GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY COLLEGE
(Autonomous)

DATABASE MANAGEMENT SYSTEM

Reaccredited by NAAC with A+

By

Mr N Kartik & Mr Jeelan
Assistant Professor, Dept of Computer Applications,
Presidency College, Bangalore-24

Presidency Group

OVER 40 YEARS OF ACADEMIC WISDOM

PRESIDENCY COLLEGE
(AUTONOMOUS)
AFFILIATED TO BENGALURU CITY UNIVERSITY, APPROVED BY AICTE, DELHI & RECOGNISED BY THE GOVT. OF KARNATAKA
RE-ACCREDITED BY NAAC WITH 'A+' GRADE

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Unit 1 Introduction to Database Management System

Introduction

Characteristics of the Database Approach

Advantages of Using DBMS Approach

Transaction management and Structure of a DBMS

Database System Concepts and Architecture: Data Models, Schemas, and Instances

The 3-level architecture of DBMS and Data Independence

Hierarchical, Network, and Relational Model

Database Languages and Interfaces

The Database System Environment

Centralized and Client-Server Architectures

Classification of Database Management Systems

Queries in DBMS

Introduction

Data:

Data is a collection of facts, figures and statistics that can be recorded and have an implicit meaning.

e.g., names, telephone numbers, addresses.

Information:

Information is a processed data with implicit meaning.

Database:

A database is a collection of logically related records, that is organized so that it can easily be accessed, managed, and updated. It consists of an organized collection of data for one or more uses, typically in digital form.

Introduction:Why database?

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

Introduction:Examples of Database

- Banking system
- Hospital management system
- University database
- Library database
- Airline reservations system etc.

Some of the commercial databases such as Amazon, Flipkart, Facebook, WhatsApp etc.

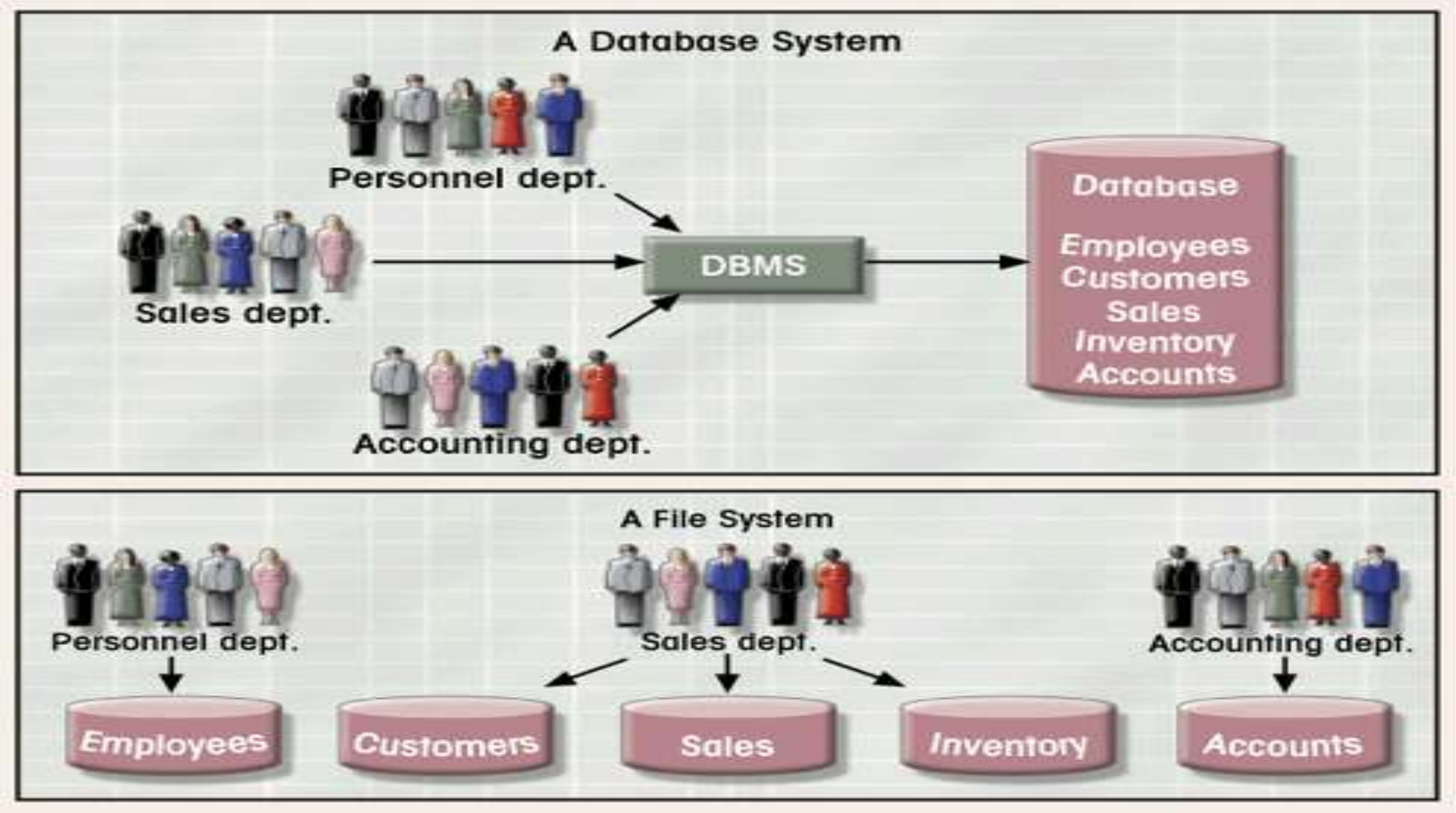
Note: **Earlier data was maintained using traditional file processing system. i.e., flies of records.**

Amazon Database Example

An example of large commercial database is Amazon.com , which contains a data of over 200 million books, CDs, Videos, DVDs, games, electronics, apparel & other items. In this database , data is stored on more than 200 servers. About 500 million visitors access Amazon.com each day & use the database to make purchases. And the database is continuously updated.

About 500 people are responsible for keeping the Amazon.com database up to date.

Database System vs. File System



What is DBMS

Database Management System (DBMS):

A database management system (**DBMS**) is system software for creating and managing databases

The **DBMS** provides users and programmers with a systematic way to create, retrieve, update and manage data

Database System:

Database system is a combination of both database & DBMS software

DBMS Functionality

The DBMS is a general-purpose software system ,basic functions of DBMS are

1. **Defining,**
2. **Constructing and**
3. **Manipulating databases for various applications**

Other features or functionalities are

- Protection or Security
- Presentation and Visualization of data
- Maintaining the database

Defining Database

1.Defining a database involves specifying the data types, structures and constraints for the data to be stored in the database.

Ex: Let us construct the process of creating the **EMPLOYEE database**

Steps involved in defining a database are:

- Specifying the data types and structures of each field in the database.
- Identifying the constraints on different elements.

Defining Database

Entity Set	Attribute	Data type	Constraints
EMPLOYEE	Emp_name	Char(20)	Only alphabet
	Emp_no	Number(6)	Value > 0
	Emp_address	Char(50)	-
	Emp_designation	Char(10)	-
	Emp_department	Char(10)	-
	Emp_salary	Number(6,2)	-

Constructing Database

2.Constructing the database is a process of storing the data itself on some storage medium that is controlled by DBMS. Eg;EMPLOYEE

Emp_name	Emp_no	Emp_address	Emp_designation	Emp_department	Emp_salary
Arun	101	#18,Hebbal, Bangalore	Manger	Sales	30237.85
Amith	102	#27,Hebbal Bangalore	Admin	Administration	35000.54
Anish	108	Yelahanka, Bangalore	Accountant	Accounts	60,000.55

Manipulating Database

3.**Manipulating** a database includes functions as querying the database to retrieve specific data, updating the database to reflect changes and generating reports from the data.

Ex: List all employees whose salaries are greater than 50,000.00

Characteristics of the Database Approach

- Self describing nature of database system
- Insulation between program & data
- Data abstraction
- Supports multiple views of data
- Sharing of data & multi-user transaction processing

Characteristics of the Database Approach

Self-describing nature of a database system:

The database system not only contains the data , but also the description of the data.

Insulation between programs and data:

- Called **program-data independence**.
- Allows changing data structures and storage organization without having to change the DBMS access programs.

PDI [Program Data Independence]

- In database system environment , if both data & program is stored independent , it is known as Program Data Independence.

POI [Program Operation Independence]

- In database system environment , if both data & program is operated independent , it is known as Program Operation Independence

Characteristics of the Database Approach

Data Abstraction:

- The characteristic that allows program-data independence and program-operation independence called data abstraction.
- A **data model** is used to hide storage details and present the users with a conceptual view of the database.

Support of multiple views of the data:

- Each user may see a different view of the database, which describes **only** the data of interest to that user.

Sharing of data and multi-user transaction processing:

- Allowing a set of **concurrent users** to retrieve from and to update the database.

Advantages of using DBMS

1. Controlling Redundancy
2. Restricting unauthorized access of database
3. Providing persistent storage of program objects
4. Provides storage structure for efficient query processing
5. Provides Back up & recovery
6. Provides multiple user interfaces
7. Representing complex relationship among data
8. Enforcing integrity constraints

Advantages of using DBMS

1.Controlling Redundancy

Redundancy leads to several problems such as “Duplication of efforts ” & inconsistency

2. Restricting unauthorized access of database

Depending on the type of data only limited users are allowed to access the database

3. Providing persistent (constant) storage of program objects

DBMS provides an environment to provide persistent storage of program objects.

4.Provides storage structure for efficient query processing

DBMS provides facilities for storage structure & efficient execution of queries & updates

5.Provides Back up & recovery

DBMS provides facilities to recover from hardware and software failures . i.e., back up and recovery mechanism

Advantages of using DBMS

6.Provides multiple user interfaces

- Query language for Casual users
- Programming language for Application programmers
- Form & command codes for parametric users
- Menu based (graphics based) interfaces for stand-alone users

7.Representing complex relationship among data

- DBMS can represent a variety of relationships among the data

8. Enforcing integrity constraints

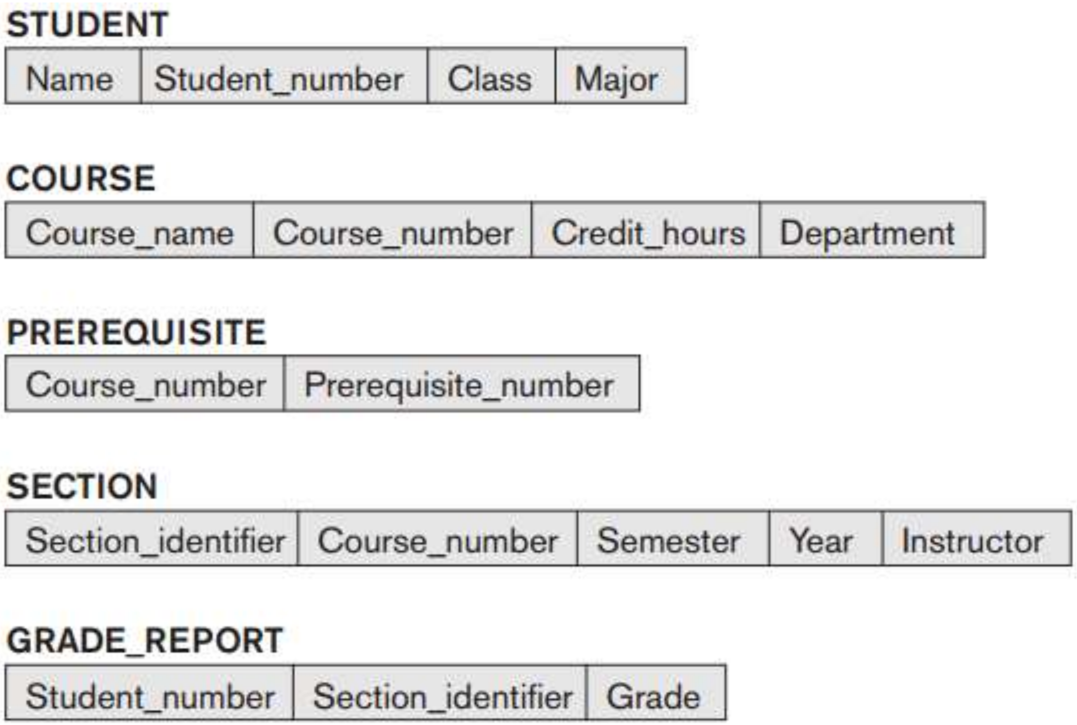
- DBMS provides facilities for defining & enforcing integrity constraints
- Simplest integrity constraint is specifying data type for each data item
- Ex: Primary key , Foreign Key , Referential integrity

Data Models, Entity, Attribute, Relationship

- A **data model**—a collection of concepts that can be used to describe the structure of a database—provides the necessary means to achieve this abstraction.
- By structure of a database, we mean the data types, relationships, and constraints that apply to the data
- An **entity** represents a real-world object or concept, such as an employee or a project from the miniworld that is described in the database
- An **attribute** represents some property of interest that further describes an entity, such as the employee’s name or salary
- A **relationship** among two or more entities represents an association among the entities, for example, a works-on relationship between an employee and a project

Schemas, Instances and Database State

- The description of a database is called the **database schema**, which is specified during database design and is not expected to change frequently.



STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

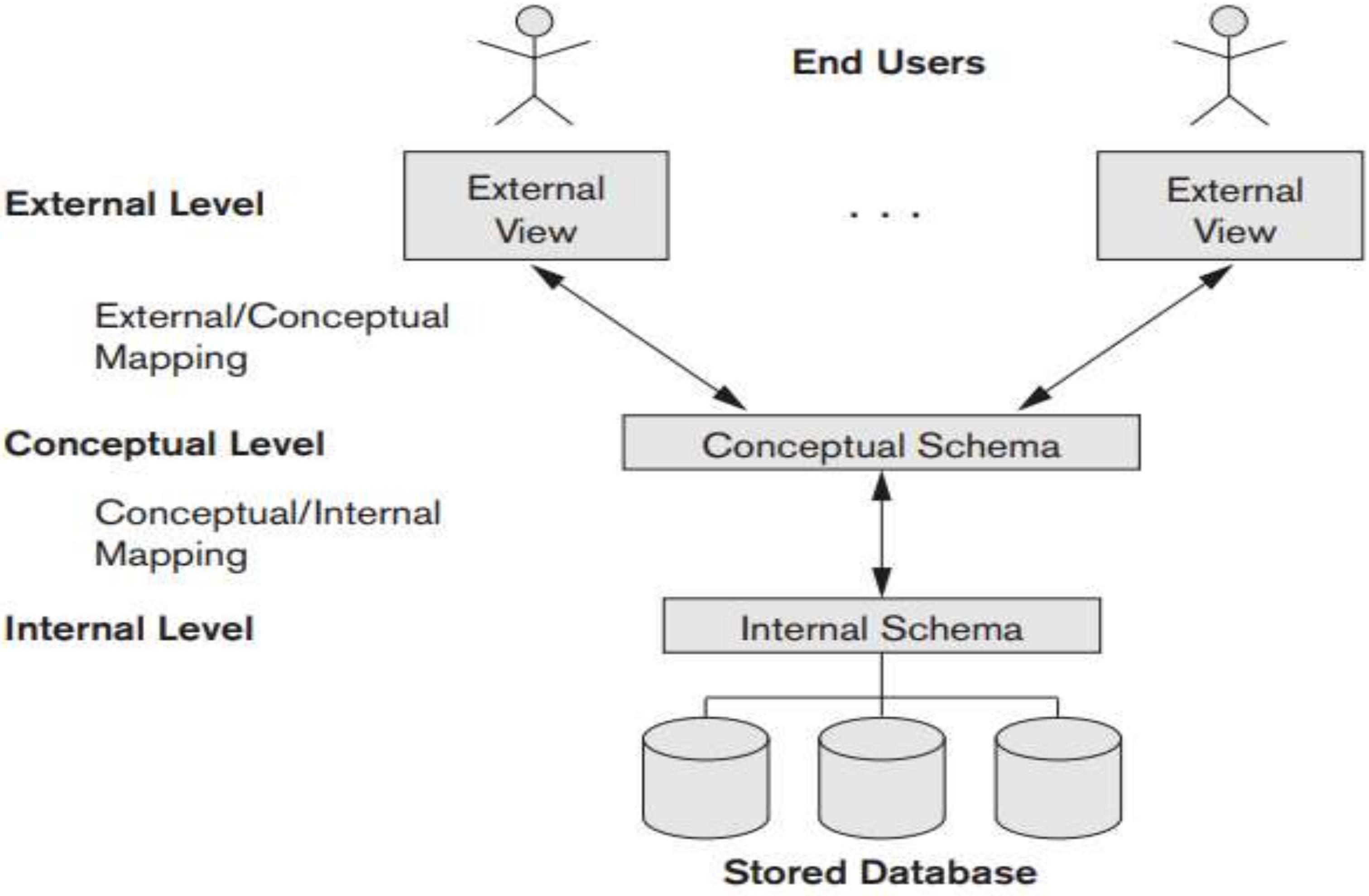
GRADE_REPORT

Student_number	Section_identifier	Grade

Schemas, Instances and Database State

- 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.

Three Schema Architecture



Three Schema Architecture

- The **internal level** has an **internal schema**, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database
- The **conceptual level** has a **conceptual schema**, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.
- The **external** or **view level** includes a number of **external schemas** or **user views**. 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

Data Independence

- **Logical data independence** is the capacity to change the conceptual schema without having to change external schemas or application programs. Ex: Adding a tuple
- **Physical data independence** is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not be changed as well. Ex: Creating a table

Database Languages

- **Data definition language** (**DDL**) is used by the DBA and by database designers to define both schemas. The DBMS will have a DDL compiler whose function is to process DDL statements in order to identify descriptions of the schema constructs and to store the schema description in the DBMS catalog.
- **Storage definition language** (**SDL**), is used to specify the internal schema.
- **View definition language** (**VDL**), is used specify user views and their mappings to the conceptual schema, but in most DBMSs *the DDL is used to define both conceptual and external schemas*.
- Manipulations include retrieval, insertion, deletion, and modification of the data. The DBMS provides a set of operations or a language called the **Data Manipulation Language** (**DML**) for these purposes.

DBMS Interfaces

- **Menu-based Interfaces for Web Clients or Browsing**. These interfaces present the user with lists of options (called **menus)** that lead the user through the formulation of a request.
- **Apps for Mobile Devices**. These interfaces present mobile users with access to their data. For example, banking, reservations, and insurance companies, among many others, provide apps that allow users to access their data through a mobile phone or mobile device.
- **Forms-based Interfaces**. A forms-based interface displays a form to each user. Users can fill out all of the **form** entries to insert new data, or they can fill out only certain entries, in which case the DBMS will retrieve matching data for the remaining entries.
- **Graphical User Interfaces**. A GUI typically displays a schema to the user in diagrammatic form. The user then can specify a query by manipulating the diagram. In many cases, GUIs utilize both menus and forms.

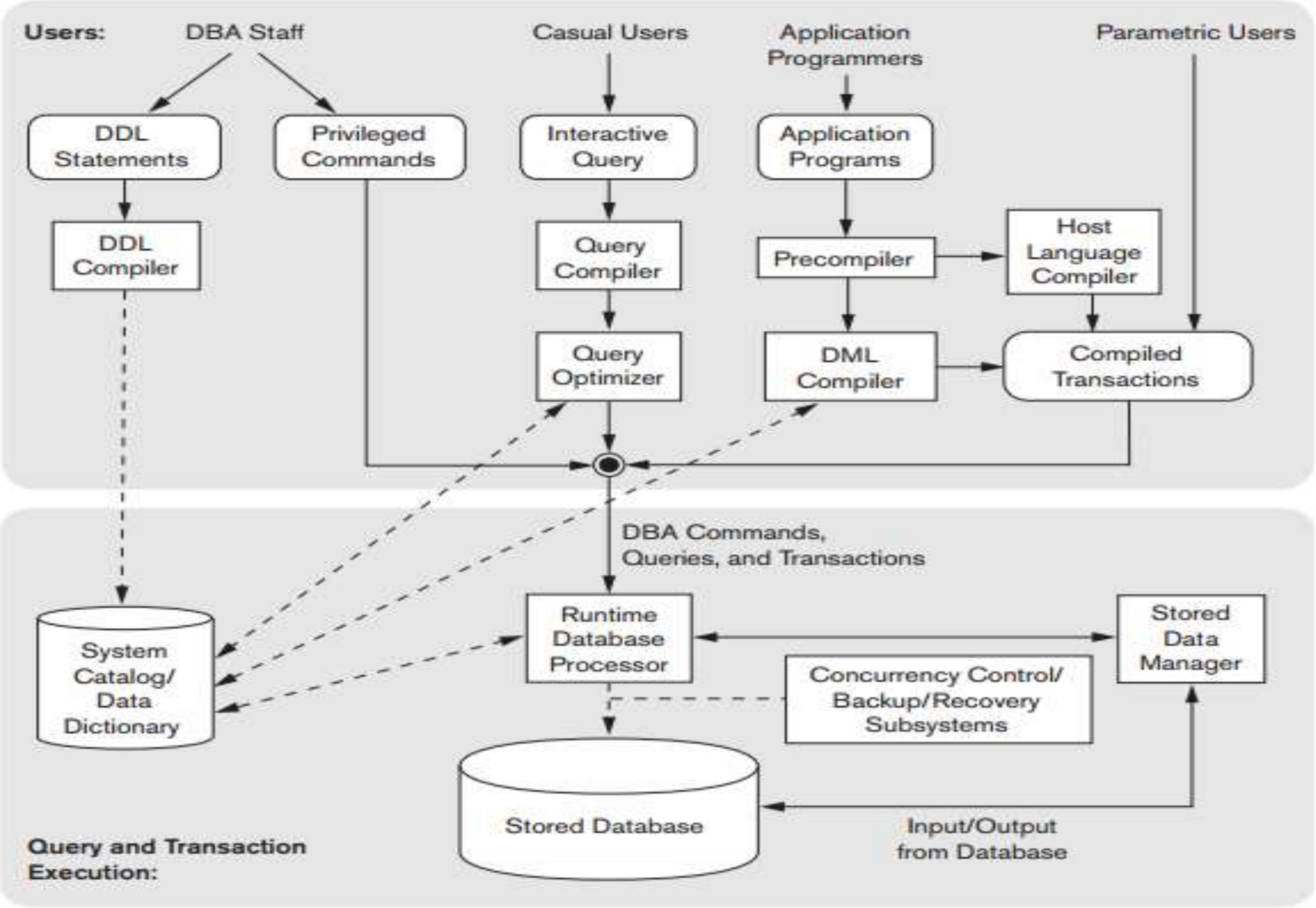
DBMS Interfaces

- **Natural Language Interfaces**. These interfaces accept requests written in English or some other language and attempt to *understand* them. A natural language interface usually has its own *schema*, which is similar to the database conceptual schema, as well as a dictionary of important words.
- **Keyword-based Database Search**. These are somewhat similar to Web search engines, which accept strings of natural language (like English or Spanish) words and match them with documents at specific sites (for local search engines) or Web pages on the Web at large (for engines like Google or Ask)
- **Speech Input and Output.** Limited use of speech as an input query and speech as an answer to a question or result of a request is becoming commonplace. Applications with limited vocabularies, such as inquiries for telephone directory, flight arrival/departure, and credit card account information, are allowing speech for input and output to enable customers to access this information

DBMS Interfaces

- **Interfaces for Parametric Users**. Parametric users, such as bank tellers, often have a small set of operations that they must perform repeatedly. For example, a teller can use single function keys to invoke routine and repetitive transactions such as account deposits or withdrawals, or balance inquiries
- **Interfaces for the DBA.** Most database systems contain privileged commands that can be used only by the DBA staff. These include commands for creating accounts, setting system parameters, granting account authorization, changing a schema, and reorganizing the storage structures of a database.

The Database System Environment



The Database System Environment

- The top part of the figure refers to the various users of the database environment and their interfaces
- The lower part shows the internal modules of the DBMS responsible for storage of data and processing of transactions
- Casual users who work with interactive interfaces to formulate queries, application programmers who create programs using some host programming languages,
- Parametric users who do data entry work by supplying parameters to predefined transactions. The DBA staff works on defining the database and tuning it by making changes to its definition using the DDL and other privileged commands
- The **query optimizer** is concerned with the rearrangement and possible reordering of operations, elimination of redundancies, and use of efficient search algorithms during execution.

The Database System Environment

- **Precompiler** extracts DML commands from an application program written in a host programming language. These commands are sent to the DML compiler for compilation into object code for database access. The rest of the program is sent to the host language compiler
- the **runtime database processor** executes (1) the privileged commands, (2) the executable query plans, and (3) the canned transactions with runtime parameters. It works with the **system catalog** and may update it with statistics.
- It also works with the **stored data manager**, which in turn uses basic operating system services for carrying out low-level input/output (read/write) operations between the disk and main memory
- **Concurrency control** and **backup and recovery systems** separately as a module in this figure. They are integrated into the working of the runtime database processor for purposes of transaction management.

The Database System Environment

Database System Utilities

Loading. A loading utility is used to load existing data files—such as text files or sequential files—into the database. Usually, the current (source) format of the data file and the desired (target) database file structure are specified to the utility, which then automatically reformats the data and stores it in the database.

Backup. A backup utility creates a backup copy of the database, usually by dumping the entire database onto tape or other mass storage medium. The backup copy can be used to restore the database in case of catastrophic disk failure.

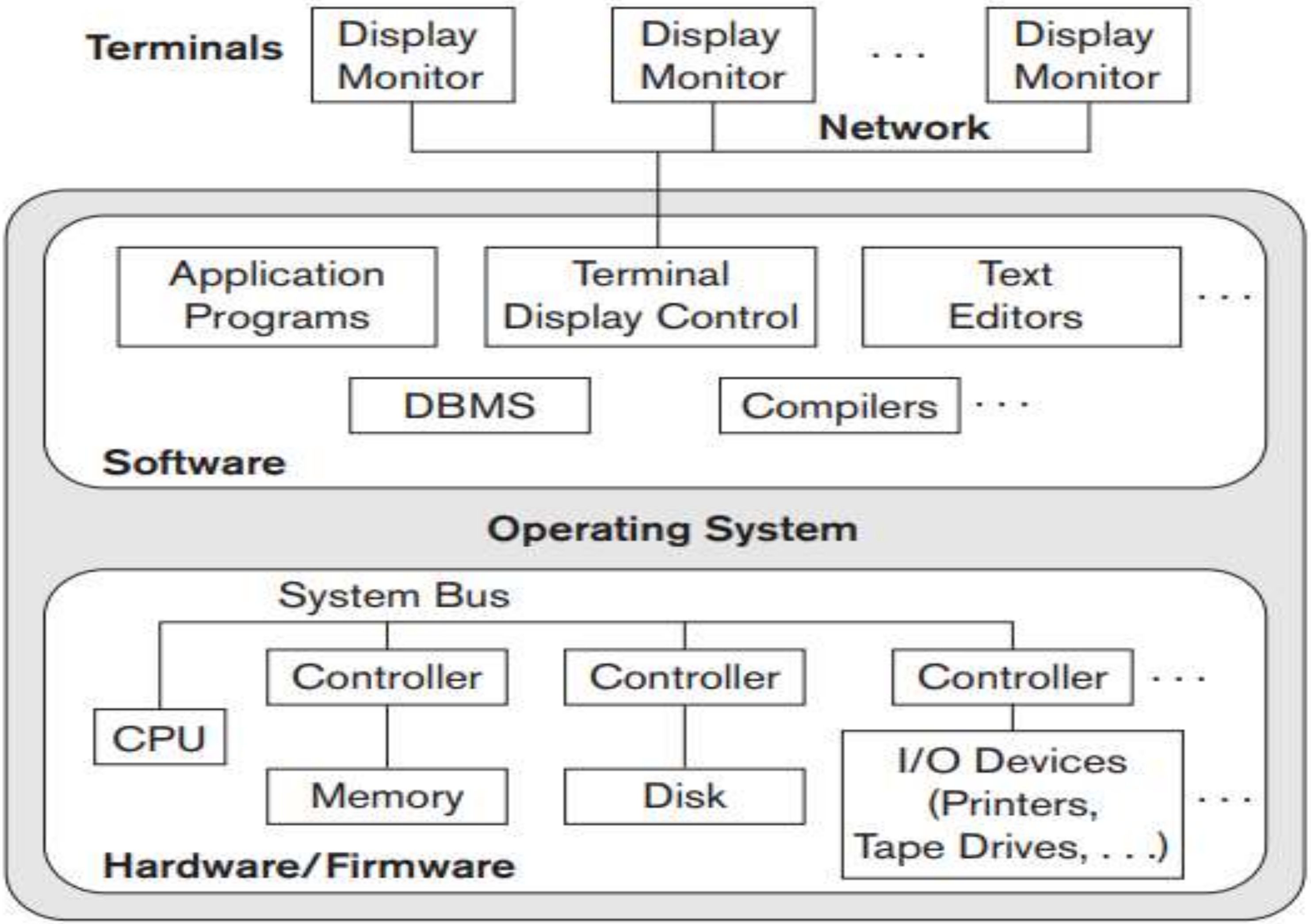
Database storage reorganization. This utility can be used to reorganize a set of database files into different file organizations and create new access paths to improve performance.

Performance monitoring. Such a utility monitors database usage and provides statistics to the DBA. The DBA uses the statistics in making decisions such as whether or not to reorganize files or whether to add or drop indexes to improve performance.

Centralized DBMS Architecture

- All processing was performed remotely on the computer system housing the DBMS, and only display information and controls were sent from the computer to the display terminals, which were connected to the central computer via various types of communications networks
- Database systems used these computers similarly to how they had used display terminals, so that the DBMS itself was still a **centralized** DBMS in which all the DBMS functionality application program execution, and user interface processing were carried out on one machine

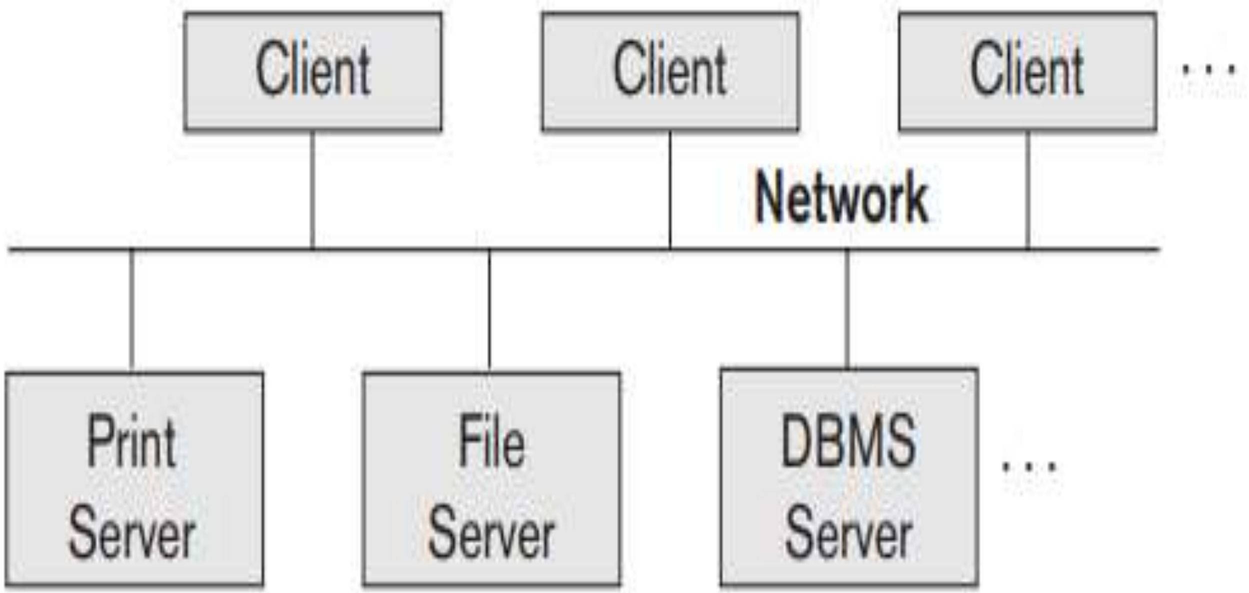
Centralized DBMS Architecture



Client Server Architecture

- The **client/server architecture** was developed to deal with computing environments in which a large number of PCs, workstations, file servers, printers, database servers, Web servers, e-mail servers, and other software and equipment are connected via a network.
- The idea is to define **specialized servers** with specific functionalities.
- A **client** in this framework is typically a user machine that provides user interface capabilities and local processing.
- A **server** is a system containing both hardware and software that can provide services to the client machines, such as file access, printing, archiving, or database access.

Client Server Architecture



Classification of DBMS

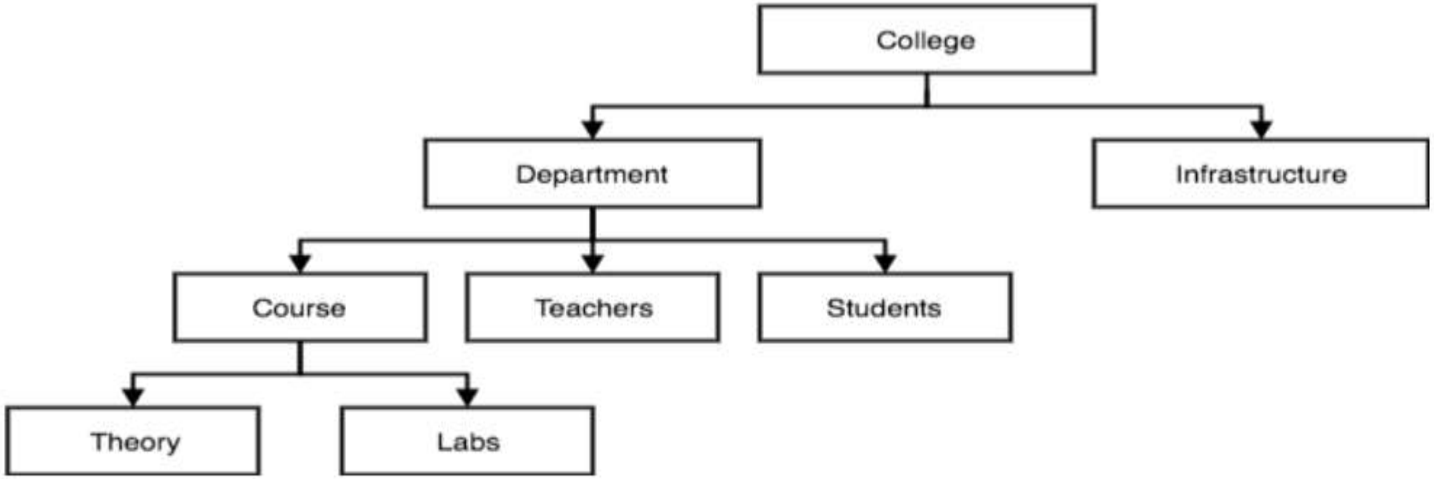
- The Main Classification Data Model
  - Relational Data Model
  - Object Data Model
  - Document(JSON), graph-based, column-based and key value data models
  - Legacy applications still work database based on the hierarchical and network(COBOL) data models
  - Experimental DBMS are based on a tree structured data model(XML)

Classification of DBMS

- Number of users:
  - Single user
  - Multi user
- Number of sites:
  - Centralized
  - Distributed(big data) homogenous vs heterogenous
- Cost:
  - Free
  - Commercial
- Purpose- types of access path:
  - General Purpose
  - Special Purpose(online transaction processing(OLTP) system

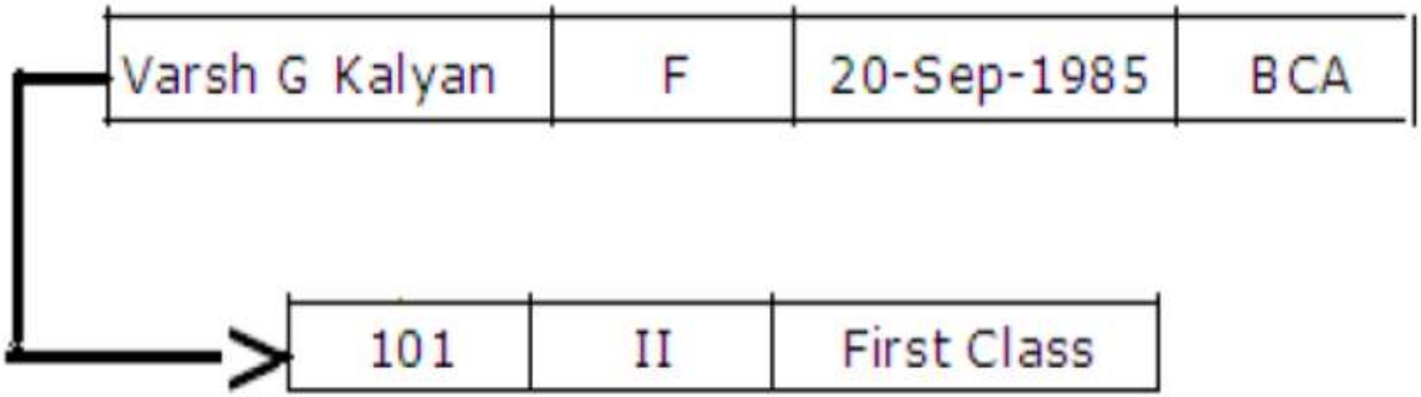
Hierarchical Model

- A hierarchical data model is a data model which the data is organized into a tree like structure.
- The structure allows repeating information using parent/child relationships: each parent can have many children, but each child has one parent. All attributes of a specific record are listed under an entity type



Network Model

- The data in the network model are represented by collection of records and relationships among data are represented by links, which can be viewed as pointers

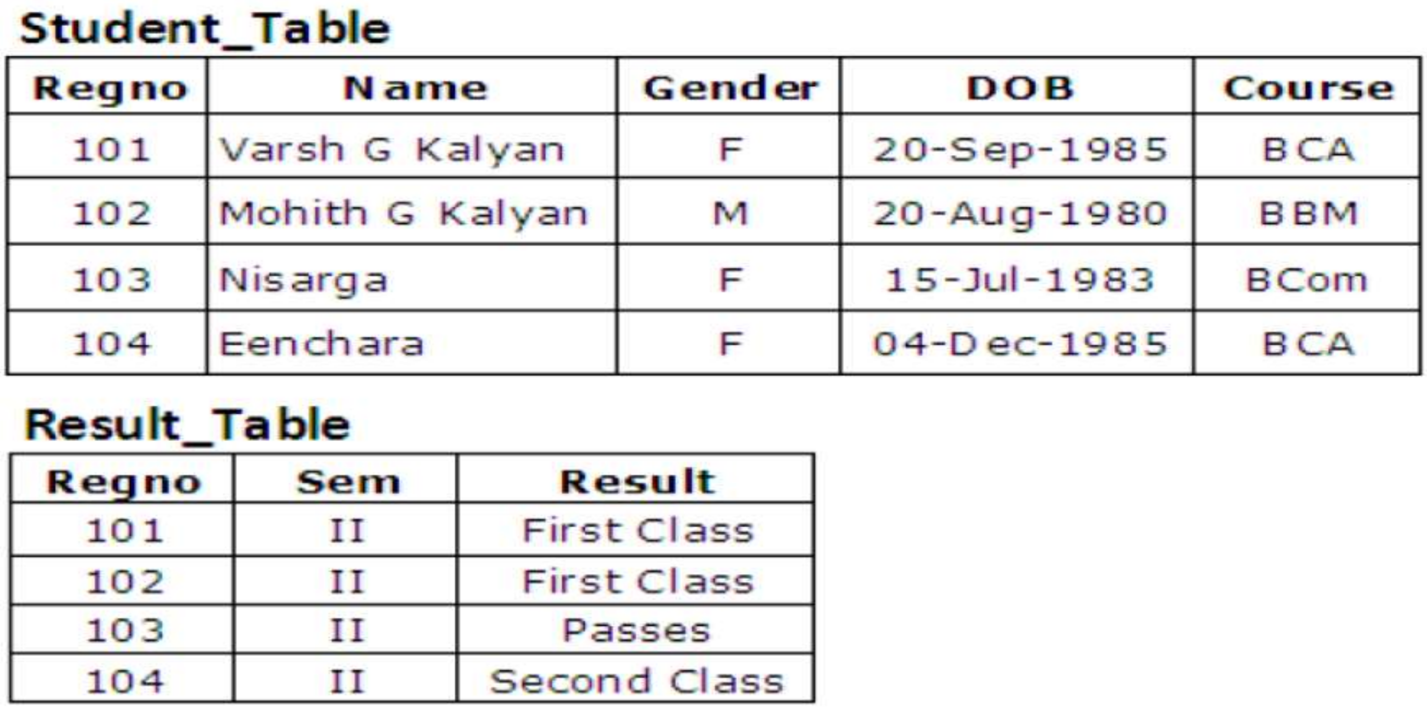


Varsh G Kalyan	F	20-Sep-1985	BCA

101	II	First Class

Relational Model

- Relational Model uses a collection of tables both data and the relationship among those data. Each table has multiple columns, and each column has a unique name.



Student_Table

Regno	Name	Gender	DOB	Course
101	Varsh G Kalyan	F	20-Sep-1985	BCA
102	Mohith G Kalyan	M	20-Aug-1980	BBM
103	Nisarga	F	15-Jul-1983	BCom
104	Eenchara	F	04-Dec-1985	BCA

Result_Table

Regno	Sem	Result
101	II	First Class
102	II	First Class
103	II	Passes
104	II	Second Class