	A
C02	Y	B
C03	Z	A
C04	X	A

(a) Customer relation

account_number	balance
A1	200
A2	300

customer_id	account_number
C04	A4
C01	A1
C02	A2
C03	A3

A3	500
A4	500

(b) Account relation (c) Depositor relation

Figure 1.4.2.1.1 Sample Relational database

1.4.2.2 The Network Model

In network model, data are represented by the set of records and relationships among data are represented by links.

Example:
The above relational model can be express in network model as follows

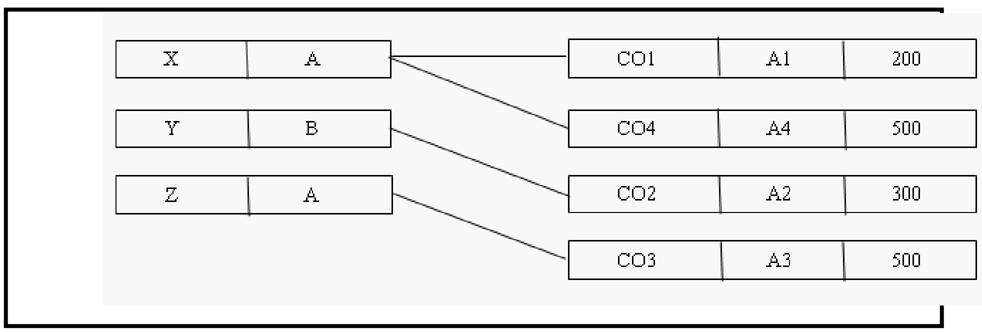

Figure 1.4.2.2.1 Sample Network Model

1.4.2.3 Hierarchical Model

Hierarchical model is also represents data by a set of records but records are organized in hierarchical or order structure and database is a collection of such disjoint trees. The nodes of the tree represent record types. Hierarchical tree consists one root record type along with zero more occurrences of its dependent subtree and each dependent subtree is again hierarchical. In hierarchical model, no dependent record can occur without its parent record. Furthermore, no dependent record may be connected to more than one parent record.

Example:

The above network model can express in hierarchical model as follows

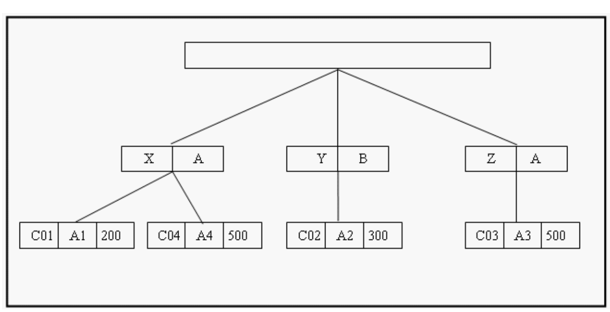

Figure 1.4.2.3.1 Sample Hierarchical Model

1.4.3 Physical Data Models

Physical data models are used to describe data at the lowest level. Unifying model and frame memory are follows under physical data model. Here in this section we are not cover physical data model.

1.5 Instance and Schemas

Database change over time as information is inserted, deleted and updated. The collection of information stored in database at a particular moment called an instance of the database. The overall design of the database is called database schema. Schemas are change infrequently, if at all.

According to the level of abstraction schema are divided into physical schema, logical schema and subschemas. The physical schema describes the database design at the physical level. Logical schema describes database design at the logical level. Database system may have several schemas at the view level, it is called sunschemas (can be query), it describes different views of database.

Logical schema is more important for the development of application programs. Programmer constructs applications by using logical schema. The physical schema is hidden under the logical schema and it can change without affecting application programs.

1.6 Data Independence

Data independence is an ability to modify a schema definition in one level without affecting scheme definition in higher level. There are two types of data independence.

Physical data independence
It is an ability to modify the physical scheme without causing application programs to be rewritten
Modification at this level usually required for performance improvement reason.

Logical data independence
It is an ability to modify the conceptual/logical scheme without causing application programs to be rewritten. Logical scheme needs to modify if we required to modify logical structure of database. Logical data independence is harder to achieve since application programs are usually dependent on logical structure of the data.

1.7 Database Languages

Database system provides two languages
(a) Data Definition Language and
(b) Data Manipulation Language

But in practice, data definition language and data manipulation language are not separate languages.

1.7.1 Data Definition Language (DDL)

Data definition language used to specify database scheme. For example, following DDL statement in SQL defines account relation.

```
create table account
(
 account_no   char(2),
 balance        integer
)
```

The execution of above DDL statement creates table account. Moreover, it updates special set of tables called data dictionary or data directory. Data dictionary contains meta data, that is data about data. For example table containing tables' information like table name, owner, created date, modified date etc refers data dictionary and contain information are example of meta data.

Data definition language also allows to define storage structure and access methods for database system, such special set of DDL statement called *data storage and definition* language.

1.7.2 Data Manipulation Language (DML)

Data manipulation language allow database user to access (query) and manipulate data. That is, DML is responsible for

- retrieval of information from the database
- insertion of new information into the database
- deletion of information in the database
- modification of information in the database

DML established communication between user and database.

There are two types of DML

(a) Procedural DML: user required to specify what data are needed and how they get those data.
(b) Nonprocedural (Declarative) DML: user only required to what data needed without specifying how to get those data.

Declarative DMLs are usually easier to learn and use than procedural DMLs. However, since a user does not have to specify how to get data, the database system has to figure out an efficient means of accessing data. The DML component of SQL is nonprocedural.

A query is statement requesting the retrieval of information. Special set of DML which only use to retrieve information from database called *query language*.

Example:

```
Select customer_name
      from customer
where customer.customer_id='c001'
```

This query retrieves those rows from table customer where the customer_id=c01.

1.8 Database Manager

The 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. Since database required lots of storage space so it must be stored on disks. Data need to moved between disk and main memory as needed.

Since the goal of database system is to simplify and facilitate access to data providing optimal performance as far as possible. So the database manager module is responsible for

- Interaction with the file manager: responsible to translate DML statements into low-level file system commands for storing, retrieving and updating data in the database.
- Integrity enforcement: responsible to check any updates in the database do not violate consistency constraints.(e.g. no bank account balance below $25).
- Security enforcement: responsible to ensure that users only have access to information they are permitted to see.
- Backup and recovery: Detecting failures due to power failure, disk crash, software errors, etc., and restoring the database to its state before the failure.
- Concurrency control: responsible to preserving data consistency when there are concurrent users.

1.9 Database Administrator

The database administrator is a person having central control over data and programs accessing that data. Database administrator has the following responsibility:

- Schema definition: responsible for the creation of original database schema. So DBA is responsible to write data definition statements in DDL.
- Storage structure and access method definition: DBA is responsible to write a set of definitions to define storage and access method using storage and access.
- Schema and physical organization modification: DBA is responsible for modification of schema and to reflect the changes in schema or to improve the performance physical organization may need to be change.
- Granting authorization for data access: DBA is responsible to grant different types of authorization for data access to various users.
- Routine maintenance:
    - Periodically backing up the database ensuring enough free disk space available for normal operations and upgrading disk space as required.
    - Monitoring jobs running on the database and ensuring that performance is not degraded too much.

1.10 Database Users

There are four different types of database users, they are differentiated according to their interaction with the system. Moreover, there are different types of user interfaces for different types of users.

(a) Naïve Users:

Naïve users are unsophisticated users who interact with the system by invoking one of the application programs that are already written. For example, banks teller who needs to transfer fund from one account to another invoking a program called transfer. This program asks the teller for the amount of money to be transferred, and account to which the money is to be transferred.

The typical user interface for the native user is a form interface, where user can fill appropriate fields of the form. Native users may also simply read reports generated from the database.

(b) application programmers:

Application programmers are computer professional who write application programs. Application programmers may choose any programming tool to develop user interfaces. They can also used RAD tools that enable an application programmer to construct forms and reports without writing the program. There are also special type of programming languages that combine imperative control structures (e.g. for loops, while loops and if-then-else statements) with the statements of data manipulation language. These languages are sometimes called fourth generation languages. It often includes special features to facilitate the generation of forms and display data on the screen. Most major commercial database system includes a fourth generation language.

(c) sophisticated users:

Sophisticated user interact with system without writing programs but they requests by writing queries in database using DML query language. This query goes to query processor and it converted into instructions for the database manager module.

(d) Specialized users:

Specialized users are responsible to write special database application programs it could be computer-aided design systems, knowledge based and expert systems that store data with complex data types (e.g. graphics data, audio/video data).

1.11 Overall System Structure

The functional component of the database system is divided into storage manager and query processor component.

1.11.1 Storage Manager

Storage manager is a program module that provides interface between the low level data stored in the database and the application programs and queries submitted to the system. The storage manager is responsible for the interaction with the file manager. The storage manager various DML statements into low level file system command. And it is responsible for storing, retrieving, and updating data in the database.

Storage manager consist following components:

Authorization and integrity manager: responsible to ensure integrity constraint does not violate and checks the authority of users to access data.

Transaction Manager: responsible to ensure database remain inconsistent state even system failure occurs. It is also responsible to manage concurrent transactions so that they could not conflict, which also helps to ensure consistency of database.

File Manager: responsible to manage the allocation of space on disk storage and the data structures used to represent information stored on disk.

Buffer Manager: responsible for fetching data from disk storage into main memory, and decides what data to cache in main memory.
The storage manager implements several data structure for physical system implementation:

Data files: stores database itself,
Data dictionary: stores meta data about structure of database, in particular schema of database.
Indices: provides fast access to data items that holds particular values.

1.11.2 Query processor

The query processor is responsible to simplify and facilitate access data. It is responsible to translate updates and queries written in nonprocedural language at the logical level, into an efficient sequence of operations at the physical level.

The query processor component includes the following components:

DDL interpreter: responsible to interprets DDL statements and records the definitions in the data dictionary.

DML Compiler: responsible to translate DML statements in a query language into low level instructions that query evaluation engine understands. Query is generally translated into no. of alternative evaluation plans that produce the same result. It is also responsible for query optimization; it required to select the lowest cost evaluation plan among the alternatives

Query evaluation: responsible to execute low level instruction generated by DML compiler.

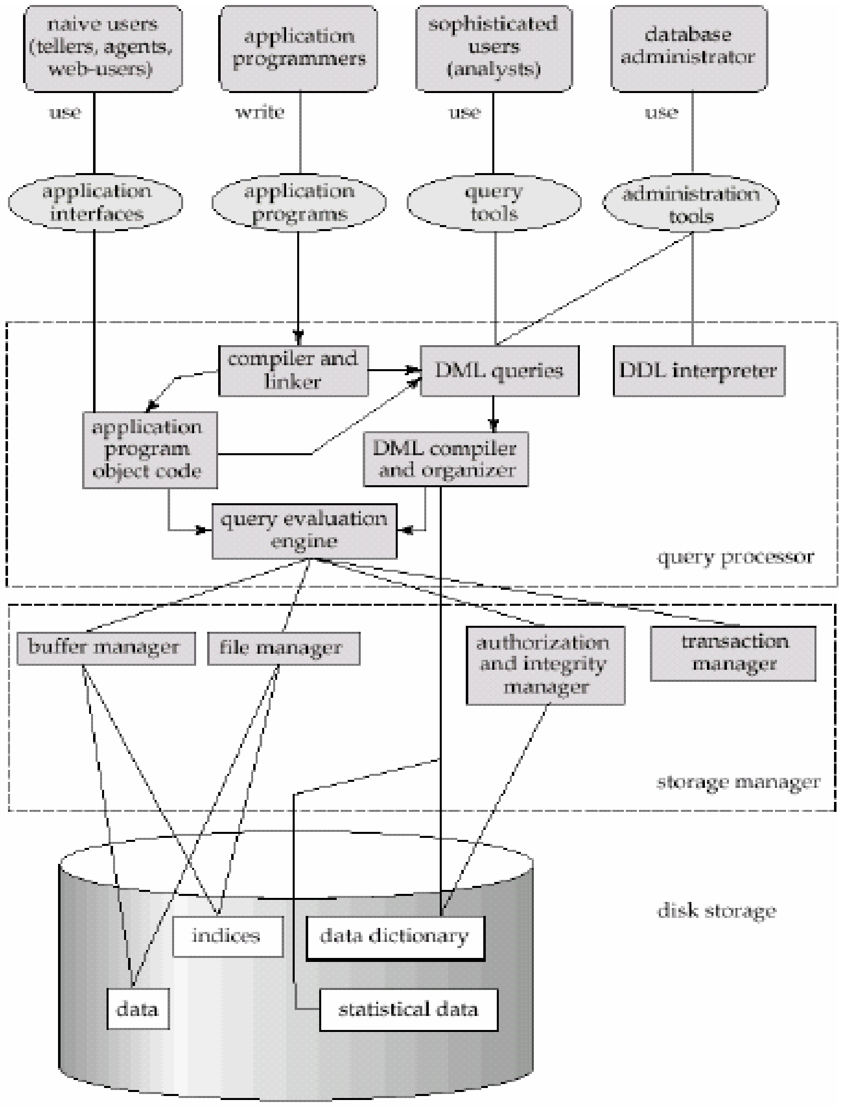

Figure 1.11.1 Overall database system structure

1.12 Advantages and disadvantages of DBMS

1.12.1 Advantages of DBMS

Data independence: DBMS provides abstract view of data. Application programs are independent from details of data representation and storage.

Efficient data access: DBMS provides verity of sophisticated techniques to store and retrieve data efficiently.

Data integrity and security: DBMS allow to enforce integrity constraints on data. For example before inserting salary information for an employee, DBMS can enforce integrity constraint to check salary is not exceeded department budget. DBMS can also enforce access controls, what data is visible to what class of users.

Data administration: DBMS provides centralized administration of data. It is appropriate when several no. of database user shares data. It improves the overall performance of database system.

Concurrent access and crash recovery: DBMS has a capability manage concurrent access. It schedules concurrent access to the data in such a manner that user fell data is being accessed by only one user at a time. Moreover, DBMS protects users from the effects of system failures.

Reduced application development time: since DBMS supports many important functions that are common to many applications accessing data stored in database. It provides high level interface to data and facilitates quick development of applications.

1.12.2 Disadvantage of DBMS

- Complex architecture of DBMS software
- DBMS software cost
- Since DBMS is optimized certain kind of workloads (e.g. answering complex queries or handling many concurrent requests) its performance may not appropriate for certain specialized applications.
- Abstract view of data presented by DBMS may not match for certain applications. For example, relational databases does not supports flexible analysis of text data
- If specialized performance or data manipulation requirements are central to an application, DBMS is not appropriate for such application. The added benefits of a DBMS (e.g. flexible querying, security, concurrent access and crash recovery) may not require for applications.