Database Management System (DBMS)

Avinash V. Gondal

email: avinash.gondal@gmail.com

Module 1

Introduction Database Concepts

1.Introduction Database Concepts

Contents :

- Introduction
- Characteristics of databases
- File system V/s Database system
- Data abstraction and Data Independence
- DBMS System architecture
- Database Administrator

Introduction to Data Processing

Data, Information and Knowledge

- Data is raw, unprocessed information and itself it may not mean more.

Or

- A data is known as the raw fact which is stored in computers memory and used by the users as the output.
- It simply exists and has no significance beyond its existence.
- Raw data refers to a collection of numbers, Characters, images etc. that are unprocessed.
- Ex :A persons age, a persons gender, the color of the car, Individually , it means not much.

Data, Information and Knowledge

- On the other hand, information is the processed data that has meaning.

- If a program takes the above data and processes it, now the result take some meaning.
- Ex: A 25 years old man likes to drive a black car.

- Information is like someone smiling. Data is like someone's poker face (no action or expression on face).

- Data that are processed to be useful; provides answers to “who”, “where”, and “when” questions.

Data, Information and Knowledge

- **Knowledge**: application of data and information.

Or

- Knowledge is the understanding of rules needed to interpret information

- Thus, data are the foundation of information, which is the bedrock of **knowledge**

Data Processing

- To achieve its aims the organisation will need to **process** data into information.
- Data needs to be turned into meaningful information and presented in its most useful format
- Good decisions require good information derived from raw facts
- Data is managed most efficiently when stored in a database
- Databases evolved from computer file systems

Drawbacks of File System

Drawbacks of File System

- In the early days, database applications were built directly on top of file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation
    - Multiple files and formats.
    - Same data is in different files at different places and different formats.

Drawbacks of File System

- Drawbacks of using file systems (cont.)
  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become "buried" in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones
  - Atomicity of updates
    - Failures may leave database in an inconsistent state with partial updates carried out
    - Example: Transfer of funds from one account to another should either complete or not happen at all

Drawbacks of File System

- Drawbacks of using file systems (cont.)
  - Concurrent access by multiple users
    - Concurrent accesses needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
      - Example: Two people reading a balance and updating it at the same time
  - Security problems
    - Hard to provide user access to some, but not all, data
- Database systems offer solutions to all the above problems

Concept of Database

Introducing the Database

- **What is database (DB) ?**

A Database is a collection of data, typically describing the activities of one or more related organizations.

e.g. university

- **What is DBMS ?**

A DBMS is a software designed to assist in maintaining and utilizing large collections of data.

or

A DBMS consists of a collection of interrelated data and a set of programs to access that data.

Database Applications

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

- Databases touch all aspects of our lives

Advantages of Database

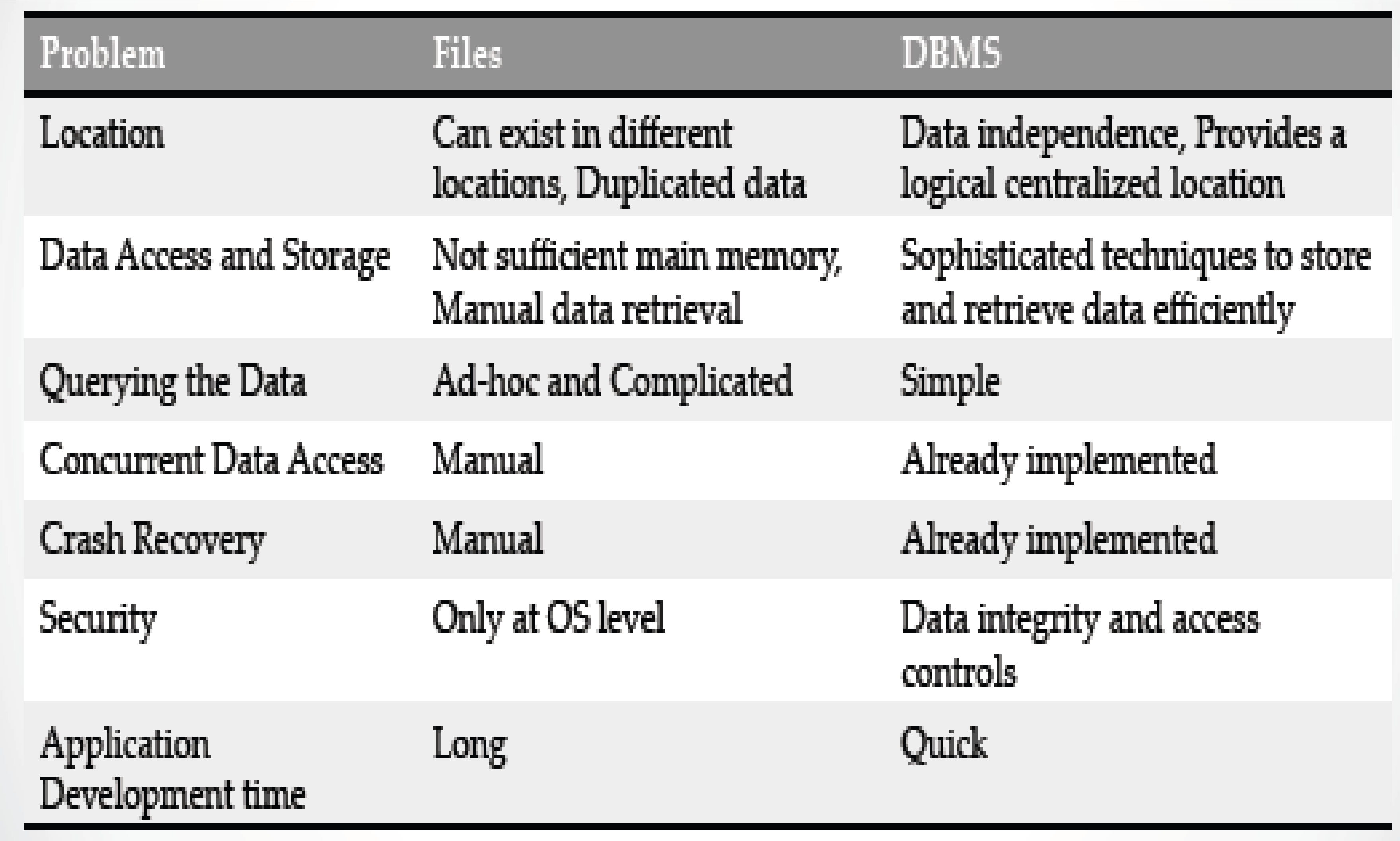
Advantages of Database

- Redundancy can be reduced
- Inconsistency can be avoided
- The data can be shared
- Standards can be enforced
- Security can be enforced
- Integrity can be maintained

(NOTE :-In Paper write the description of each point)

Comparison of Database System and File System

Files Vs. Database



Problem	Files	DBMS
Location	Can exist in different locations, Duplicated data	Data independence, Provides a logical centralized location
Data Access and Storage	Not sufficient main memory, Manual data retrieval	Sophisticated techniques to store and retrieve data efficiently
Querying the Data	Ad-hoc and Complicated	Simple
Concurrent Data Access	Manual	Already implemented
Crash Recovery	Manual	Already implemented
Security	Only at OS level	Data integrity and access controls
Application Development time	Long	Quick

Database Users and Administrators

Database Users and Administrators

- People who work with a database can be categorized as :
  - database users
  - database administrators

Database Users

- There are four different types of database-system users, differentiated by the way they expect to interact with the system.
  - Database Administrators
  - Database Designers
  - End Users
  - System Analysts

End Users

- There are different types of database-system users, differentiated by the way they expect to interact with the system.
  - Casual users
  - Naïve users
  - Application programmers
  - Sophisticated users
  - Specialized users
- **Casual Users :**
  - access database occasionally by sophisticated query language when needed.
  - Example : Bank Manager

End Users

- **Naive users :**
  - are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.
  - they make up a large section of the end-user population. Learn only a few facilities that they may use repeatedly
  - Example : bank clerk
- **System Analysts and Application programmers :**
  - are computer professionals who write application programs.
  - Application programmers can choose from many tools to develop user interfaces.
  - Rapid application development (RAD) tools are tools that enable an application programmer to construct forms and reports without writing a program.

End Users

- **Sophisticated users :**
  - interact with the system without writing programs.
  - Instead, they form their requests in a database query language.
  - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
- **Specialized users :**
  - are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework.
  - among these applications are computer-aided design systems, knowledge base and expert systems, systems that store data with complex data types (for example, graphics data and audio data)

Database Designers

- **Database Designers :**
  - Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

Advantages of Database Management System

Advantages of DBMS

Using a DBMS to manage data has many advantages

- Data Independence
- Efficient Data Access
- Data Integrity and Security
- Data Administration
- Concurrent Access and Crash Recovery
- Reduced Application Development Time

(NOTE :-In Paper write the description of each point)

Concerns When Using an Enterprise Database

Issues of DBMS

- A DBMS is a complex software.
- Confidentiality, Privacy and Security.
- Requires the large amount of primary memory and secondary storage.
- High cost
- The centralization of all data resources increases vulnerability of the system.

Data Abstraction,
3-Layered Architecture,
Data Independence

Data Abstraction

- A major purpose of a database system is to provide users with an *abstract* view of the data.
- That is, the system hides certain details of how the data are stored and maintained.
- Levels of Abstraction :
    - External or view Level
    - Conceptual Level
    - Internal or Physical Level

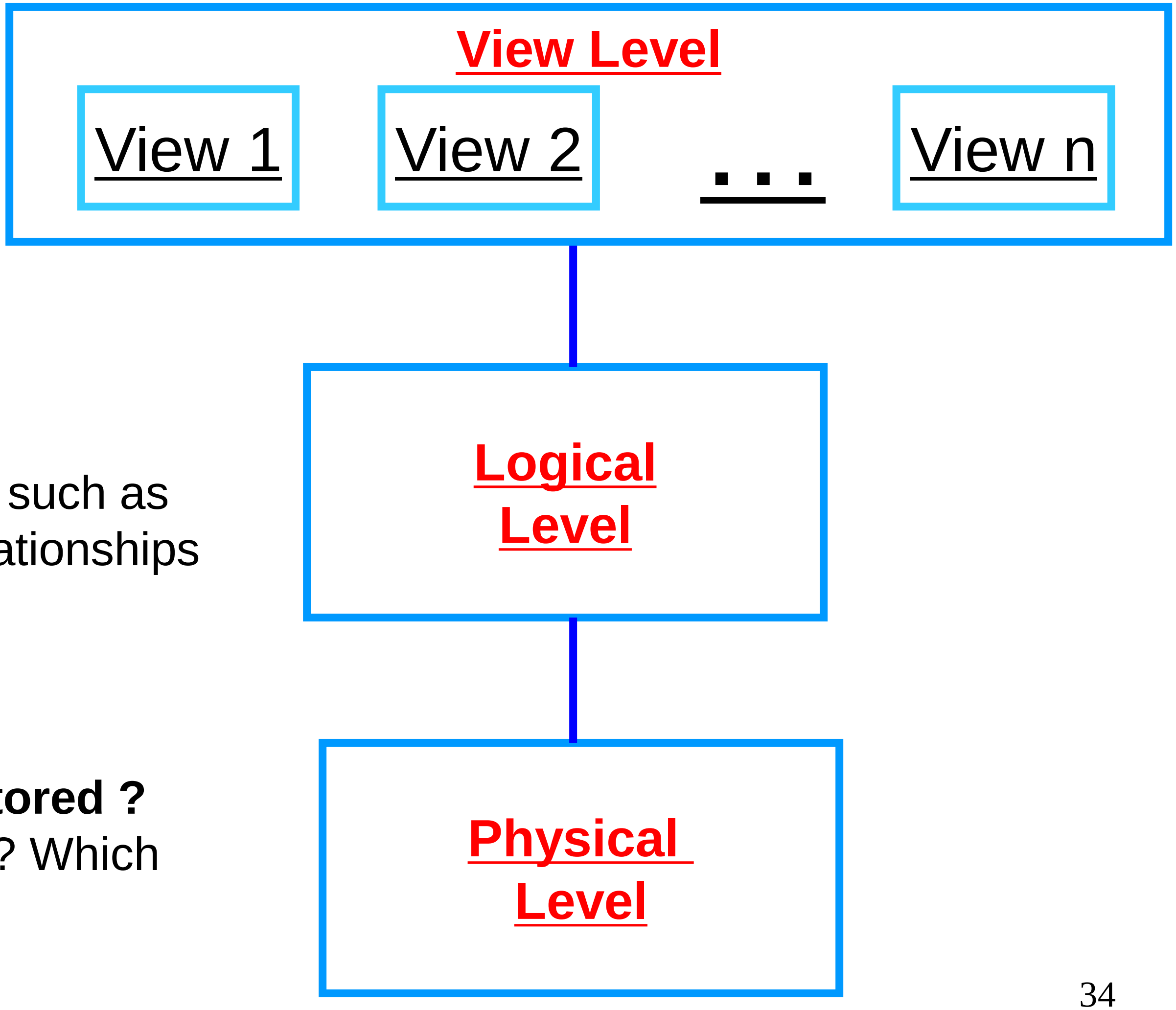
Levels of Abstraction

- Objectives :
  - insulation of application programs and data
  - support of multiple user views
  - use of schema to store the DB description (mete-data)
- Levels of Abstraction :
  - External or View Level
  - Conceptual Level
  - Internal or Physical Level

3- Layered Architecture of DBMS

- An architecture for a database system

What data users and application programs see ?



View Level

View 1 View 2 . . . View n

Logical Level

Physical Level

What data is stored ?
describe data properties such as data semantics, data relationships

How data is actually stored ?
e.g. are we using disks ? Which file system ?

3- Layered Architecture of DBMS

External Level

Sales Officer

View 1
Item_Name
Price

Inventory Controller

View 2
Item_Name
Stock

Conceptual Level

Conceptual	
Item_Number	**Character (6)**
Item_Name	**Character(30)**
Price	**Numeric(5,2)**
Stock	**Numeric(4)**

Physical Level

Physical	
Stored_Item	**Length=50**
Item #	**Type = Byte(6), offset = 0, Index = Ix**
Name	**Type = Byte(30), offset = 6**
Price	**Type = Byte(8), offset = 36**
Stock	**Type = Byte(4), offset = 44**

Levels of Abstraction

- External Schema or View Level :
  - Describes a subset of the database that a particular user group is interested in, according to the format the format user wants, and hides the rest
  - May contain virtual data that is derived from the files, but is not explicitly stored

Levels of Abstraction

- Conceptual schema
  - Hides the details of physical storage structures and concentrates on describing entities, data types, relationships, operations, and constraints.
- Internal schema
  - Describes the physical storage structure of the DB
  - Uses a low-level (physical) data model to describe the complete details of data storage and access paths

Instances and Schemas

- Databases change over time as information is inserted and deleted.
- The collection of information stored in the database at a particular moment is called an **instance** of the database.
- The overall design of the database is called the database **schema**.
- Schemas are changed infrequently, if at all.
- Database systems have several schemas, partitioned according to the levels of abstraction.

Instances and Schemas

- The **physical schema** describes the database design at the physical level.
- The **logical schema** describes the database design at the logical level.
- A database may also have several schemas at the view level, sometimes called **sub-schemas**, that describe different views of the database.

Data Independence

- The ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called Data Independence
  - Physical Data Independence
  - Logical Data Independence

Physical Data Independence

- It refers to the ability to modify the schema followed at the physical level without affecting the schema followed at the conceptual level.
- The application programs remain the same even though the schema at the physical level gets modified.
- Modifications at the physical level are occasionally necessary in order to improve performance of the system (e.g., creating additional access structure.)

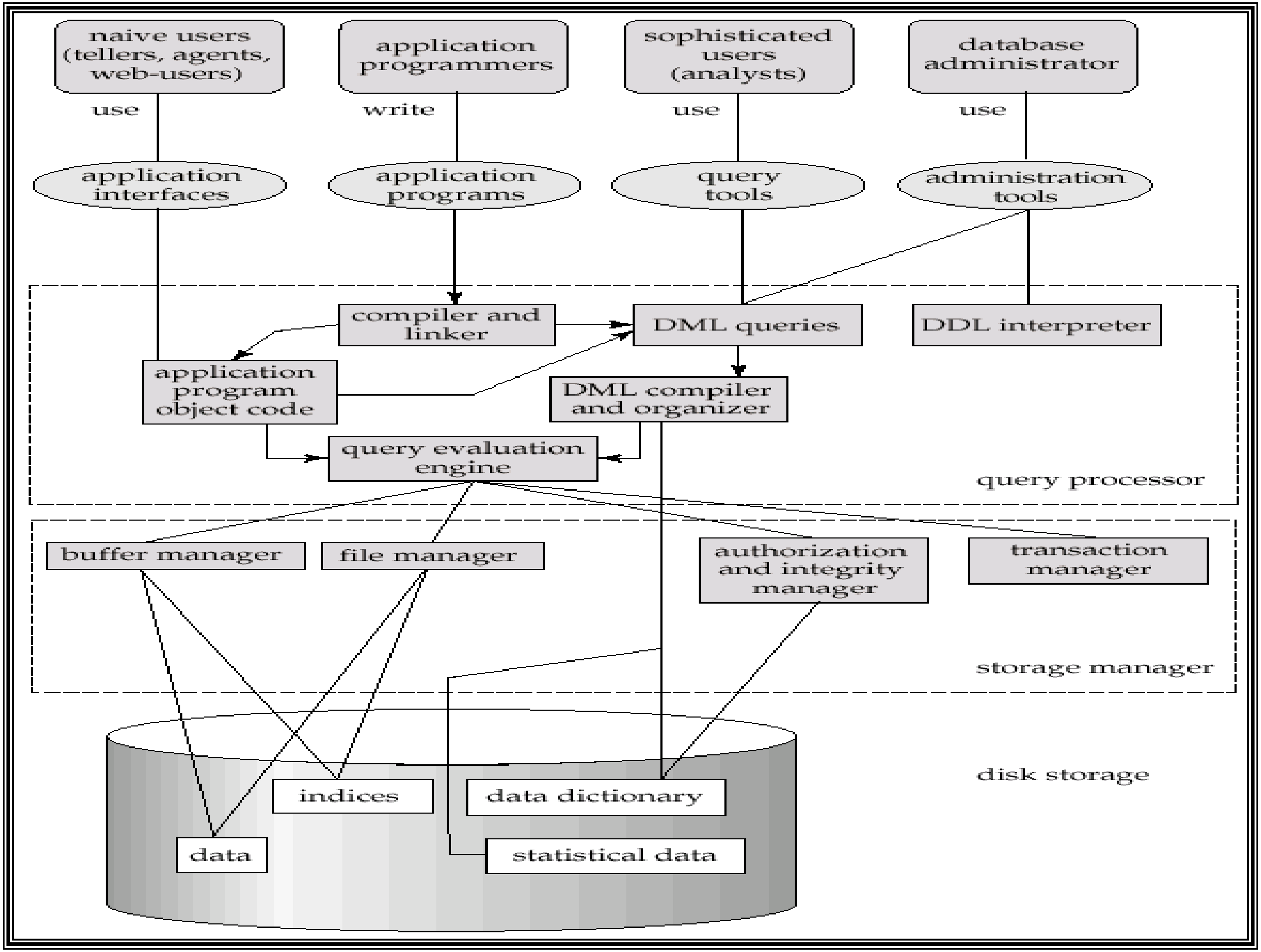
Physical Data Independence

- E.g. : Employee (E#, Name, Address, Salary)
- A view including only E# and Name is not affected by changes in any other attributes

Logical Data Independence

- It refers to the ability to modify the conceptual schema without causing any changes in the schemas followed at view levels.
- The logical data independence ensures that the application programs remain the same.
- Modifications at the conceptual level are necessary whenever logical structures of the database get altered because of some unavoidable reason

Database System Structure


naive users (tellers, agents, web-users)
application programmers
sophisticated users (analysts)
database administrator
use
write
use
use
application interfaces
application programs
query tools
administration tools
compiler and linker
DML queries
DDL interpreter
application program object code
DML compiler and organizer
query evaluation engine
query processor
buffer manager
file manager
authorization and integrity manager
transaction manager
storage manager
disk storage
indices
data dictionary
data
statistical data

Database System Structure

- The overall system structure of DBMS is as shown in figure.
- The functional components of a database system can be broadly divided into two categories as follows :
  - the storage manager
  - the query processor components.

Database System Structure

- A *storage manager* is a program module that provides the 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 translates the various DML statements into low-level file-system commands.
- Thus, the storage manager is responsible for storing, retrieving, and updating data in the database.

Database System Structure

- The storage manager components include:
  - **Authorization and integrity manager :** which tests for the satisfaction of integrity constraints and checks the authority of users to access data.
  - **Transaction manager :** which ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.
  - **File manager :** which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

Database System Structure

- The storage manager components include:
  - **Buffer manager :** which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory.
    - The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory.

Database System Structure

- The storage manager implements several data structures as part of the physical system implementation:
  - **Data files :** which store the database itself.
  - **Data dictionary :** which stores metadata about the structure of the database, in particular the schema of the database.
  - **Indices :** which provide fast access to data items that hold particular values.

Database System Structure

- **The Query Processor :**
- The query processor components include :
  - **DDL interpreter** : which interprets DDL statements and records the definitions in the data dictionary.
  - **DML compiler** : which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.
    - A query can usually be translated into any of a number of alternative evaluation plans that all give the same result.
    - The DML compiler also performs **query optimization**, that is, it picks the lowest cost evaluation plan from among the alternatives.

Database System Structure

- **The Query Processor :**
- The query processor components include :
  - **Query evaluation engine** : which executes low-level instructions generated by the DML compiler.

Database Administrator (DBA)

Database Administrator (DBA)

- One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data.
- A person who has such central control over the system is called a **database administrator** (**DBA**).
- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.

Database Administrator (DBA)

- The functions of a DBA include:
  - Schema definition : The DBA creates the original database schema by executing a set of data definition statements in the DDL.
  - Storage structure and access-method definition.
  - Schema and physical-organization modification : The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.
  - Granting of authorization for data access : By granting different types of authorization, the database administrator can regulate which parts of the database various users can access.
  - Specifying integrity constraints
  - Acting as liaison with users

Database Administrator (DBA)

- The functions of a DBA include:
  - Monitoring performance and responding to changes in requirements.
  - Routine maintenance : examples of the database administrator’s routine maintenance activities are :
    - Periodically backing up the database, either onto tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding.
    - Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.

Database Languages

Database Languages

- A database system provides a **data definition language** to specify the database schema and a **data manipulation language** to express database queries and updates.
- Data Definition Language (DDL)
- Data Manipulation Language (DML)

Data Definition Language (DDL)

- We specify a database schema by a set of definitions expressed by a special language called a **data-definition language** (**DDL**).
- Used by the DBA and database designers to specify the conceptual schema of a database.
- In many DBMSs, the DDL is also used to define internal and external schemas (views).
- DDL compiler is exist to process the DDL statements and store the schema into the DBMS catalog.
- DDL is a set of SQL commands used to create, modify and delete structures but not data.

Data Definition Language (DDL)

- Example

```
create table Employee
        (
            name        char[15]        not null
            age         smallint
            ss#         integer         not null
        )
```

Example of DDL Commands are:

- Alter : Alter the structure of the database
- Drop : Deletes objects from the database
- Truncate : Removes all records from a table, including all spaces allocated for the records are removed.
- Comment : Add comments to the data dictionary.

Data Definition Language (DDL)

- In some DBMSs, separate storage definition language (SDL) and view definition language (VDL) are used to define internal and external schemas.
  - SDL is typically realized via DBMS commands provided to the DBA and database designers.
  - VDL is used to define internal and external schemas

Data Manipulation Language (DML)

- **Data manipulation** is
  - The retrieval of information stored in the database
  - The insertion of new information into the database
  - The deletion of information from the database
  - The modification of information stored in the database.
- A **data-manipulation language (DML)** is a language that enables users to access or manipulate data as organized by the appropriate data model.
- There are basically two types:
  - Procedural (low level) DMLs
  - Declarative (nonprocedural / high level) DMLs

Data Manipulation Language (DML)

- **Procedural (low level) DMLs :**
  - require a user to specify what data are needed and how to get those data.
  - It is used in constructing a complex DB operations concisely.
  - It can be entered interactively from terminal or embedded in a general-purpose language.
  - It can specify and retrieve many records in a single statement (set-at-a-time or set oriented).
  - If it is used in a stand-alone interactive manner, it is called Query language.

Data Manipulation Language (DML)

- **Declarative (nonprocedural / high level) DMLs**
  - require a user to specify what data are needed without specifying how to get those data.
  - It is embedded in a general-purpose-language.
  - It retrieves individual records and processes each separately (record-at-a time)
- In both low and high level languages the general-purpose language is called the host language and DML is called sublanguage.

Data Manipulation Language (DML)

- **Example of DML commands are :**
  - Insert : insert data into a table.
  - Update : updates existing data within a table.
  - Delete : Deletes all records from a table.
- Query :

```
select acc_no, balance
from account
where acc_no=‘A101’;
```

University Questions

1. Describe the overall architecture of a DBMS with a diagram. [7T– 10 M]
2. Explain four main differences between a file processing system and DBMS. [8T – 10 M]
3. What do you mean by DBMS ? Why would you choose a database system instead of simply storing the data in operating system files ? [2T – 10 M]
4. What are the different database users ? [4T– 10 M]
5. Explain three levels of data abstraction in Database Management System [2T – 10 M]

University Questions

6. What are five main functions of database administrator ? [7T – 10 M]

7. List the disadvantages of the file system. Explain how database system can overcome this. [4T – 10 M]

8. Write Short notes on [5 M each]

   i) Database Languages [1T]

   ii) DBA [4T]

   iii) Data Independence and its types [2T]

   iv) 3 layer : data abstraction [1T]

   v) Database Users [2T]