Module 1

Introduction to Databases

Introduction

- A **database** is a collection of related data with an implicit meaning. By data, we mean known facts that can be recorded and that have implicit meaning.
- *Examples: University database, Banking database, Library database, Company database, Movie database, etc.*

A database has the following implicit properties:

- It represents some aspect of the real world, sometimes called the **mini world** .
- It is a logically coherent collection of data to which some meaning can be attached.
- It is designed, built, and populated with data for a specific purpose .

- **Database Management System (DBMS)** is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.
- **Defining** a database involves specifying the data types, structures, and constraints of the data to be stored in the database.
- **Constructing** the database is the process of storing the data on some storage medium that is controlled by the DBMS.
- **Manipulating** a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world, and generating reports from the data.
- **Sharing** a database allows multiple users and programs to access the database simultaneously.

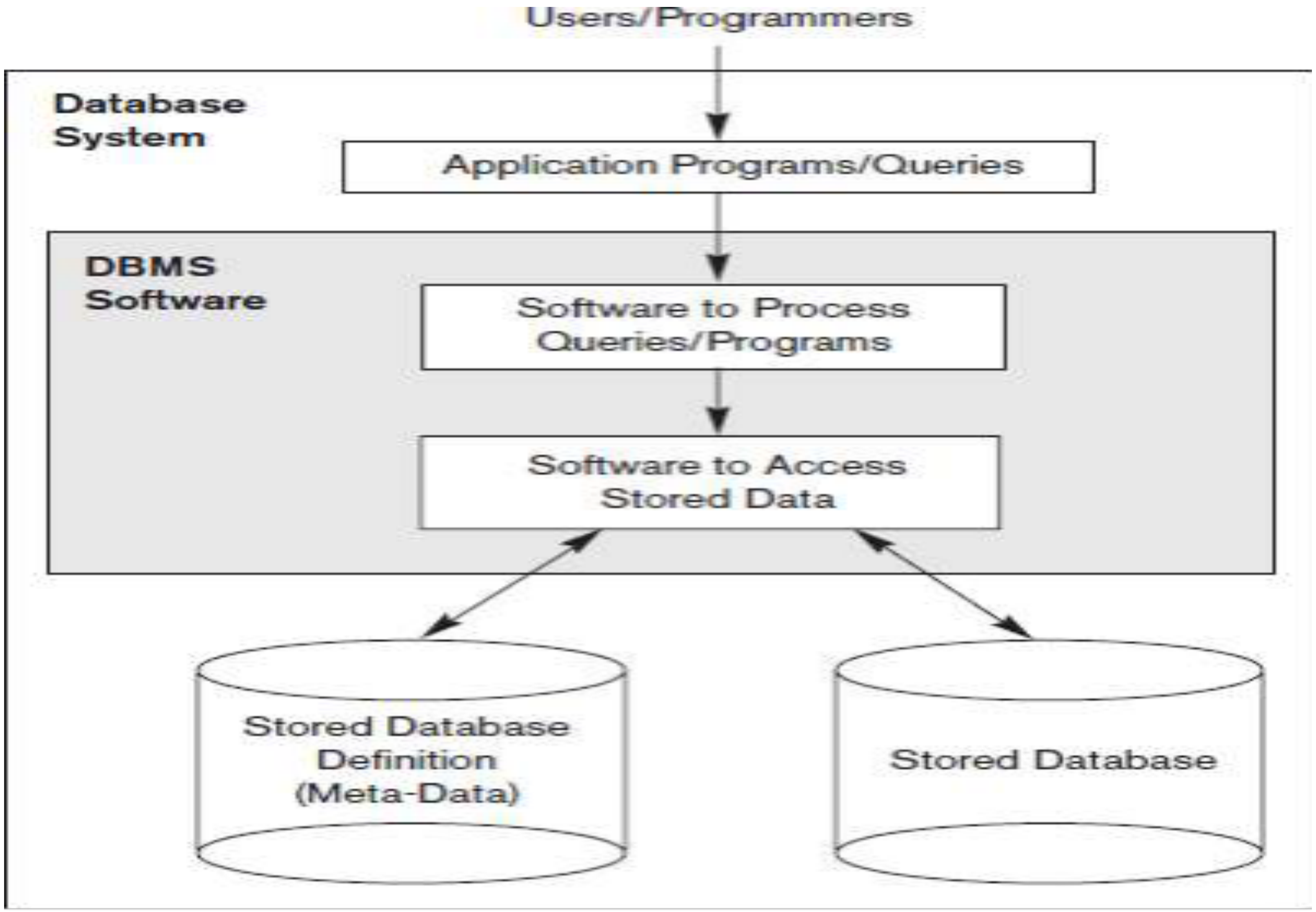

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Fig. A simplified database system environment

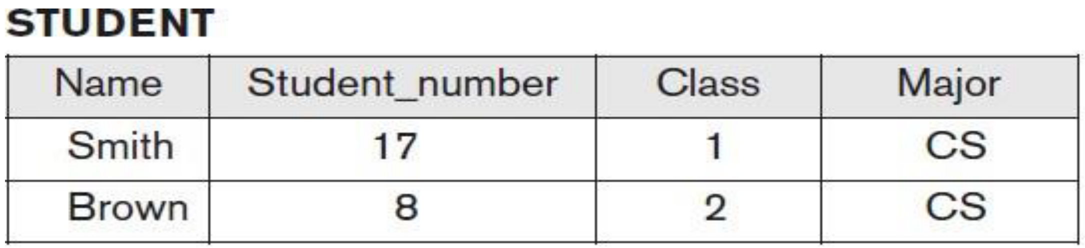
- An **Application program** access the database by sending queries or request for data to the DBMS.
- A query typically causes some data to be retrieved.
- Other important functions provided by the DBMS include **protecting** the database and **maintaining** it over a long period of time.
- **Protection** includes system protection against hardware or software malfunction (or crashes) and security protection against unauthorized access.
- A typical large database may have a life cycle of many years, so the DBMS must be able to maintain the database system by allowing the system to evolve as requirements change over time.
- A database together with the DBMS software is referred to as a **database system.**

An Example

Consider a UNIVERSITY database for maintaining information concerning students, courses, and grades in a university environment. The database is organized as five files, each of which stores data records of the same type.

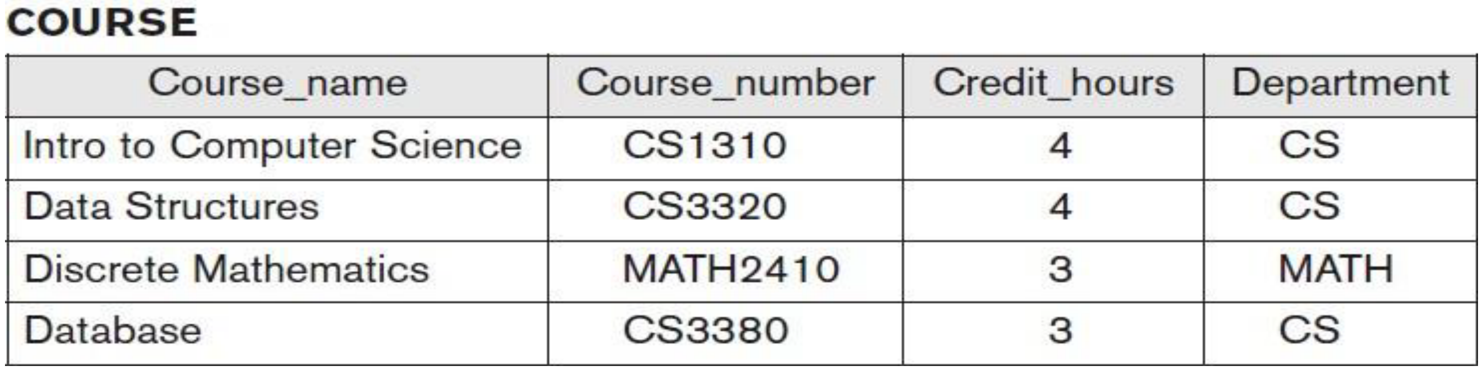
1. STUDENT file: stores data on each student.

2. COURSE file: stores data on each course.



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS



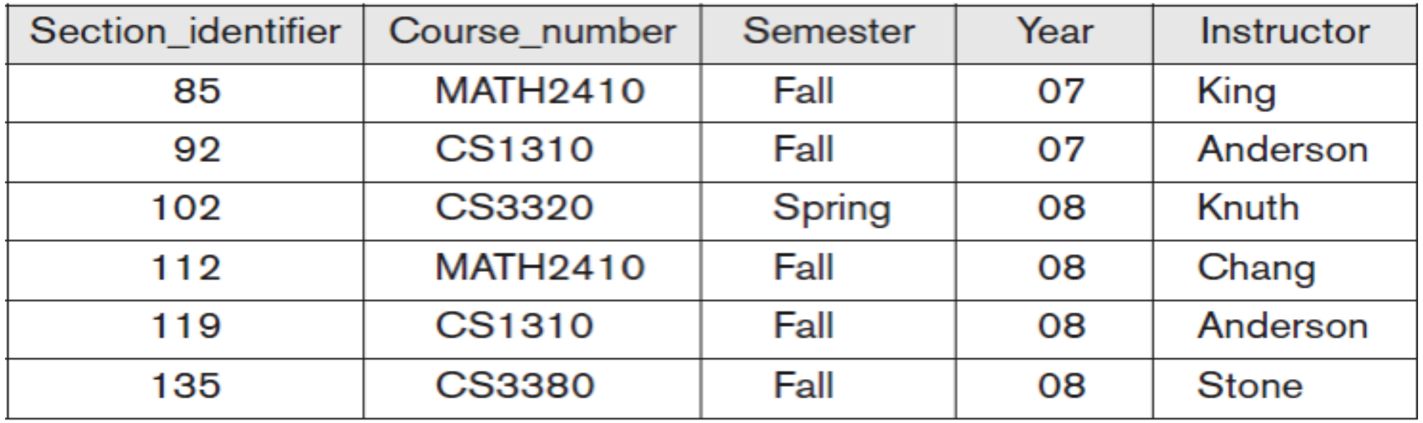
COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

3. SECTION file: stores data on each section of a course.

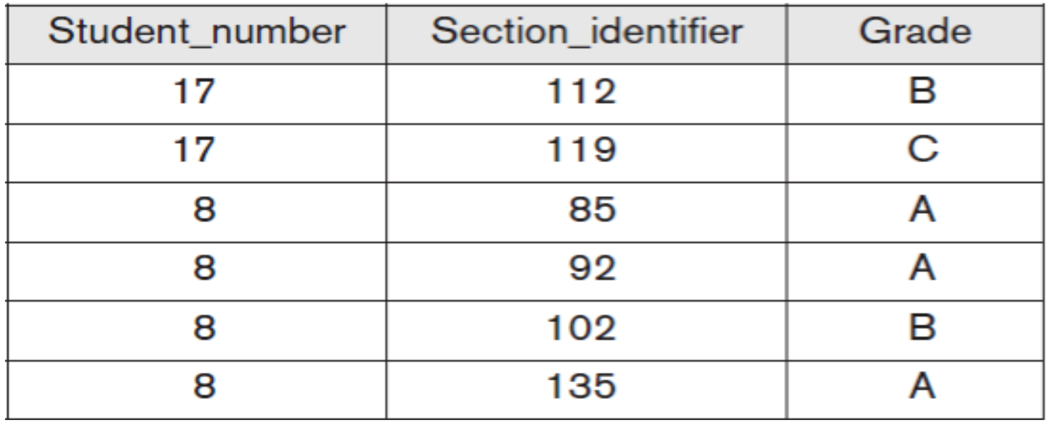
4. GRADE_REPORT file: stores the grades that students receive in the various sections they have completed.

SECTION



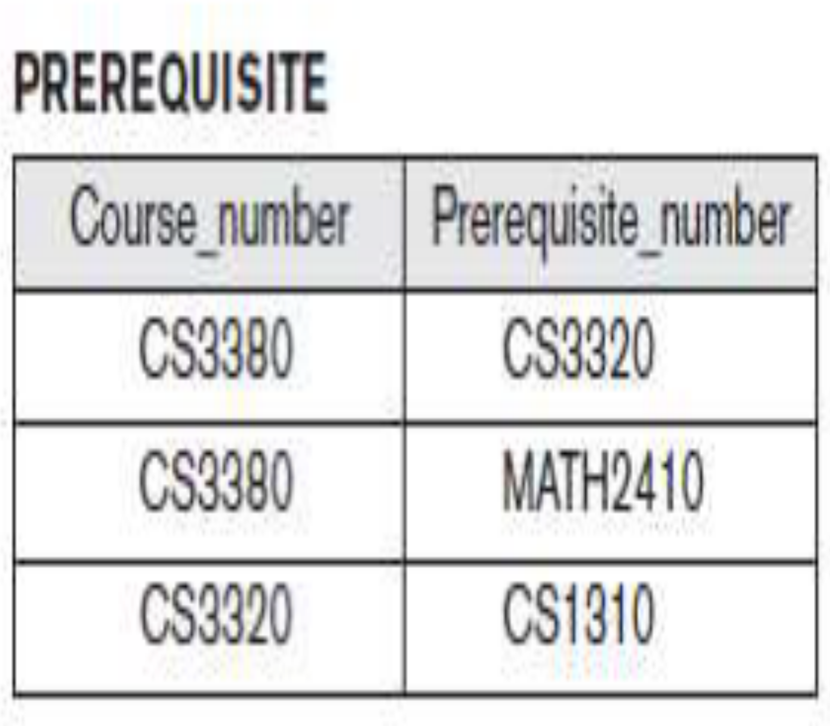
Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT



Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

5. PREREQUISITE file :stores the prerequisites of each course.



PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Defining a UNIVERSITY database

Specify the structure of the records of each file - **data elements** to be stored in each record.

For example: each STUDENT record includes data to represent the Student's Name, Student_number, Class and Major.

Specify a data type for each data element within a record.

For example: is a string of alphabetic characters Student_number is an integer.

Constructing the UNIVERSITY database

- To construct the UNIVERSITY database, we store data to represent each student , course, section, grade report, and prerequisite as a record in the appropriate file.

- Records in the various files may be related.

For example, the record for Smith in the Student GRADE_REPORT File is related to two records in the GRADE_REPORT file that specify Smith's grades in two sections

Manipulating a UNIVERSITY database

Database manipulation involves querying and updating.

- Examples of queries are as follows:
    - Retrieve the transcript a list of all courses and grades of Smith.
- Examples of updates include the following:
    - Change the class of ‘Smith to sophomore.

Characteristics of the Database Approach

- Self-describing nature of a database system
- Insulation between programs and data, and data abstraction
- Support of multiple views of the data
- Sharing of data and multiuser transaction processing

1. **Self-describing nature of a database system** means that the database system not only contains the database but also the information about the structure and constraints of the database.
    - This information about database structure and constraints of the database is know as Meta data.
    - The Meta data is stored in the catalogue of the database.

2. Insulation between programs and data, and data abstraction

- In a **file system** if some changes are made in the file structure, then to handle these changes more changes have to be made in all the programs that access this file.
- For example: You want to add a piece of data,

date of birth of student,

just adding it to the database is not enough the whole program will have to be re-written to make it work.

- But in a **database system** all you need to do is define another data item in the catalogue called date of birth and all changes will be reflected and there is no need to change the whole program.

(Catalogue in DBMS is a collection of tables and views that describe the structure of the database)

(Form adding data into the table we use INSERT sql command)

(VIEW in SQL are a type of virtual table that simplifies how users interact with data across one or more tables)

3. Support of multiple views of the data

- A database has multiple users each user may need different view of the database.
- Database view can be the information that is stored in the database or is derived from the database.
- Database system allows multiple such ‘views’ from the same database without letting the user know whether the information is stored just derived.

4. Sharing of data and multiuser transaction processing

- A Multiuser DBMS as the name suggest means has multiple users.
- It must allow multiple users to access and use the database at the same time
- This is a must when multiple application use a single database to store and integrate data in one single database.
- Multiple user can not modify same data in same time this achieve concurrency control.
- Concurrency Control Software make sure that the information is correct when multiple users try to access and update data.
- Such application are known as **online Transaction Processing (OLTP) Applications.**

Database Users

- Users may be divided into
- Those who actually use and control the database content, and those who design, develop and maintain database applications called **Actors on the Scene.**
- Those who design and develop the DBMS software and related tools, and the computer systems operators called **Workers behind the Scene.**

Actors on the Scene

In large organizations, many people are involved in the design, use, and maintenance of a large database with hundreds or thousands of users.

Identify the people whose jobs involve day to day use of a large dataset.

- Database Administrator (DBA)
- Database Designers
- End Users
- System Analysts and Application Programmers (Software Engineers)

Database Administrator

- In Database Environment
    - Primary resource – Database
    - Secondary resource – DBMS and related software.
- Database Administrator (DBA) responsibilities:
    - Administering primary/secondary resources
    - Authorizing access to the database
- Coordinating and monitoring use of database
- Acquiring , Hardware and software resources as needed.
- To implement new features and trouble shoot issues.

Database Designers

- Responsible for:
    - Identifying the data to be stored in the database
- Choosing appropriate structures to represent and store data.
- Communicating with database users
    - understand their requirements
    - design database.

3) End users are users are people who jobs require access to the database for querying, updates and generate reports.

- **Casual end user**: Occasionally access the Database but they need different information each time
- They use a sophisticated database query language (based on the requirement they develop the database application) to specify their requests.

(Canned transaction in DBMS is a pre-defined set of actions that can be used to perform routine tasks like Frequently updating the database).

- **Naive or parametric end users:** Constantly querying and updating the database using standard type of queries and update called canned transactions.
- **Sophisticated end users:** Including engineers, scientists, business analysts and other who thoroughly familiarise with the facilities of the DBMS in order to implement their application to meet their complex requirements.

- **Stand alone end users:** maintain personal database by using readymade programming packages that provide easy-to-use.

Ex: Canva

4) System analysts application program

- System analysts determine the requirements of end users develop the specifications for canned transaction.
- Application programmers implement these specification as program then they test, debug, maintain these canned transaction.

Workers behind the Scene

- Worker Behind the scene are the people whose work is to maintain the database system environment.
- These people maintain the DBMS environment that is they deal with the design development and operation of the database system and they are not interested in the database itself.

1. System designers and implementers
2. Tool developers
3. Operators and maintenance personnel

1) **System designers and implementers** *design and implement the DBMS modules and interfaces as a software package*.

- DBMS is very complex software system that consist of modules such as implementing catalog, processing query languages, Processing interfaces and accessing data controlling concurrency to implement all these modules as a one software package.

2) Tool developers *design and implement tools* of the software packages that facilitate database modelling and design, database system design, and improved performance.

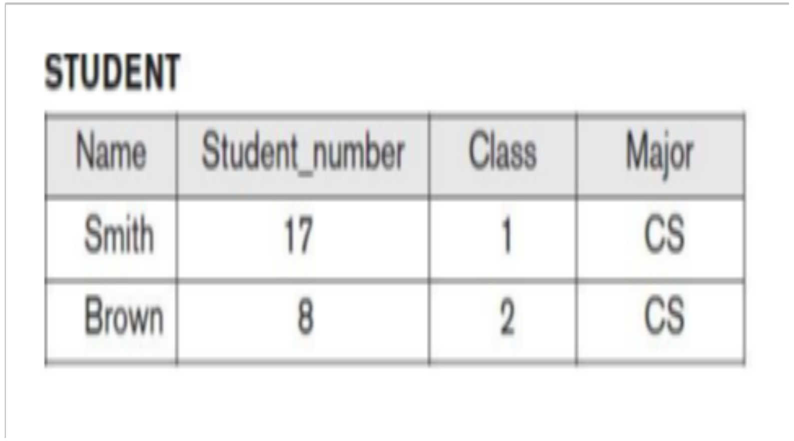
3) Operators and maintenance personnel (system administration personnel) are responsible for the actual *running and maintenance of the hardware and software environment* for the database system.

Advantages of Using the DBMS Approach

1. Controlling Redundancy
2. Restricting Unauthorized Access
3. Providing Persistent Storage for Program Objects:
   (Persistent Storage: non-volatile Storage)
4. Providing Storage Structures and Search Techniques for Efficient Query Processing.
5. Providing Backup and Recovery.
6. Providing Multiple User Interfaces

7. Representing Complex Relationships among Data

A database may include numerous varieties of data that are interrelated in many ways.



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

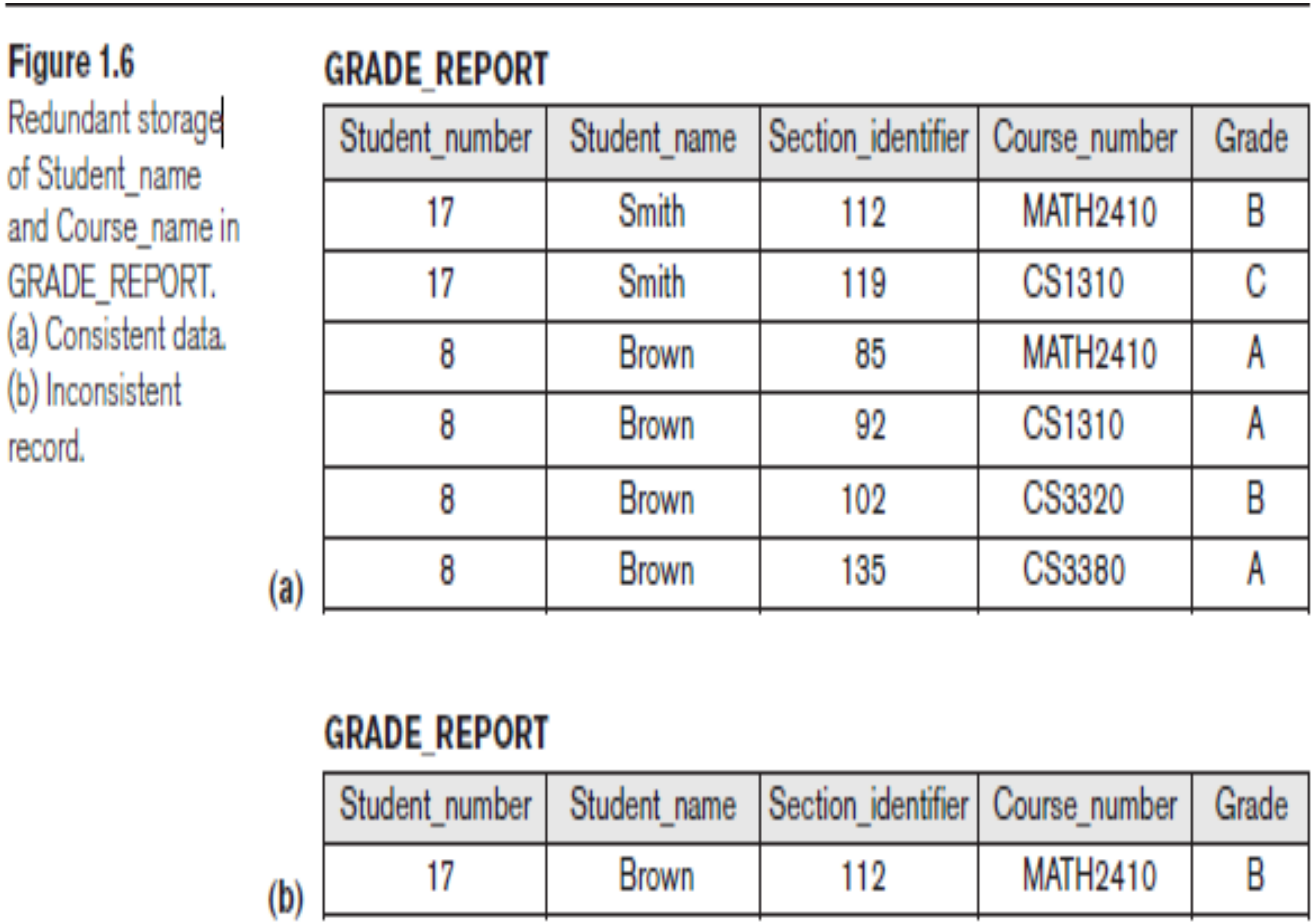


Figure 1.6
Redundant storage of Student_name and Course_name in GRADE_REPORT.
(a) Consistent data.
(b) Inconsistent record.

GRADE_REPORT

(a)

Student_number	Student_name	Section_identifier	Course_number	Grade
17	Smith	112	MATH2410	B
17	Smith	119	CS1310	C
8	Brown	85	MATH2410	A
8	Brown	92	CS1310	A
8	Brown	102	CS3320	B
8	Brown	135	CS3380	A

GRADE_REPORT

(b)

Student_number	Student_name	Section_identifier	Course_number	Grade
17	Brown	112	MATH2410	B

8. **Enforcing Integrity Constraints:** (Integrity Constraints are rules that ensure the accuracy, consistency and validity of data in a database.)

9. **Permitting Inferencing and Actions Using Rules:**

- Permitting Inferencing and Actions Using Rules is a feature of deductive database system.
- deductive database system is database system that combines logic programming languages with data models to analyze and query database data.(data models is a frameworks that defines how data is organized, connected and interpreted)

10. Additional Implications of Using the Database Approach

- Potential for Enforcing Standards: used by large organization for communication, cooperation among various department and projects.
- Reduced Application Development Time
- Availability of Up-to-Date Information
- Economies of Scale: The DBMS approach permits consolidation of data and application thus reducing the amount of wasteful overlap between activities of data processing personnel in different projects

History of Database Applications

1. Early Database Applications Using Hierarchical and Network Systems

Early database applications-maintained records in large organizations such as corporations, universities, hospitals, and banks

Problems with the early database systems

- lack of data abstraction and program-data independence capabilities
- provided only programming language interfaces. This made it time-consuming and expensive to implement new queries and transactions, since new programs had to be written, tested, and debugged.

2. **Providing Data Abstraction and Application Flexibility with Relational Databases:**Relational databases were originally proposed to separate the physical storage of data from its conceptual representation and to provide a mathematical foundation for data representation and querying.

3. **Object-Oriented Applications and the Need for More Complex Databases**: Object-oriented databases (OODBs) mainly used in specialized applications, such as engineering design, multimedia publishing, and manufacturing systems. In addition, many object-oriented concepts were incorporated into the newer versions of relational DBMSs, leading to object-relational database management systems, known as ORDBMSs.

4. Interchanging Data on the Web for E-Commerce Using XML

- The World Wide Web provides a large network of interconnected computers. Users can create documents using a Web publishing language, such as Hypertext Markup Language (HTML), and store these documents on Web servers where other users (clients) can access them. Documents can be linked through hyperlinks, which are pointers to other documents..

5. Extending Database Capabilities for New Applications

- The success of database systems in traditional applications encouraged developers of other types of applications to attempt to use them. The following are some examples of these applications:
- Scientific applications that store large amounts of data resulting from Scientific experiments
- Storage and retrieval of videos, such as movies, and video clips from news or personal digital cameras.
- Data mining applications that analyze large amounts of data searching for the occurrences of specific patterns, and for identifying unusual patterns (credit card).

Databases versus Information Retrieval

- Database technology is heavily used in manufacturing, retail, banking, insurance, finance, and health care industries, where structured data is collected through forms, such as invoices or patient registration documents.

When Not to Use a DBMS

- DBMS may involve unnecessary overhead costs that would not be incurred in traditional file processing. The overhead costs of using a DBMS are due to the following:
  - High initial investment in hardware, software, and training
  - The generality that a DBMS provides for defining and processing data
  - Overhead for providing security, concurrency control, recovery, and integrity functions

Therefore, it may be more desirable to use regular files under the following circumstances:

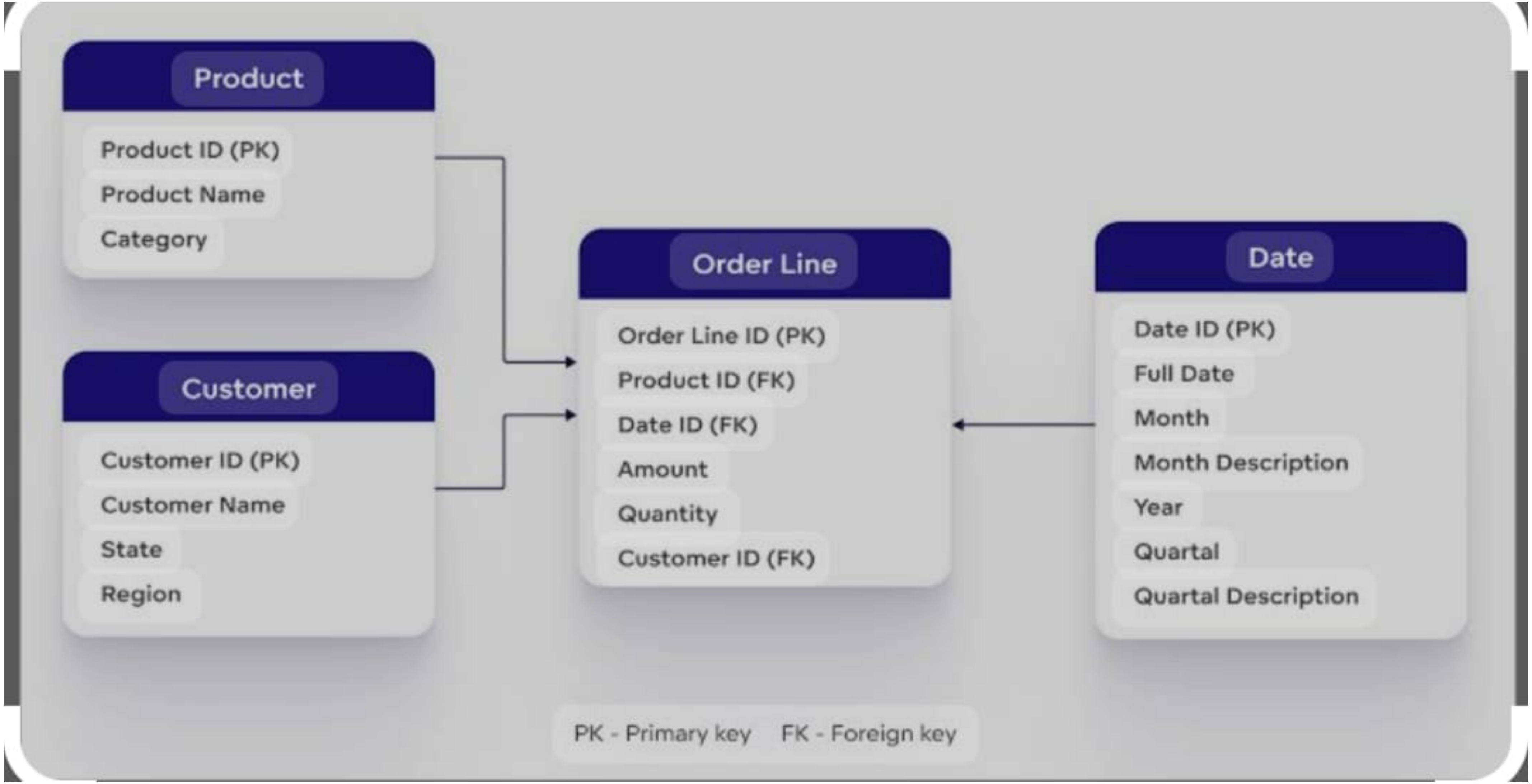
- Simple, well-defined database applications that are not expected to change at all
- Stringent, real-time requirements for some application programs that may not be met because of DBMS overhead
- Embedded systems with limited storage capacity, where a general-purpose DBMS would not fit
- No multiple-user access to data

Chapter 2:

Overview of Database Languages and Architectures

Data Model

- Data modeling in DBMS is the process of representing data in a database management system (DBMS).
- It helps to organize data, identify relationships, and ensure data security.
- It is a framework that defines how data is organized and connected.
- Its like architecture of a building that ensures data is organized.
- A data model is a collection of concepts that can be used to describe the structure of a database.
- By structure of a database, we mean the data types, relationships and constraints that apply to the data.



Product

Product ID (PK)
Product Name
Category

Customer

Customer ID (PK)
Customer Name
State
Region

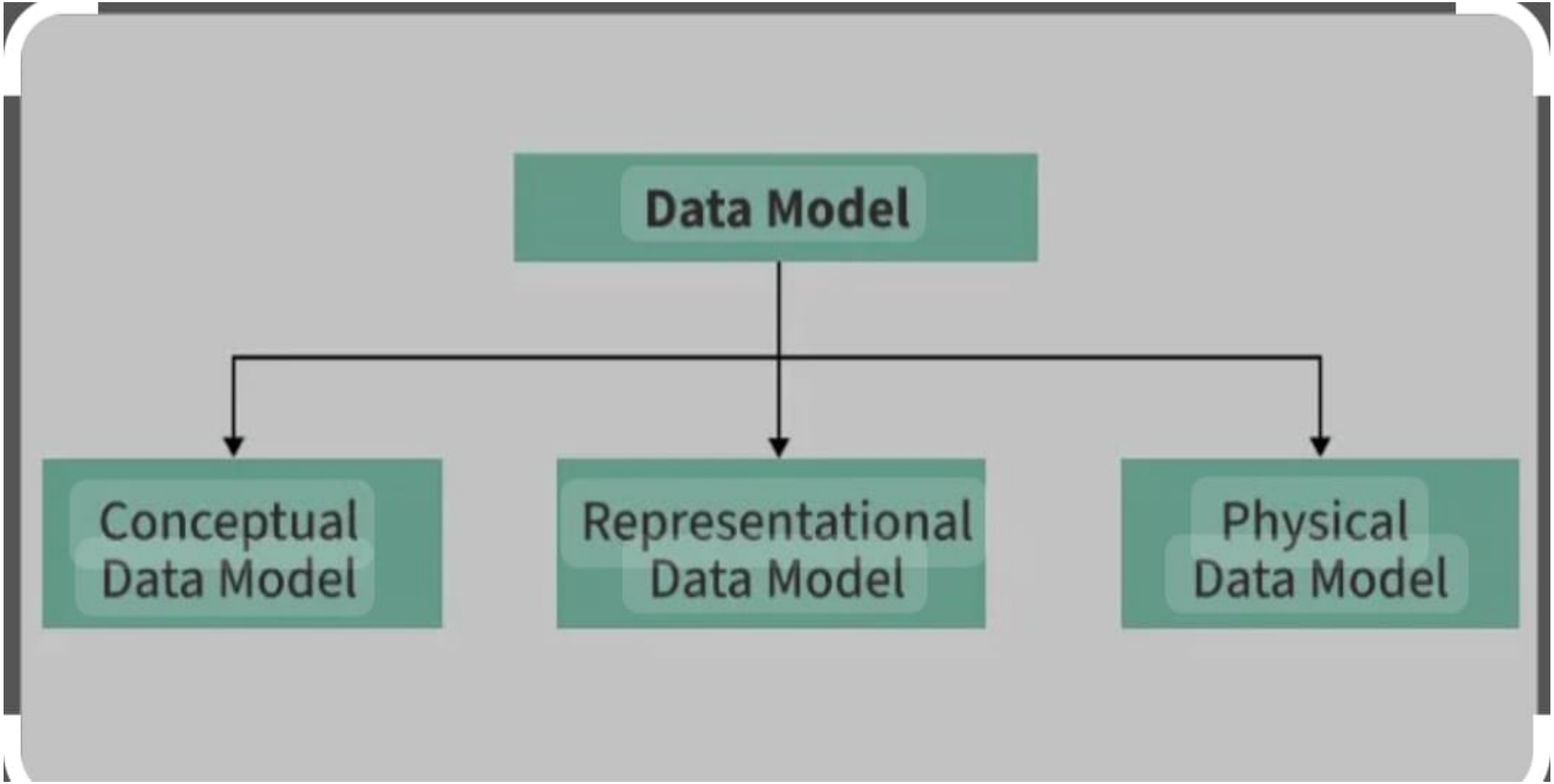
Order Line

Order Line ID (PK)
Product ID (FK)
Date ID (FK)
Amount
Quantity
Customer ID (FK)

Date

Date ID (PK)
Full Date
Month
Month Description
Year
Quartal
Quartal Description

PK - Primary key FK - Foreign key


Data Model
Conceptual Data Model
Representational Data Model
Physical Data Model

Conceptual Data Model: To represent the high-level structure of the data without worrying about how it will be physically implemented.

It focuses on **what** data is important and the relationships between different data entities.

Representational Data Model (Logical Data Model): It serves as a bridge between the conceptual model and the physical model, providing a more detailed structure of the data.

It outlines how data will be represented and organized in the database without specifying physical storage details.

Physical data Model: This model describes how the data will be physically stored in the database. It is concerned with performance, optimization, and efficient storage, considering the specific database system and hardware in use.

Categories of Data Models

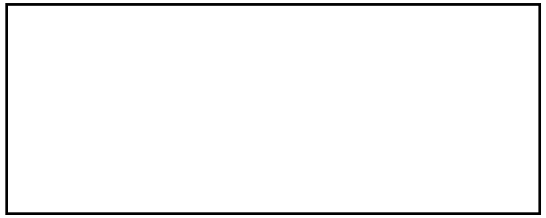
1. **High-level or conceptual data models:** The Conceptual data model describes the database at a very high level and is useful to understand the **needs or requirement** of the database.

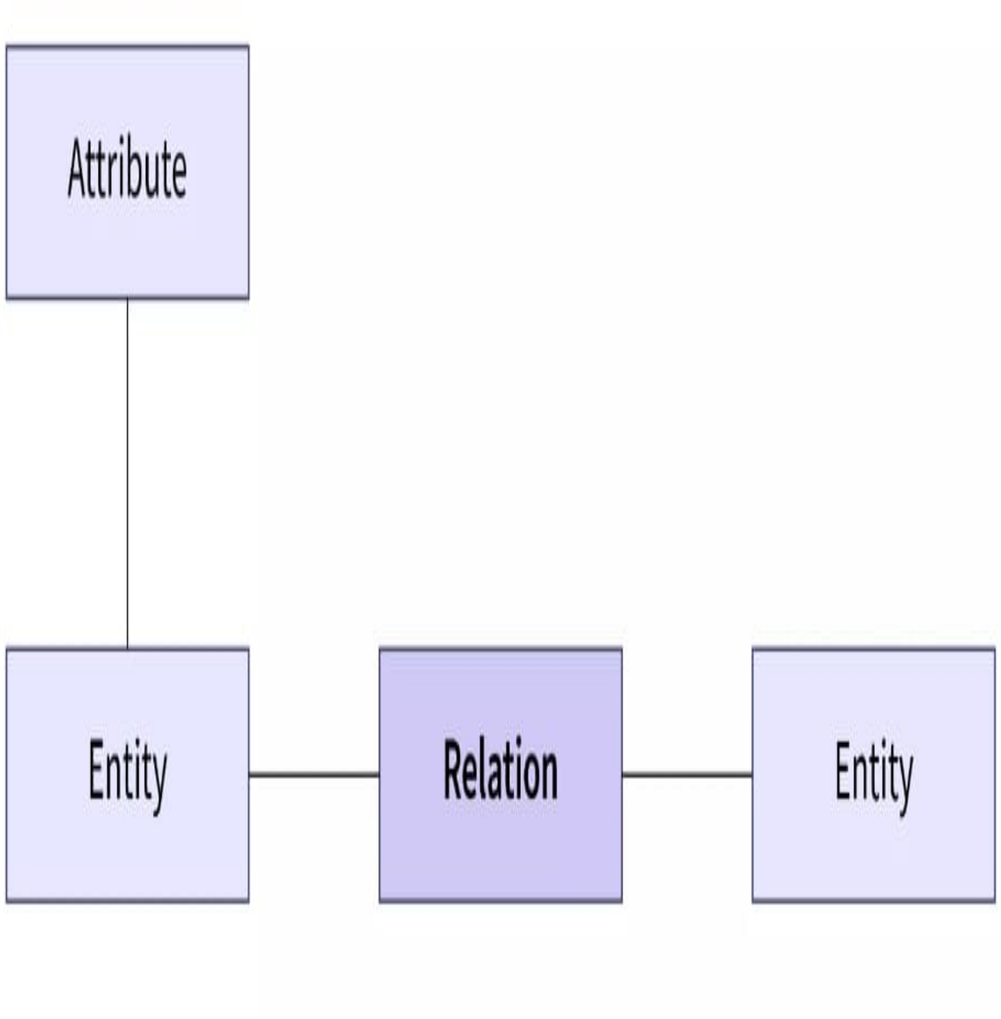
- It is used in the requirement gathering process.
- Before the database Designer start making a particular database, Conceptual data models use concepts such as entities, attributes, and relationships.

A **Data Entity** is a structured framework designed to simplify access and management of data within an application or across multiple systems

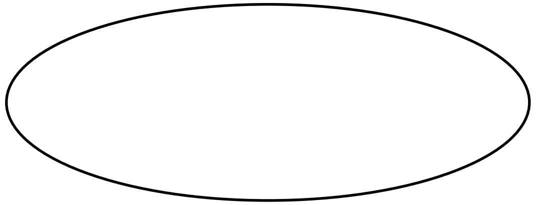
An Data Entity is referred as a real world object .

This are represented by rectangle in ER Diagram

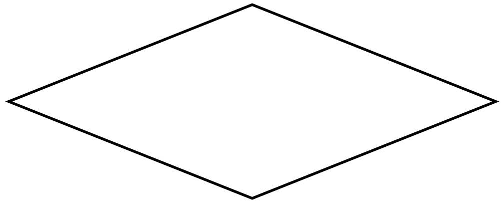


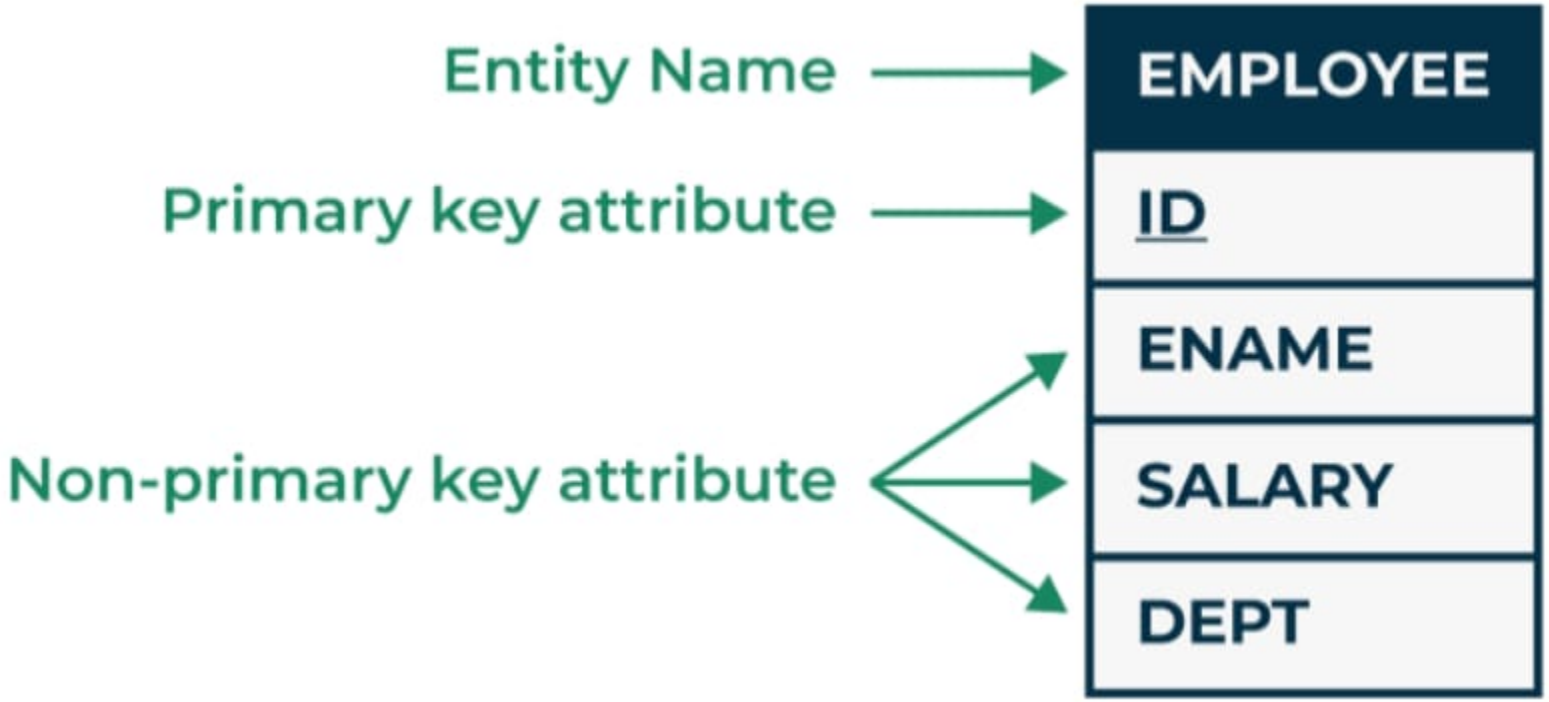


- **Attributes** can be defined as the description of an entity
- attributes are the characteristics that define entities.
- If name is an entity, Vinutha is an attribute



- **Relationship** are used to define relation among different entities.
- Relationships help organize entities and work with data.
- For example, in a university database, relationships could connect students to courses they can take.





Entity Name

Primary key attribute

Non-primary key attribute

EMPLOYEE

ID

ENAME

SALARY

DEPT

2. Representational or implementation data models:

Representational data model is used to represent only the logical part of the data model and does not represent the physical structure of the database.

3. Low-level or physical data models:

provide concepts that describe the details of how data is stored on the computer storage media, typically magnetic disks.

Physical data models describe how data is stored as files in the computer by representing information such as record formats, record orderings, and access paths.

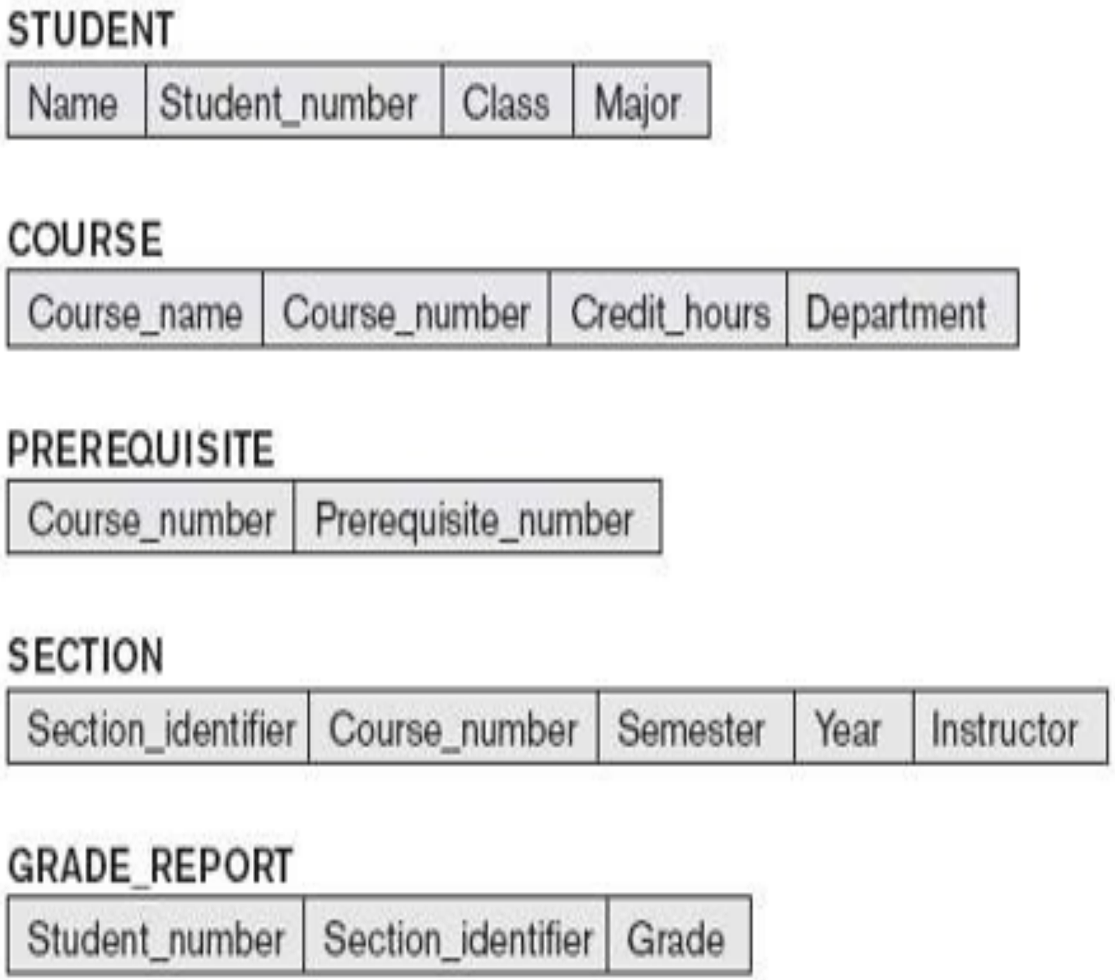
(**Physical storage**: A 500GB hard drive with specific sectors where data is physically stored. **Logical storage:** A "Documents" folder on your computer which appears as a single location to the user, even though the data is spread across different physical sectors on the hard drive.)

Database schema

- The description of a database is called the **database schema**, which is specified during database design and is not expected to change frequently.
- An **"entity**" represents a real-world object or concept, like a customer or product, while a **"schema"** defines the overall structure of the database, including the tables, columns, data types, and relationships between entities, essentially acting as a blueprint for how data is organized within the database; an entity is a single building block within the schema.
- It is known as blueprint with the help of which we can explain that how data is organized into tables.
- The data in the database is called **instances**.

Schema diagram

- A displayed schema is called a schema diagram. A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints.



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

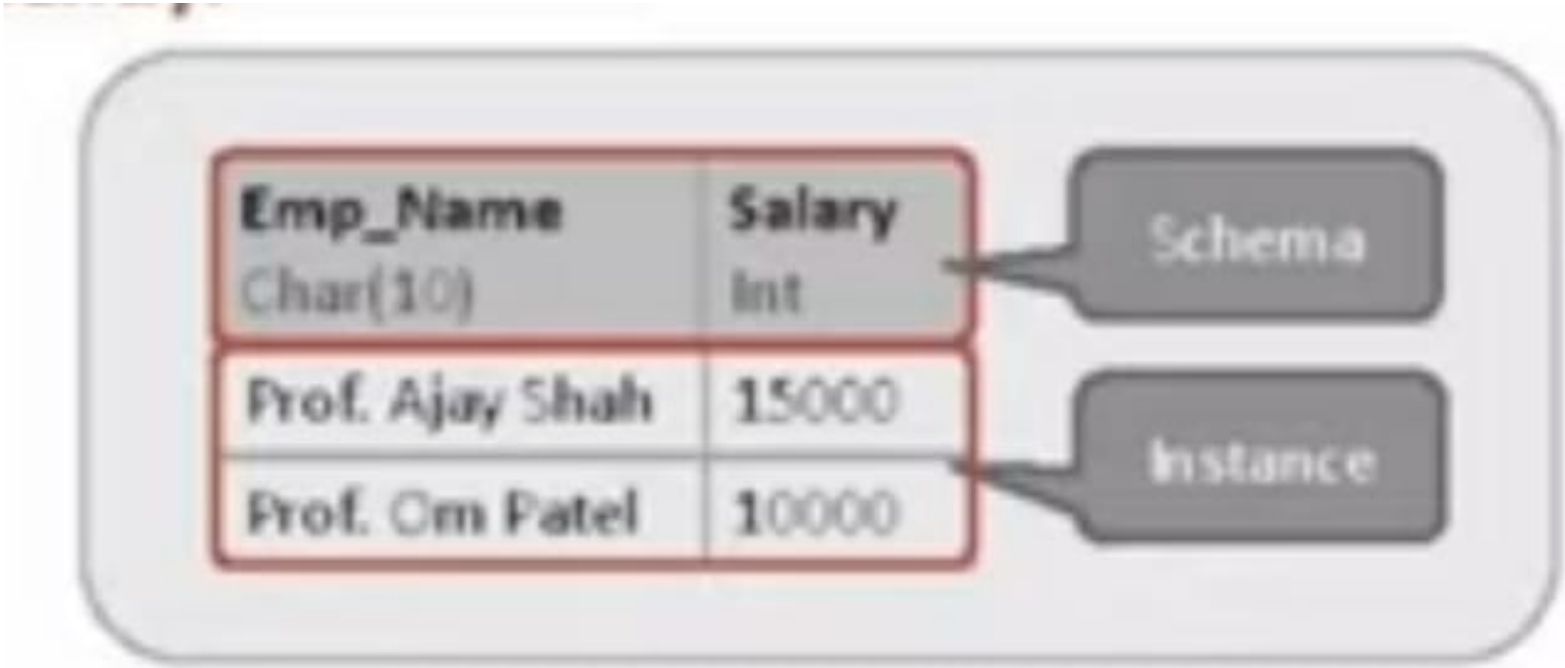
Figure 2.1: Schema diagram for the database

Instance

- The collection of information stored in the database at particular moment is called Instance.
- Instances are changed **frequently.**

Schema

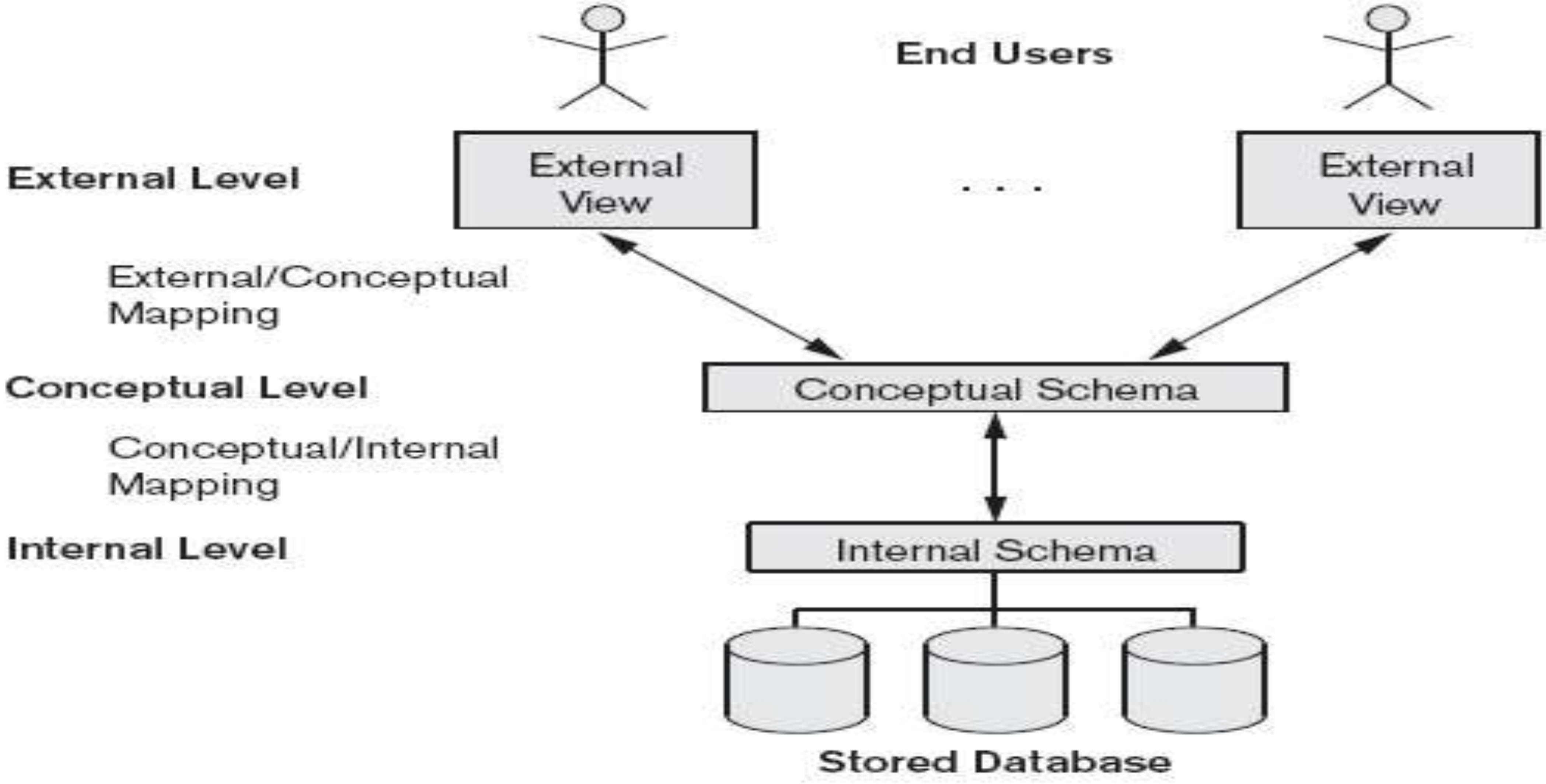
- The overall design of database is called database Schema.
- Schemas are changed **rarely.**



Emp_Name	Salary
Char(10)	Int
Prof. Ajay Shah	15000
Prof. Om Patel	10000

The Three-Schema Architecture

Goal: To separate the user applications and the physical database.



3 Level:

The external or view level includes a number of external schemas or user views 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.

- **The conceptual level has a conceptual schema** hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.
- **The internal level has an internal schema** Describes the physical storage of the database. Describes the complete details of data storage and access paths.

Data Independence

- The ability to change the schema at one level without changing the schema at another level.
- It refers that changes made in any level of database should not affect to its next high level.
- There are two types:

1) Physical data independence:

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.

2) Logical data Independence:

Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.

Database Languages or DBMS languages

- In order to perform operation on the database like CREATE, SELECT, INSERT, DELETE, UPDATE , database language is use.

- To perform Operations on database, SQL COMMENTS are use.

Types of database language

1) **Data Definition language(DDL) :**It contain commends which are required to define the database.

Defines database schema.(define conceptual Schema)

Define logical structure of database.

Ex: CREATE, ALTER, DROP, TRUNCATE.

2) Data Manipulation Language: It contains commands which are required to manipulate the data present in the database. Ex: SELECT, UPDATE, INSERT, DELETE.

Allow user to perform operation such as insertion, deletion, modification and retrieval.

3) Storage Definition Language: Is used to specify the internal storage Schema (Physical Storage)

4) View Definition Language: Specifies user views/ mapping to conceptual schema

5) Data Control Language: It contains command which are required to deal with user permission and control the database system. Ex: Grant, Revoke

6) Transaction Control Language: It contain commands which are required to deal with transaction of database. Ex: Commit, Rollback.

- CREATE DATABASE-Create a new database.
- CREATE TABLE-Create a new table.
- ALTER TABLE-used to add, delete or modify columns in an existing table.

Alter command will perform the action on structure level and not on the data level.

- SELECT-Extract the data from database.
- UPDATE-update data in database.
- The UPDATE statement is used to modify the existing records in a table.
- INSERT- Inserts new data into the database
- GRANT command is used to give access to users.
- REVOKE withdraws the access.
- ROLLBACK-used to undo those transactions that aren't saved yet in the databased.
- (Transaction in DBMS is a series of tasks that are treated as a single unit of work)

DBMS Interfaces

- An interface is a program that allows users to query the DBMS without writing the code query language .
- An interface can be used to manipulate the database either for adding the data or deleting some data or updating some data or even for viewing the data present in the database.

Form based Interface

Form based interface displays a form to each user.

The user fills out all the details and submits the form, to make a new entry into the database.

Or the user fills out some details and then the system retrieves rest of the details from the database.

Form based interfaces are built for Naïve users who have to do only a limited numbers of operations.

Example

Example of form-based interface is a teacher adding the attendance of students in the system.

Or a student entering his roll number, class and branch in a form to retrieve his or her exam results.

Menu based user interface

- These interfaces present the user with lists of options (called menus) that lead the user through the formulation of a request.
- In Menus, the user don't need to memorize the specific commands and syntax of a query language; rather, the query is composed step-by step by picking options from a menu that is displayed by the system.
- Pull-down menus interfaces are mostly used in Web-based user interfaces.

Example

- Lets say you are searching for some item on a shopping website.
- First thing you do is go the categories menu and select the category in which the item falls
- .Then you select the brand from the menu of brands.
- Then you search through different models of that item in the menu.
- Then you might select the budget you have from the menu of budget range.

Graphical user interface

- 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.
- Most GUIs use a pointing device, such as a mouse, to select certain parts of the displayed schema diagram

Example of GUI

- Let's say you are watching this video on YouTube.
- If you press that subscribe button, either with a finger or a mouse, it will turn grey.
- There are forms used in YouTube too, like you search some topic on YouTube.
- There are Menus also used on YouTube, like YouTube shows you different videos on different channels, also it gives you menu to select videos of certain length or from certain time.
- And all this interface is visually graphics and changes according the action the user takes.
- Hence Graphical User Interface.

Natural language Interface

- Natural language interface has high-level own unique schema more like the high-level conceptual schema.
- It also has its own dictionary of important words.
- Basically, it generates a query based on the interpretation of important words in the input by the user and if the interpretation is successful, then it displays the result to the user.

Example

- Lets say, you want to find out which is the fastest car in the India, so you search for fastest car in India on a search engine like Google, which is a natural language interface.
- The Natural language interface then looks at the important words in the input by the user: Fastest, car, India.
- Now it will go into the database of cars and search for all the cars that are available in India. Then it will check the car with highest speed among the cars available in India.
- And finally, it will show the outcome of the search to the user.

Speech Input and output

- Nowadays it has become a common type of interface.
- The users queries the interface with speech, and gets an answer or response in speech if the interface understands the query and interpret it.

Example

- Most of you must have used either Siri on apple, or OK google, or Alexa or Cortana to ask some questions like,

"OK, google, find the value of square root of 729“

"Or Alexa, what is the capital of Nepal“

And these speech user interfaces interpret your speech input and processes the data from the database and after successful interpretation answers you back in speech.

- If the query is not understood, it will ask you to repeat the query.

Interface for Parametric users

- Parametric users are users that have a small set of operations that need to be performed and that too repeatedly.
- This makes it important to reduce the amount of keystrokes for each operations.
- So the programmers program such interface so that the operations can be performed with single key stroke.

Example

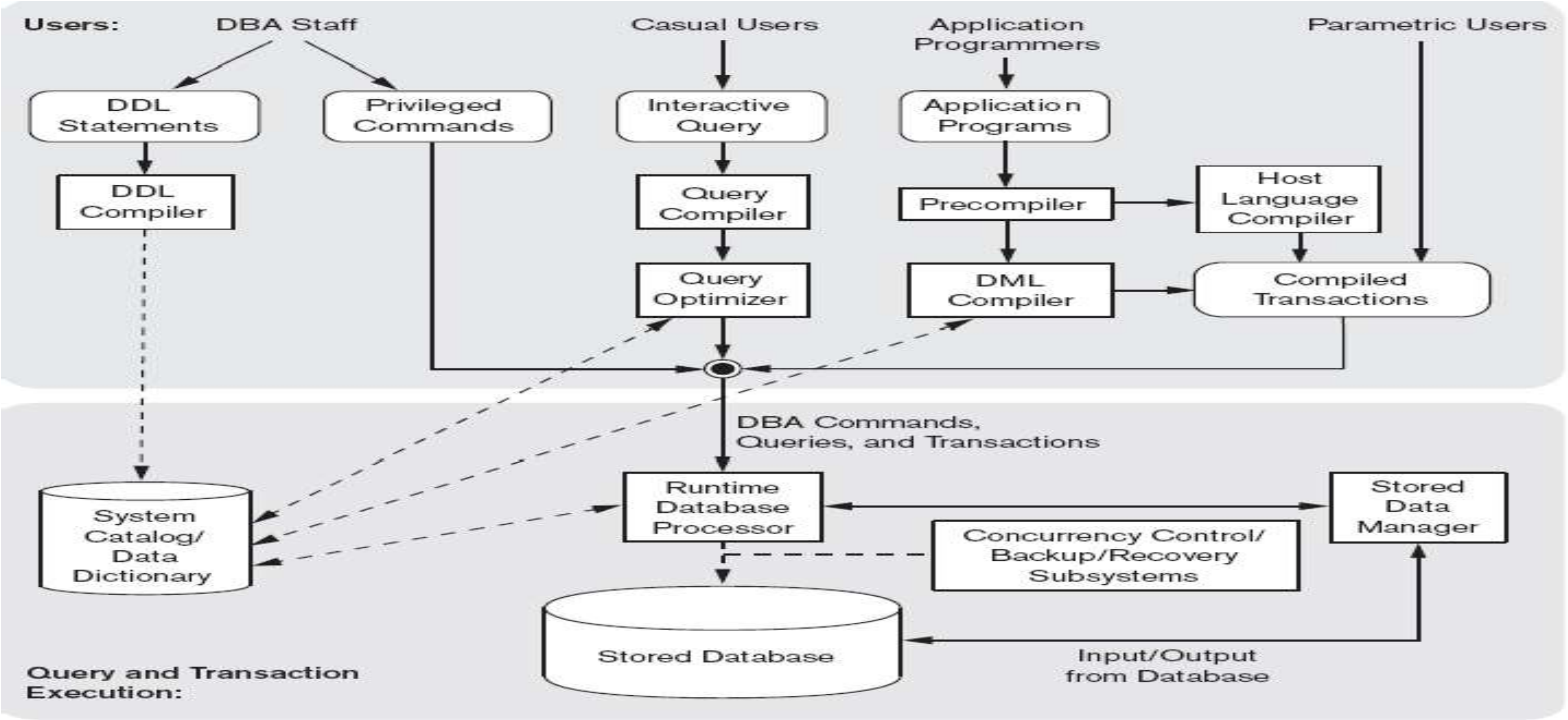
- Let's say there is an ATM. The ATM has limited operations to do like, check the balance, withdraw money or change pin of the card.
- So as you know, any of these operations which are complex programmatically but the queries are generated with minimal key strokes, as there are only three operations the ATM has to perform and nothing more.

Interface for 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

DBMS Component Modules



- This Diagram shows the different component has to be defined by database administrator and the database designer.

- The top part of the figure refers to the various users of the database environment and their interfaces. The lower part shows the internals of the DBMS responsible for storage of data and processing of transactions.

- Once the database Design is completed the schemas has to be defined by the database administrator and the database designer.

- So that DBA and its staff uses data definition language or DDL to specify these schemas.

DDL compiler-processes schema definitions that is specified in the DDL and stores descriptions of the schemas (meta-data) in the DBMS catalog.

Interactive query interface: interface for Casual users and persons with occasional need for information from the database.

Query compiler- validates for correctness of the query syntax, the names of files and data elements & compiles them into an internal form.

Query optimizer- concerned with the rearrangement and possible reordering of operations, elimination of redundancies and use of correct algorithms and indexes during execution.

- **Pre-compiler -** extracts DML commands from an application program and sends to the DML compiler for compilation into object code for database access.
- **Host language compiler -** rest of the program is sent to the host language compiler.

The object code and remaining program are linked together to form the canned transaction

- **Runtime database processor** executes the privileged commands the executable query plans, and the canned transactions with runtime parameters.

- **Stored data manager** Controls the access to the information that is stored on the disk and main memory. This module controls the data transfer between the disk and the main memory. Once the data is in main memory then it can be processed by another DBMS module.
- **Concurrency control and backup and recovery systems** integrated into the working of the runtime database processor for purposes of transaction management.

Database System Utilities

Database utilities help the DBA to manage the database system

Loading: used to load existing data files into the database.

Backup: creates a backup copy of the database, usually by dumping the entire database onto tape or other mass storage medium.

Incremental backups are also often used, where only changes since the previous backup are recorded. Incremental backup is more complex, but saves storagespace.

- **Database storage reorganization:** used to reorganize a set of database files into different file organizations, and create new access paths to improve performance.
- **Performance monitoring:** 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.

Tools

CASE (Computer aided software engineering) used in the design phase of database systems

Data dictionary (or data repository) : Used to store catalog information about schemas and constraints, the data dictionary stores other information, such as design decisions, usage standards, application program descriptions, and user information. Such a system is also called an information repository. This information can be accessed directly by users or the DBA when needed.

(Data repository is a centralized place to store manage and organize data)

Application development environments

- **PowerBuilder (Sybase) or JBuilder (Borland):** provide an environment for developing database applications including database design, GUI development, querying and updating, and application program development.
- **Communications software:** allow users to access the database through computer terminals, workstations, or personal computers.

(Computer terminal is a device that allows users to input data into and receive data from a computer)

Centralized and Client/Server Architectures for DBMSs

Centralized DBMSs Architecture

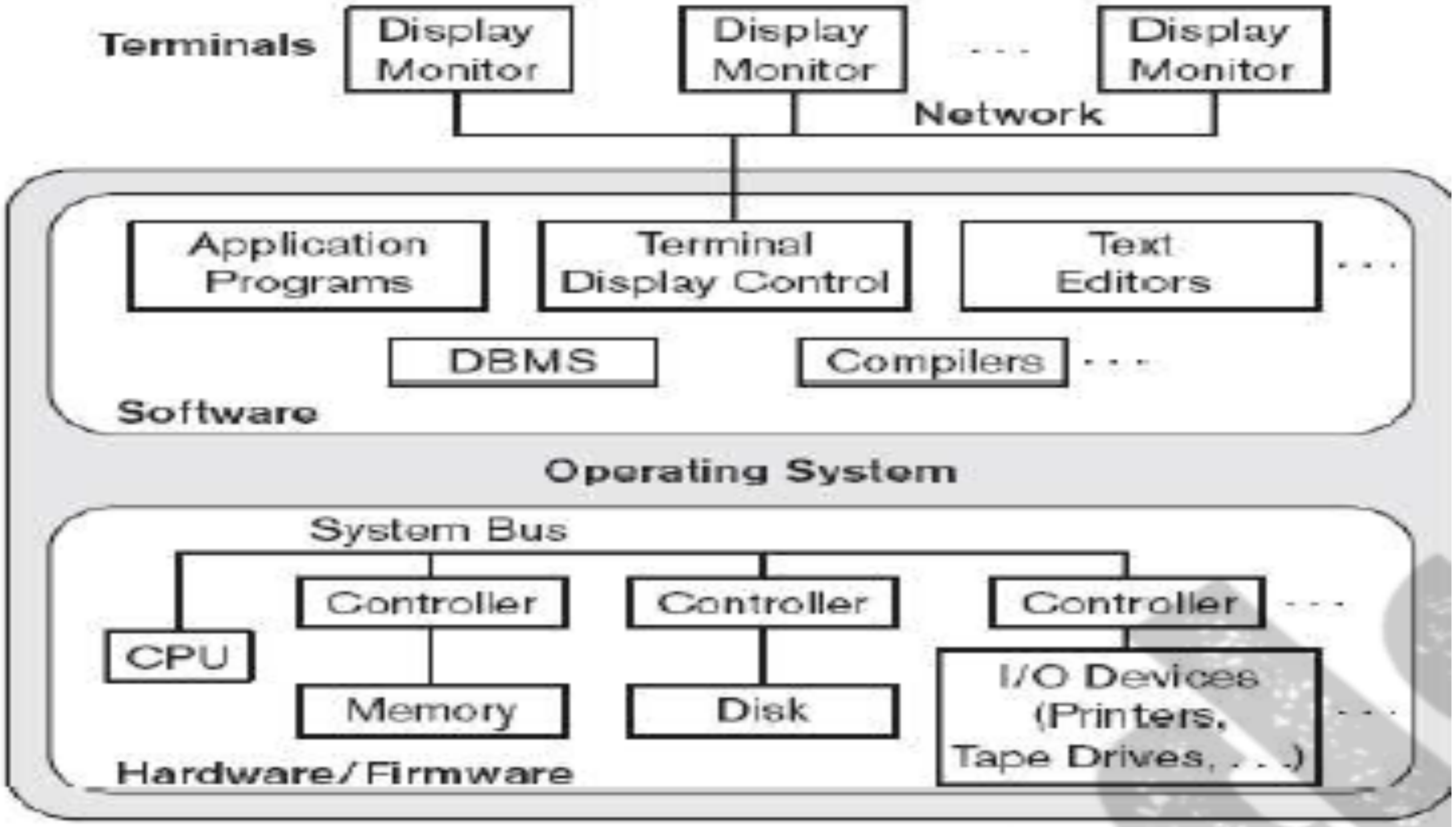


Figure 2.5.1: A physical centralized architecture

- In earlier architecture mainframe computers were used to process all system functions.
- Users accessed only computer terminals that provided only display capabilities but no processing capabilities.
- Computer terminals used by the users were not capable of processing but it was only capable of displaying any information to the user .
- Processing were performed on the computer system.
- Only displaying information was sent to the terminals and these terminals were connected to the computer system (network).
- As a prices of hardware started to reduce, the terminals were replaced by personal computer.

- Initially database system used these computers in a similar way as the display terminals.
- But the DBMS was centralized where all the functionalities of the DBMS user processing and execution of the application program all were done on one machine.
- (Computer terminal is a device made up of a display unit and a keyboard. It is used to access a computer system in order to run programs and manage computer files
- It is a hardware device used to interact with a computer system by sending commands and receiving outputs essentially acting as a user interface
- Computer system refer to the entire collection of hardware and software components that work together to process data which are accessed through a terminal)

Basic Client/Server Architectures

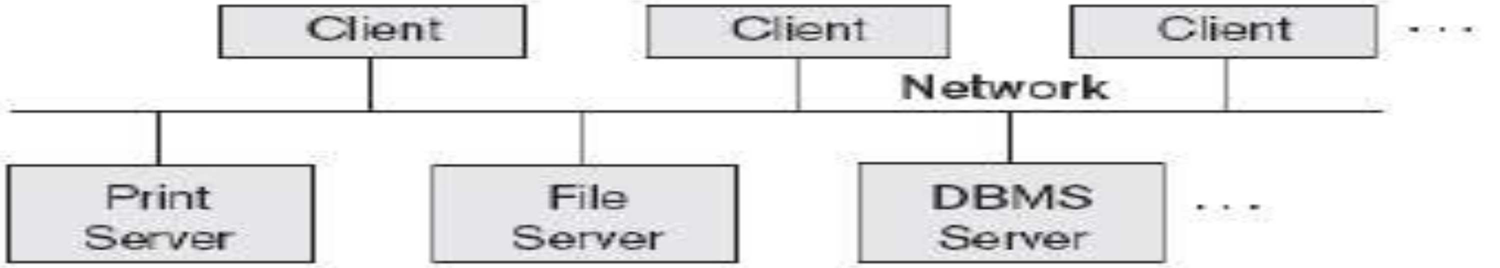


Figure 2.5.2(a) : Logical two-tier client/server architecture

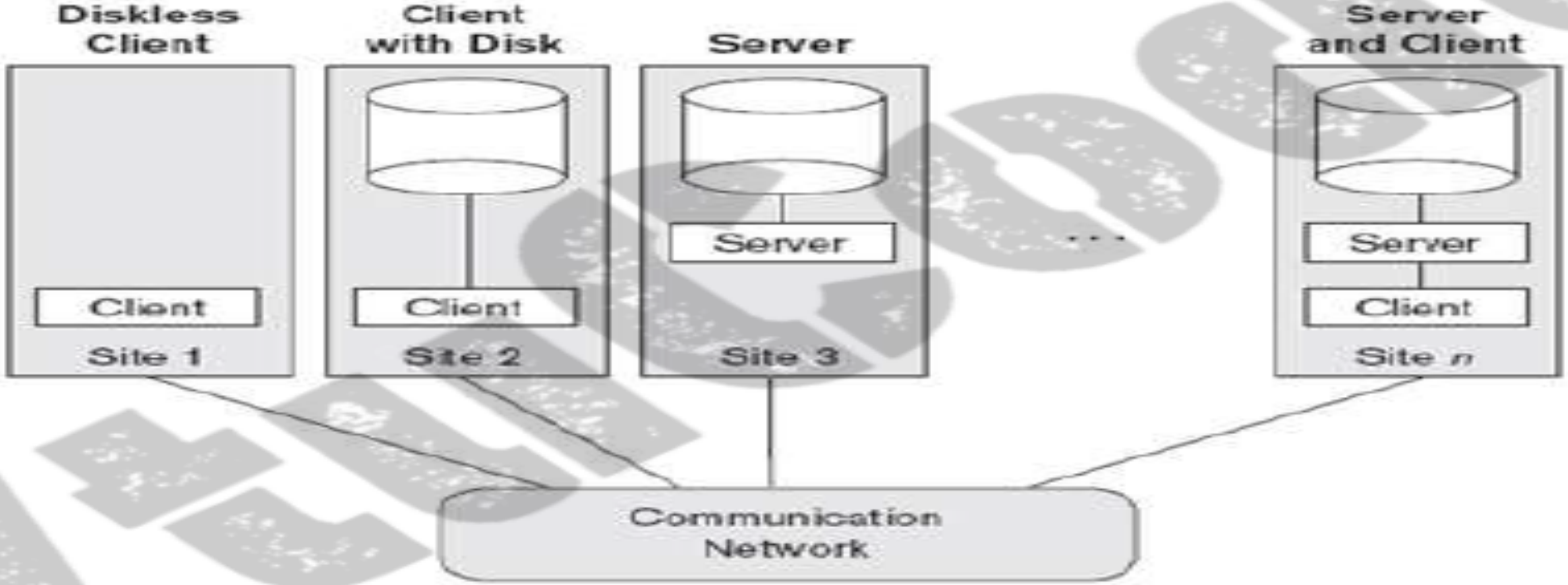


Figure2.5.2(b) : Physical two-tier client/server architecture.

- 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.

idea

- - define specialized servers with specific functionalities.
- - for example file server that maintains the files of the client machines
- - The resources provided by specialized servers can be accessed by many client machines.
- The client machines provide the user with the appropriate interfaces to utilize these servers and local processing power to run local applications

- A **client** is 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.

Two tier client/server Architectures

- In RDBMS user interfaces and application programs moved to client side.
- Query and transaction functionality are on server side(query server/transaction server).

- When DBMS access is required the application program establishes a connection with the DBMS (server side).
- Open database connectivity(ODBC) provides API (application programming interface) allows programs (client-side) to call DBMS (server side).
- ODBC helps in establishing the connection.

Advantages of two tier architecture

  - Simplicity
  - Compatibility

- It is know as two tier architecture because its components are distributed over 2 system the client and the server.
- Emergence of world wide web led to three tier architecture.

Three Tier architecture.

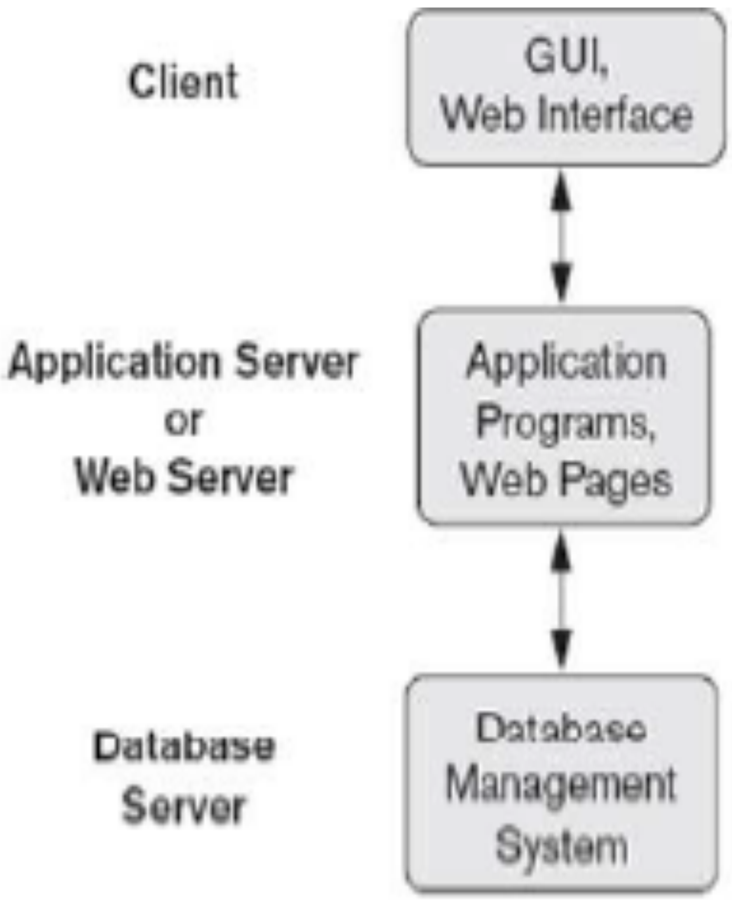
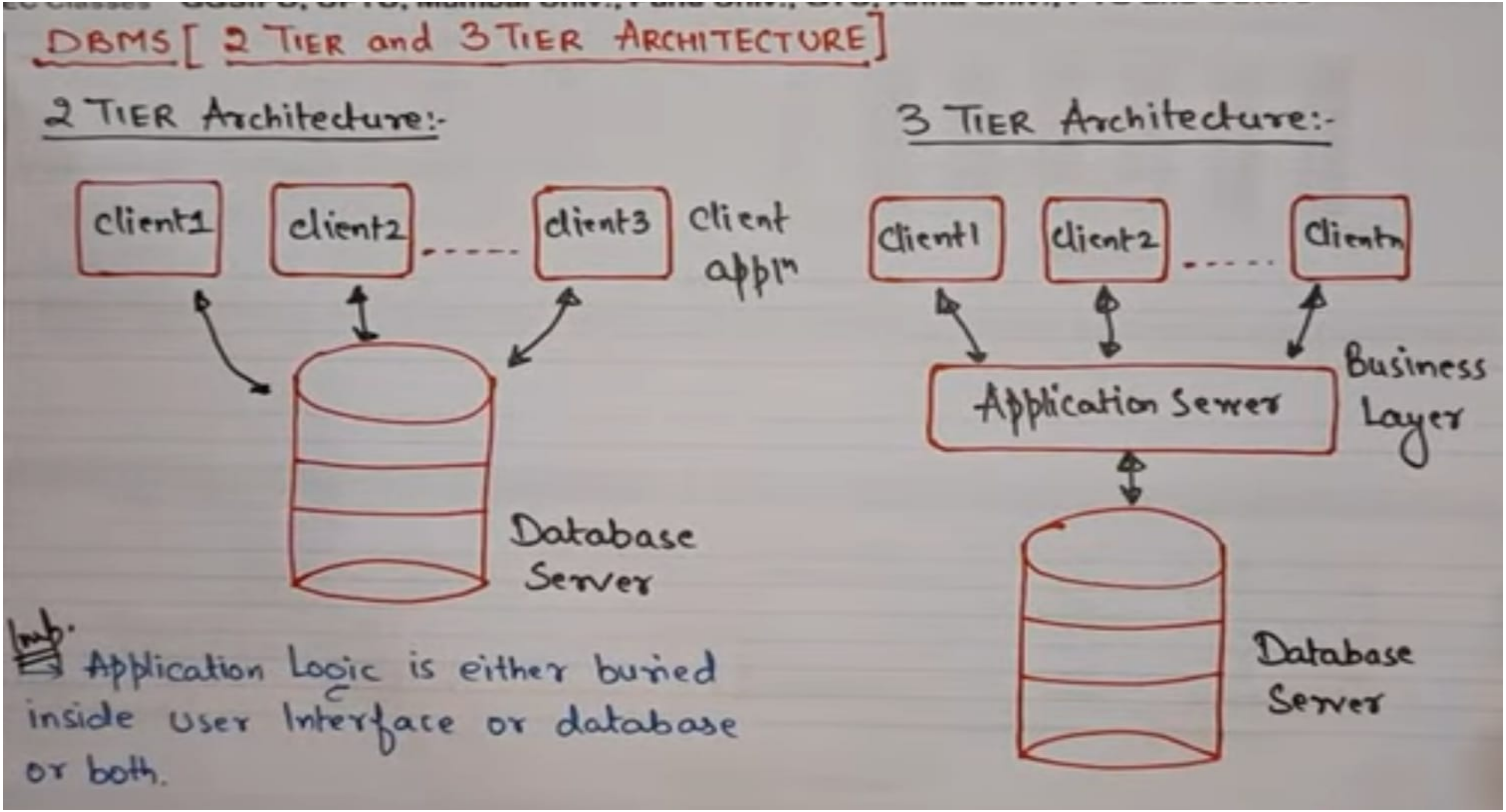


Figure 2.5.4(a): Logical three-tier client/server architecture

- Its called as three tier architecture because it has a additional intermediate layer between client and database server .
- It is called application server or web server based on the application .
- Application server or web server stores rules used to access data from the database server.
- This Intermediate layer accepts requests from the client, processes the requests and sends database commands to the database server and then this intermediate layer acts as channel for passing the process data from database server to client.



DBMS [2 TIER and 3 TIER ARCHITECTURE]

2 TIER Architecture:-

3 TIER Architecture:-

Imp. Application Logic is either buried inside user Interface or database or both.

N-tier Architecture

- It is possible to divide the layers between the user and the stored data further into finer components, thereby giving rise to n-tier architectures; where n may be four or five tiers. The business logic layer is divided into multiple layers

Advantage

Any one tier can run on an appropriate processor or operating system platform and can be handled independently.

Classification of Database Management Systems

Criteria used to classify DBMSs are

1. Data model on which the DBMS is based

Relational: represents a database as a collection of tables, where each table can be stored as a separate file.

Object: defines a database in terms of objects, their properties, and their operations.

Objects with the same structure and behavior belong to a class, and classes are organized into hierarchies (or acyclic graphs). The operations of each class are specified in terms of predefined procedures called methods.

- **Hierarchical and network (legacy):** The network model represents data as record types and also represents a limited type of 1:N relationship, called a set type. The hierarchical model represents data as hierarchical tree structures. Each hierarchy represents a number of related records.
- **Native XML DBMS:** uses hierarchical tree structures. It combines database concepts with concepts from document representation models. Data is represented as elements; with the use of tags, data can be nested to create complex hierarchical structures.

2. Number of users supported by the system

Single-user: support only one user at a time and are mostly used with PCs.

Multi user: support concurrent multiple user.

3. Number of sites over which the database is distributed

Centralized: data is stored at a single computer site

Distributed: can have the actual database and DBMS software distributed over many sites, connected by a computer network

- **Homogeneous DDBMSs** use the same DBMS software at all the sites
- **Heterogeneous DDBMSs** can use different DBMS software at each site

4. Cost

- **Open source:** products like MySQL and PostgreSQL that are supported by third party vendors with additional services.
- **Different types of licensing:** Standalone single user versions of some systems like Microsoft Access are sold per copy or included in the overall configuration of a desktop or laptop. In addition, data warehousing and mining features, as well as support for additional data types, are made available at extra cost.

5. On the basis of the **types of access path options** for storing files

- One well-known family of DBMSs is based on inverted file structures.

(Inverted file structure is a data structure that maps content to its location in a database.)

6. General purpose or Special purpose

- When performance is a primary consideration, a special-purpose DBMS can be designed and built for a specific application; such a system cannot be used for other applications without major changes.
- Many airline reservations and telephone directory systems developed in the past are special-purpose DBMSs.