Introduction of DBMS

by

Indrajeet Gupta

(Assistant Professor)

Department of Computer Science Engineering
Bennett University
Greater Noida – 201310

Course Goal

1: To understand the fundamentals relational database systems.

2: To construct databases using DBMS products such as MySQL/Oracle/My SQL Server.

3: Design of real time database systems and understand new developments and trends in databases.

Reference Books

- *Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", 6th Edition, McGraw-Hill , 2010, ISBN:0-07-352332-1.*

- *Elmasri, Ramez and Shamkant B. Navathe, Fundamentals of database systems, (7th Edition) Pearson, 2015. ISBN- 978-0133970777.*

Online Courses

1) Database Systems Concepts and Design: A course on the fundamentals of Relational Database systems.
   https://www.edx.org/course/database-systems-concepts-design-gtx-cs6400x

2) Developing SQL Databases:- Learn the technologies and features needed to create and design SQL databases, including how to implement SQL views, indexes and tables.
   https://www.edx.org/course/developing-sql-databases-microsoft-dat215-1x-1

Evaluation Components

Components of Course Evaluation	Percentage
Mid Term Examination	15
Quiz 1	05
Lab Mid Term Examination	10
Continuous Lab Evaluation	15
Quiz 2	05
End Term Examination	30
End Term Lab Exam (Project)	20

Database Overview

-File Management vs Database Management (why do we need database?)

Advantages of Database systems: storage persistence, programming interface, transaction management

-Data Model (What is Data?)

-Database Language (How to manipulate data?)

-DBMS Architecture and Database System Components (How can you build a billion-dollar software, like Oracle? Or you can get it free, mySQL?)

-Users classification (What you can do and what you cannot do?)

Where is Database?

- You cannot avoid it and it's everywhere!
- You can say it actually makes the current society and your life work!
- Banking/Credit card /Social Security Info…
- 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
- So many fields….

DBMS Marketplace

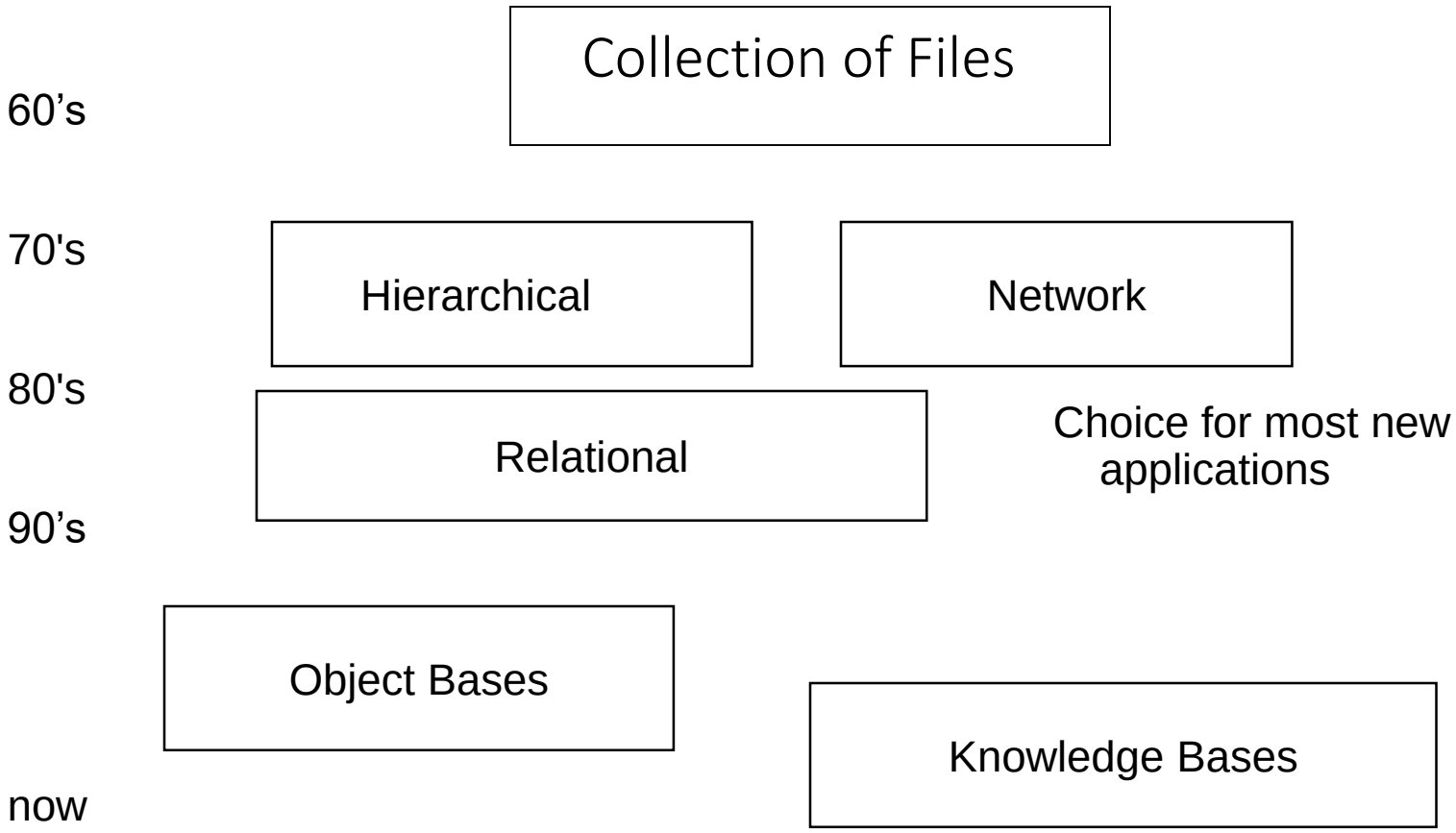
- Relational DBMS companies – Oracle, Sybase – are among the largest software companies in the world.
- IBM offers its relational DB2 system. With IMS, a nonrelational system, IBM is by some accounts the largest DBMS vendor in the world.
- Microsoft offers SQL-Server, plus Microsoft Access for the cheap DBMS on the desktop, answered by “lite” systems from other competitors.
- OpenSource: mySQL, postgreSQL

Pre-Database Era

- Imagine you want build an online shopping website
  - Maintain products/categories (price, picture, properties, …)
  - Customers accounts
- File is uninterpreted, unstructured collection of information
- File operations: delete, catalog, create, rename, open, close, read, write, find, …
- Access methods: Algorithms to implement operations along with internal file organization
- Examples: File of Customers, File of Products; Access method: implementation of a set of operations on those files

File Management System Problem

- Any question (access) on the data is a small program!!
- Data redundancy
- Data is not isolated from the access implementation (different format…)
- Multiple application (concurrent program) on the same file


60’s
Collection of Files
70's
Hierarchical
Network
80's
Relational
Choice for most new applications
90’s
Object Bases
Knowledge Bases
now

Case 1: Amazon Database

Amazon uses their own proprietary NoSQL database for their humongous product and marketplace info which is scaled horizontally and renders very many pages, and is dynamic.

However, Amazon does use Relational Databases for their own human resources management. For instance, Amazon is a major Oracle client, spending some 50 million dollars on RDMS.

The databases presented my AWS is to be used by AWS clients and is for hosting and that includes DynamoDB which is a relational database.

Case 2: Google Database

Although Google uses **BigTable** for all their main applications, they also use **MySQL**for other (perhaps minor) apps. And it's maybe also handy to know that **BigTable** is not a relational database (like **MySQL**) but a huge (distributed) **hash table** which has very different characteristics.

Task for Students

1) Does Facebook use the RDBMS?
2) Does Salesforce use RDBMS?
3) Does Microsoft use RDBMS?
4) Enlist the name of anyone big IT Company which use RDBS frequently.

Modern Database Applications

- Client – Server architecture
  - DBMS serves as a server and client queries are sent to servers
  - Where to locate servers
- Multimedia Applications
- Multidatabase Applications
- Data Warehouses
- It’s everywhere!!

Three Aspects to Studying DBMS's

1. Modeling and design of databases.
   - Allows exploration of issues before committing to an implementation.
2. Programming: queries and DB operations like update.
3. DBMS implementation.

What is Data?

- Different view points:
  - A sequence of characters stored in computer memory or storage
  - *Interpreted* sequence of characters stored in computer memory or storage
  - *Interpreted set of objects*
- This maybe one of the most profound questions in computer science! It is still open and keep evolving!!

Data is the fact or information which is storable.

What is DataBase?

Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

DBMS

DBMS stands for **D**ata**b**ase **M**anagement **S**ystem. We can break it like this DBMS = Database + Management System. Database is a collection of data and Management System is a set of programs to store and retrieve those data.

Based on this we can **define DBMS** like this: DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

- Represents some aspect of the real world (universe of discourse) generally relevant to an enterprise/company/organization
- Logically coherent
- Organized to reflect relationships among the data
- Persistent (not lost in power outage or hardware failure)
- Mirrors the state of the company/organization/enterprise, an asset in its own right (account balances, inventory, employee status, locations of people/things)
- Usually a specific purpose and for a set of users when built -- but a good design should allow for uses that are unanticipated.

Use of DBMS

• Consumer websites • Search engines • Travel reservations • Online banking • Health care • Libraries	Advanced applications • Geographic Information Systems • Software development • Scientific research • Decision Support Systems • Customer Relations Management

Purpose of DBMS

- It is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

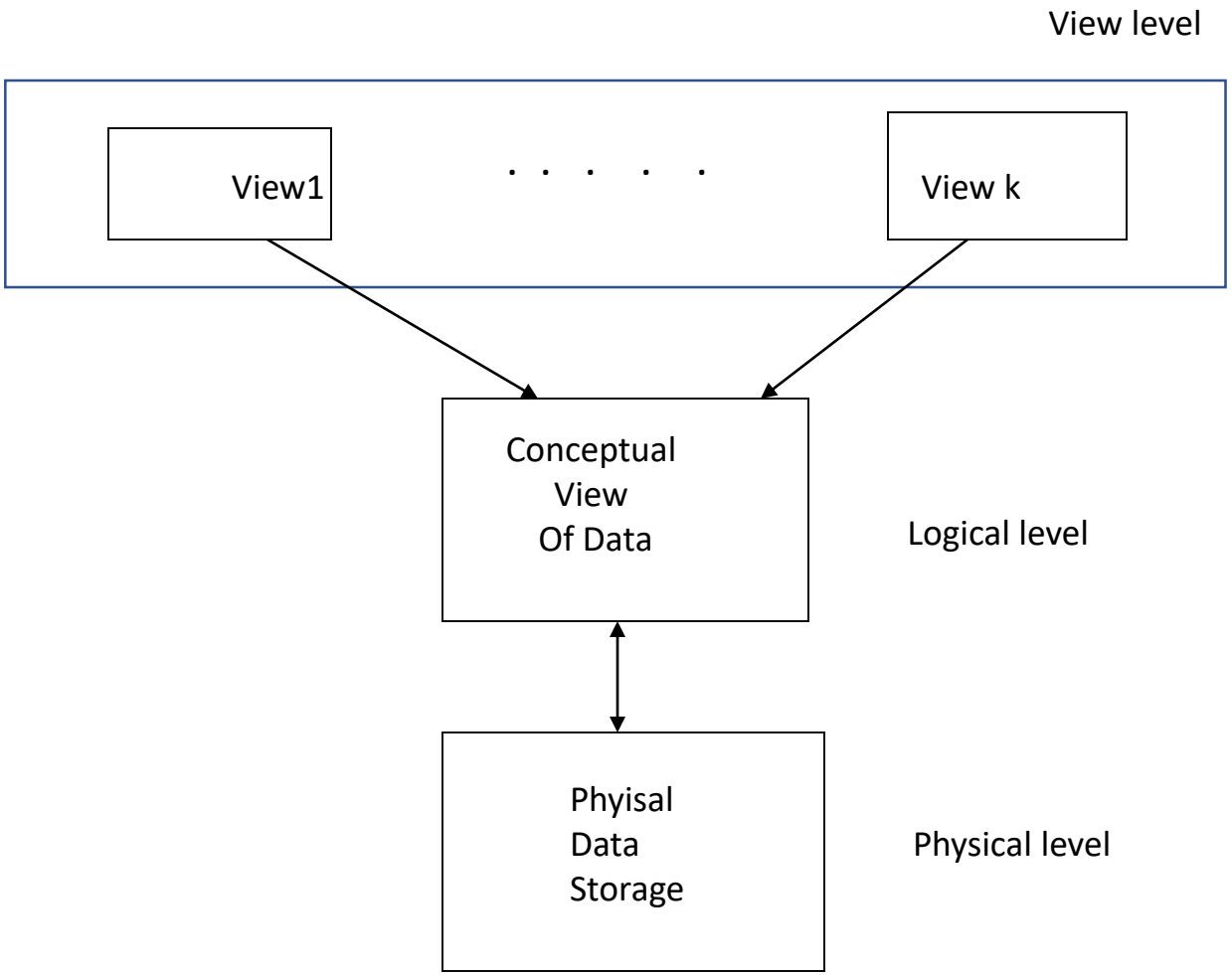
- Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results.

View of Data

Data Levels and their Roles (Data Abstraction)

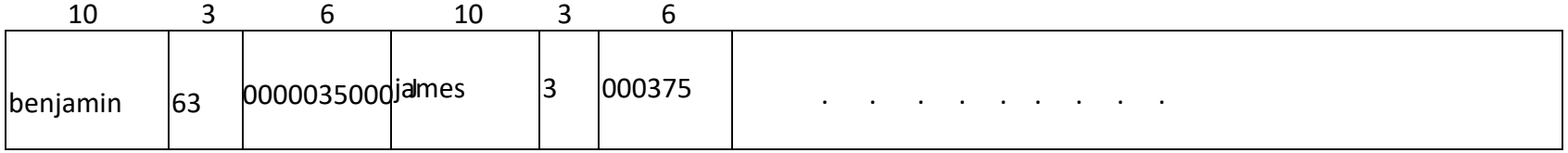
- *Physical – corresponds to the first view of data*: How data is stored, how is it accessed, how data is modified, is data ordered, how data is allocated to computer memory and/or peripheral devices, how data items are actually represented (ASCI, EBCDIC,…)
- *Conceptual – corresponds to the second view of data*: What we want the data to express and what relationships between data we must express, what " story" data tells, are all data necessary for the "story' are discussed.
- *View – corresponds to the third view of data:*What part of the data is seen by a specific application

Three Level Data View – Data Abstractions



Physical Data - Example

- Physical



Examples

- Conceptual/Logical
  - TA

    Name char(10),

    Age char (3),

    Salary Fixed Dec(6);
  - Student

    Name char(10),

    Year-of_study char(3)

    GPA Fixed Dec(5,2);

Examples

- STUDENTS-TA

  Name char(25),

  Age char (3),

  Salary Fixed Dec(8,2), —— A view

  Year-of_study char(3)

  GPA Fixed Dec(3,2);

Data model, Schema and Relation

- A *data model* is a collection of concepts for describing data.
- A *schema* is a description of a particular collection of data, using the a given data model.
- The *relational model of data* is the most widely used model today.
  - Main concept: *relation*, basically a table with rows and columns.
  - Every relation has a *schema*, which describes the columns, or fields.

Schema and Instance

- The data which is stored in the **database** at a particular moment of time is called an **instance** of the **database**.

- The overall design of a **database** is called **schema**. A **schema** contains **schema** objects like table, foreign key, primary key, views, columns, data types, stored procedure, etc.

Data Models

- Underlying the structure of a database is the data model.
- A collection of conceptual tools for describing **data, data relationship, data semantic and consistency constraints**.
- A data model provides a way to describe the design of the database at the Physical, Logical, and View level.

Data Models

- Hierarchical
- Network
- **Relational**
- **Entity relationship**
- **Object oriented (OO)**

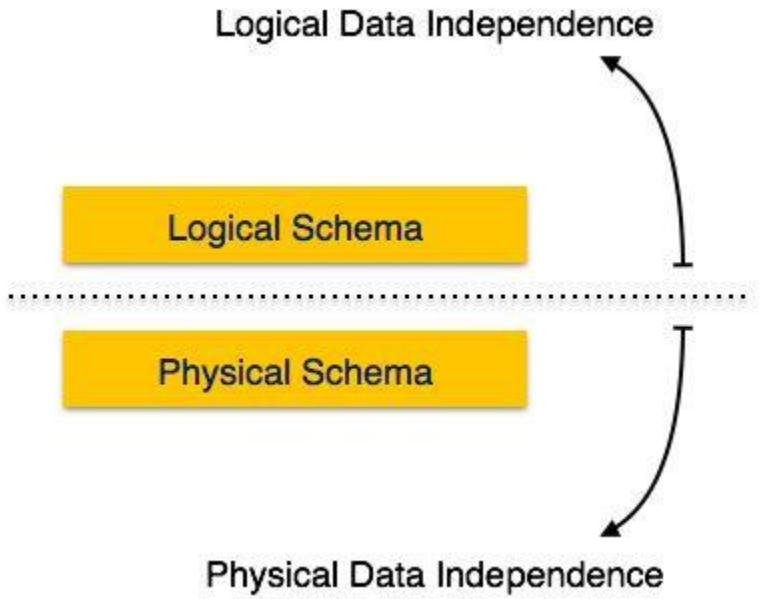
Note: The last 3 models would be studied in the upcoming chapters Hierarchical and Network model will be covered later on because they are used little now.

Students are advised to study about these models from book of Henry F. Korth. (Database System Concepts)

Data Independence

A database system normally contains a lot of data in addition to users' data. For example, it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.

Physical and Logical Data Independence



Logical data independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk.

All the schemas are logical, and the actual data is stored in bit format on the disk. Physical data independence is the power to change the physical data without impacting the schema or logical data.

DBMS Languages

DBMS system provides a **Data Definition Language** to specify the database schema and a **data-manipulation language** to express database queries and updates.

In Practice, the DDL and DML are not to separate languages; instead they simply form parts of a single database language, such as the widely used SQL language.

Data Definition Language

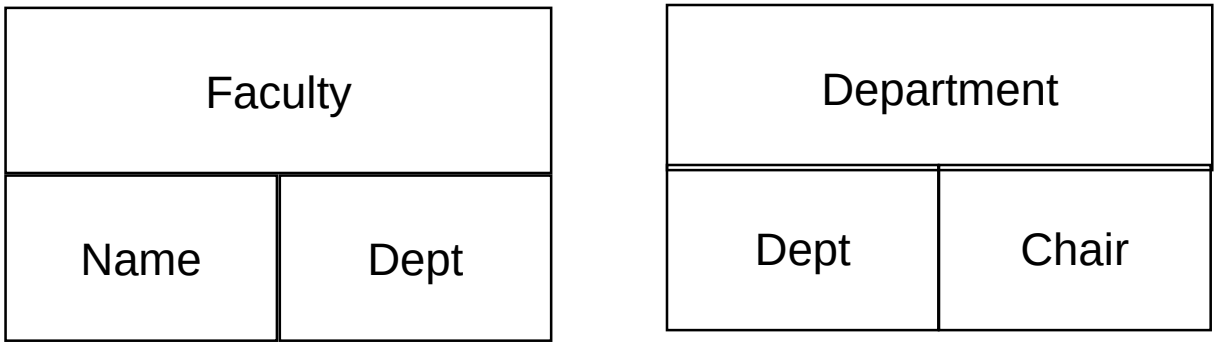
- Specification notation for defining the database schema
  - E.g.

    create table *account* (
    account-number **char**(10),
    balance **integer**)
- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  - database schema
  - Data *storage and definition* language
    - language in which the storage structure and access methods used by the database system are specified
    - Usually an extension of the data definition language

Data Manipulation Language:

- Language for accessing and manipulating the data organized by the appropriate data model
- Two classes of languages
  - Procedural – user specifies what data is required and how to get those data
  - Nonprocedural – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

Database Languages



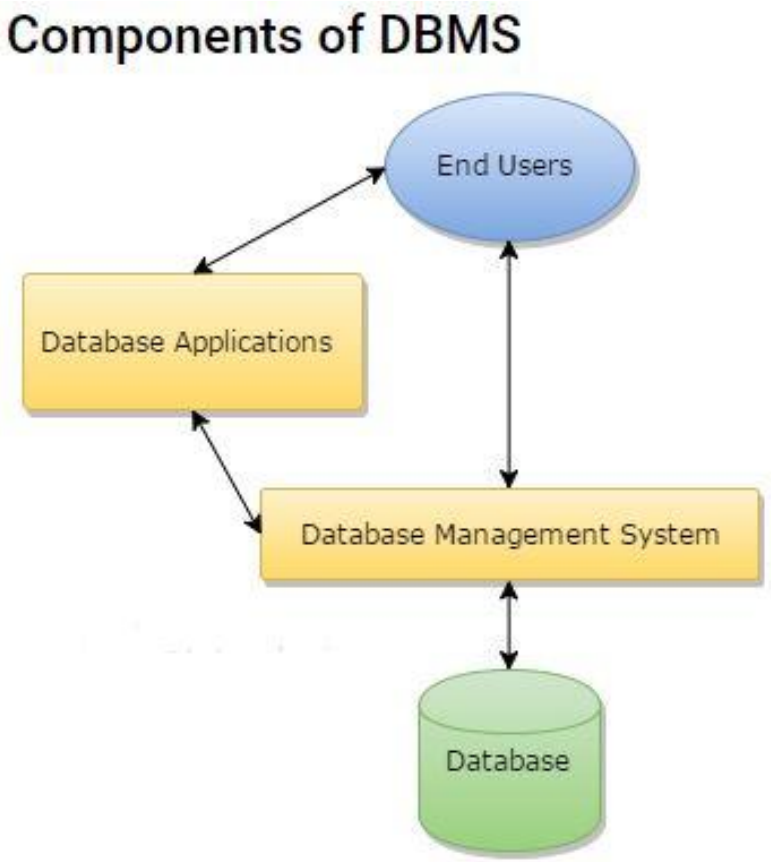
SQL

SELECT Chair

FROM Faculty, Department
WHERE Faculty.name = “Ken Noname”
AND Faculty.Dept = Department.Dept

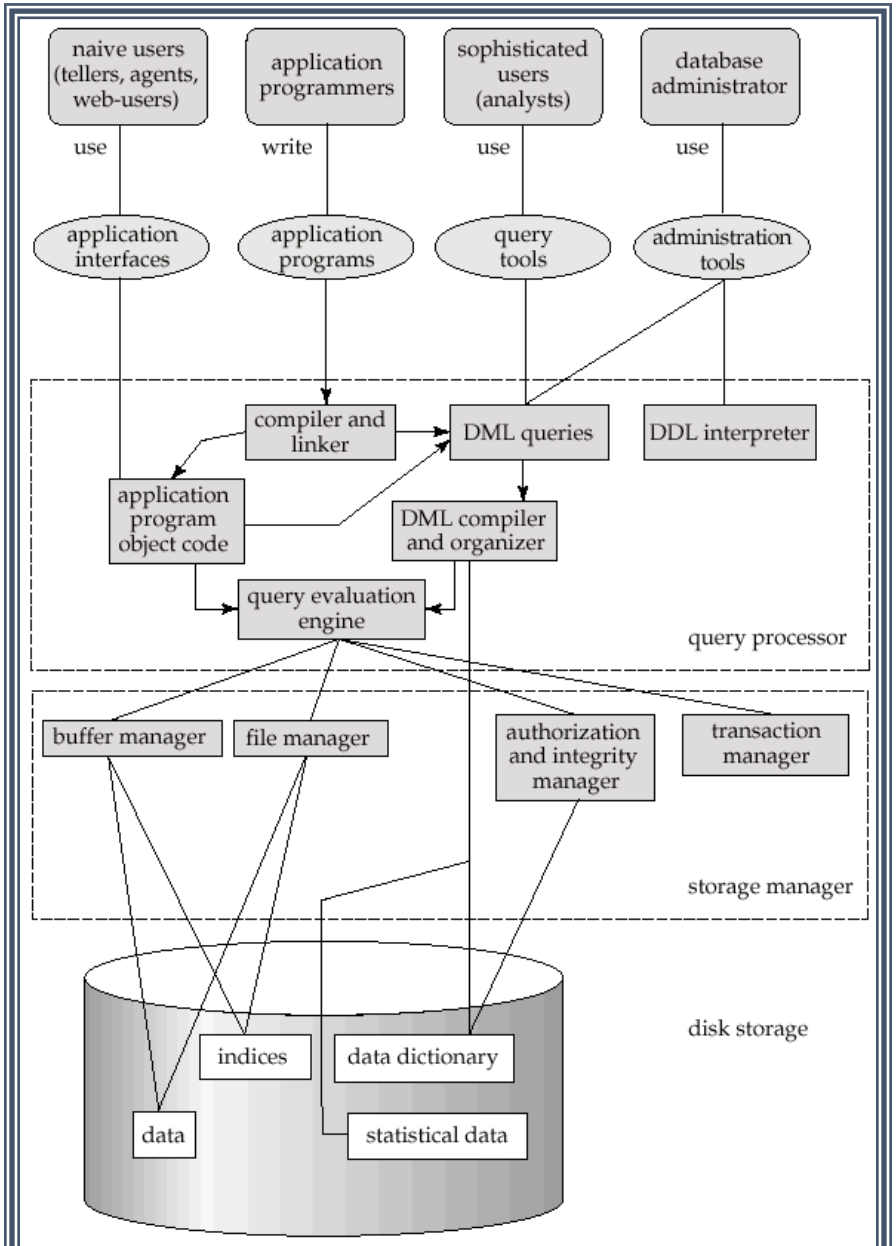
Data definition language (DDL) ~ like type definitions in C or C++

Data Manipulation Language (DML)
Query (SELECT)
UPDATE < relation name >
SET <attribute> = < new-value>
WHERE <condition>



•Users: Users may be of any kind such as DB administrator, System developer or database users.
•Database application: Database application may be Departmental, Personal, organization's and / or Internal.
•DBMS: Software that allows users to create and manipulate database access,
•Database: Collection of logical data as a single unit.

DBMS Architecture



Logical and Physical Database Components

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Host Language Interface
- Data Administrator
- Users
- Query Processor
  - Compiler
  - Optimizer
- Management
  - Transaction Manager
  - File Manager
  - Buffer Manager
  - Authorization and Integrity Manager

Logical

Physical

Query Processor

- Compiler – verifies whether a program or query is written in accordance with DDL and DML rules
- Optimizer – Finds the most effective way to access the required data and supply it in a user requested form. Monitors the query execution and modifies a query evaluation plan if necessary.

Storage Management

- 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 to the following tasks:
  - interaction with the file manager
  - efficient storing, retrieving and updating of data

Transaction Manager

- A *transaction* is a collection of operations that performs a single logical function in a database application
- Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

File Manager

- File Manager is responsible for mapping logical database units (objects, relations, etc.) into a set of low level files.
- It is responsible for maintenance of files and indexes on them. It should be able to create and destroy index and collect unused storage space to eliminate an unneeded gaps on disks.

Buffer Manager

- Buffer Manager is responsible for the allocation and maintenance buffer space in a memory to facilitate processing database data by several concurrent applications.
- Buffer Manager decides when to load data from a buffer to a database or discard the data and under what conditions a new data should be put into a buffer

Authorization and Integrity Manager

- This manager is responsible for granting an access to database or portions thereof only to authorized users and preventing the access to unauthorized users
- Integrity manager must assure data integrity during normal database operations as well as during the database failures

Data Administrator

- 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's duties include:
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements

Database Users

- Naïve – do not know about database too much, invoke application programs that are prepared already
- Application Programmers – know how to interact with the system but may not know how DBMS is designed
- Sophisticated users that know advanced use of the system and can use the system and packages on the top of the system
- DBMS system users – write specialized database applications that do not fit into the traditional data processing framework

Database Design

- Design Process
- ER Diagram
- Relational Data Model
- Normalization
- Concurrency control and recovery
- Access control and database security

Note that: the above-mentioned topics will be covered one by one in upcoming chapters.

Assignments for Students

- Describe 3 disadvantages of DBMS.
- Difference between DBMS and Data warehouse.
- Write major steps for setting up your database for an enterprise.
- What are the main functions of a database administrator?
- Draw at least one relational table named as Student which consists of yours, name , admn. no. , city, CGPA, contact no. current semester subjects and teachers as the attributes of table