ST. XAVIER’S COLLEGE

**Maitighar, Kathmandu**

**(Affiliated to Tribhuvan University)**



**Database Management System**

Report on Introduction to DBMS

**Submitted By**

Alok Shrestha

B.Sc. CSIT

Year II/IV Semester

013BSCIT005

**Submitted To**

Er. Sanjay Kumar Yadav

Lecturer

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On**

July 24, 2015

**Introduction to Database Management Systems**

**Database Management System – An Introduction**

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

**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 cannot easily access required data stored in particular file. For each new task we have to write a new application program. File processing system cannot 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.

**Data Abstraction**

Data abstraction in database system is a mechanism to hide complexity of database. It allows database system to provide 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**

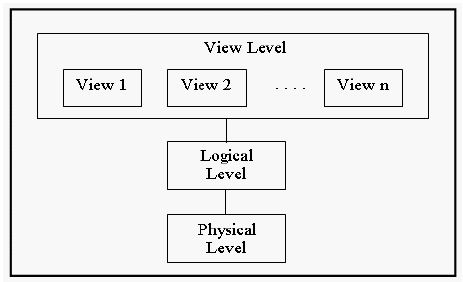
It is a lowest level of abstraction. It describes how data are actually stored in database. It describes complex low level data structures in detail.

**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.

**

*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.

**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.

**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.

**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.

**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 move 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.

**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:

o Periodically backing up the database ensuring enough free disk space available for normal operations and upgrading disk space as required.

o Monitoring jobs running on the database and ensuring that performance is not degraded too much.

**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).

**Overall System Structure**

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

**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.

**Query processor**

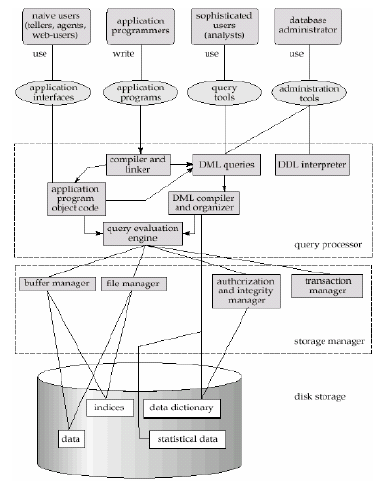
The query processor is responsible to simplify and facilitate access data. It is responsible to translate updates and queries written in nonprocedural language at the logical level, into an efficient sequence of operations at the physical level.

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.



*Overall database system structure*

**History of Database Systems**

**Early Manual System**

* Before-1950s
  + Data was stored as paper records.
  + Lot of man power involved.
  + Lot of time was wasted. e.g. when searching
  + Therefore inefficient.

**Revolution began**

* 1950s and early 1960s:
  + Data processing using magnetic tapes for storage
  + Tapes provide only sequential access
  + Punched cards for input
* Late 1960s and 1970s:
  + Hard disks allow direct access to data
  + Data stored in files
  + Known as File Processing System

**File based systems**

* Adequate for small applications
* Drawbacks
  + Separation and isolation of data
    - Each program maintains its own set of data.
    - Users of one program may be unaware of potentially useful data held by other programs.
  + Duplication of data
    - Same data is held by different locations.
    - Wasted space and potentially different values and/or different formats for the same item.
  + Data dependence
    - File structure is defined in the program code.
  + Incompatible file formats
    - Programs are written in different languages, and so cannot easily access each other’s files.
  + Fixed Queries/Proliferation of application programs
    - Programs are written to satisfy particular functions.
    - Any new requirement needs a new program.

**Database Approach**

* Arose because:
  + Definition of data was embedded in application programs, rather than being stored separately and independently.
  + No control over access and manipulation of data beyond that imposed by application programs.
* Result:
  + The database and Database Management System (DBMS).

**Database Management Systems (DBMS)**

1960’s Hierarchical Network

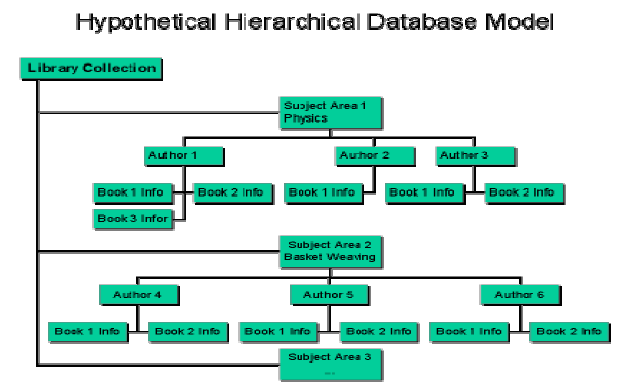
1970’s Relational

1990’s Object-oriented Object-relational

1995+ Java XML CMDB Mobile IMDB Embedded

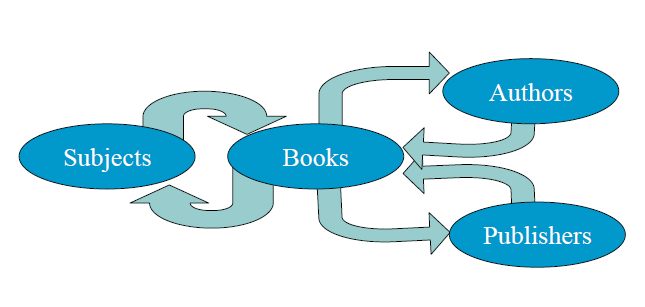
**Hierarchical Model**

* Well suited for data which are in some way related
* Hierarchically begin with a strictly defined tree of data nodes
* Each node can contain some identifying data, plus a set of subnodes of a specific child type



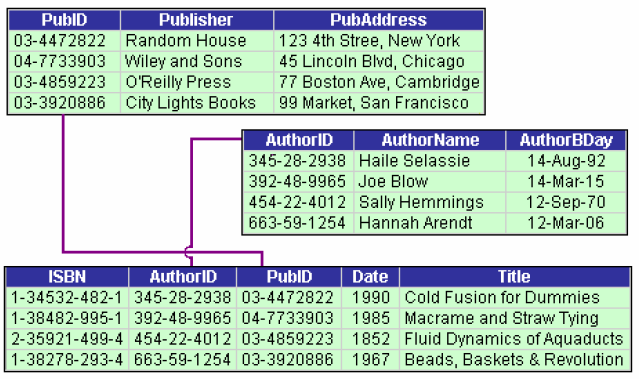
**Network Model**

* Supported more complex relations
* Physical file pointers were used to model the relations between files
* Relations had to be decide in advance
* Most suitable for large databases with well defined queries and welldefined applications.



**Relational Model (1970’s)**

* E.F. Codd introduced the relational model in 1970
* Provides a conceptually simple model for data as relations (typically considered “tables”) with all data visible.
* DB2 from IBM is the first DBMS product based on the relational



* Other DBMS based on the relational model were developed in the late 1980s
* Today, DB2, Oracle, and SQL Server are the most prominent commercial DBMS products based on the relational model Object Oriented Data Model (1990’s)
* Goal of OODBMS is to store object oriented programming objects in a database without having to transform them into relational format.
* Extend the entity-relationship data model by including encapsulation, methods and object identity

**Object-relational models**

* Extend the relational data model by including object orientation and constructs to deal with added data types.
* Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.
* Preserve relational foundations, in particular the declarative access to data, while extending modeling power.

**Modern Database Management Systems**

* DBMS are large complex pieces of software designed specifically for the efficient management of data.
* Examples:
  + Oracle (Oracle Corporation)
  + Ingres (Computer Associates)
  + SQL Server (Microsoft Corporation)
  + Access (Microsoft Corporation)
  + IMS, DB2 (IBM)
  + And many more…

**REFERENCE**

http://courses.dbnet.ntua.gr/fsr/5706/history\_dbms\_long-3.pdf