UNIVERSITY INSTITUTE of COMPUTING

Asia's Fastest Growing University

CU CHANDIGARH UNIVERSITY

Database Management System CAT -202

Syllabus

UNIT-I

- **Introduction:** Overview of Database Management System: Various views of data Models, Schemes and Introduction to database Languages & Environments, Advantages of DBMS over file processing systems, Responsibility of Database Administrator. Three level architecture of Database Systems: Introduction to client/Server architecture.

Syllabus

UNIT-II

- **Data Models:** E-R Diagram (Entity Relationship), mapping Constraints, keys, Reduction of E-R diagram into tables. Network & Hierarchical Models,
- **File Organization:** Sequential File, index sequential files, direct files, Hashing, B-trees Index files, Inverted Lists., Relational Model.
- **Relational Algebra:** Meaning & various operations (set operations, select, project, join, division), Order
- **Relational calculus:** Domain, Tuple, Well Formed Formula, specification, quantifiers, Introduction to Query Language, QBE.

Syllabus

UNIT-III

- Integrity constrains, functional dependencies & Normalization, 1st, 2nd, 3rd and BCNF.
- Introduction to Distributed Data processing, Concurrency control: Transactions, Time stamping, Lock-based Protocols.

Reference Books

- Fundamentals of Database Systems by R.Elmasri and S.B.Navathe, 3rd Edition, Pearson Education, New Delhi.
- An Introduction to Database Systems by C.J. Date, 7th Edition, Pearson Education, New Delhi.
- A Guide to the SQL Standard, Data, C. and Darwen, H.3rd Edition, Reading, Addison-Wesley Publications, New Delhi.
- Introduction to Database Management system by Bipin Desai, Galgotia Pub, New Delhi.
- Database System Concepts by A. Silberschatz, H.F.Korth and S.Sudarshan, 3rd Edition, McGraw-Hill, International Edition.
- SQL / PL/SQL, by Ivan Bayross, BPB Publications.

Lecture Contents

- Data model :Meaning & Types
- Database Schema
- Database Instance
- Database State
- Three schema architecture of DBMS
- DBMS Languages
- DBMS Interfaces
- Database Systems utilities
- Client-Server Architecture

Data Models

- **Data Model**: A set of concepts to describe the *structure* of a database, and certain *constraints* that the database should obey.
- **Data Model Operations**: Operations for specifying database retrievals and updates by referring to the concepts of the data model. Operations on the data model may include *basic operations* and *user-defined operations*.

Categories of data models

- **Conceptual** (**high-level**, **semantic**) data models: Provide concepts that are close to the way many users *perceive* data. (Also called **entity-based** or **object-based** data models.)

- **Physical** (**low-level**, **internal**) data models: Provide concepts that describe details of how data is stored in the computer. concepts provided by this data model are generally meant for computer specialist not for end users.

- **Implementation** (**representational**) data models: Provide concepts that fall between the above two, balancing user views with some computer storage details.Also called as record based data model.

Basic Terminology

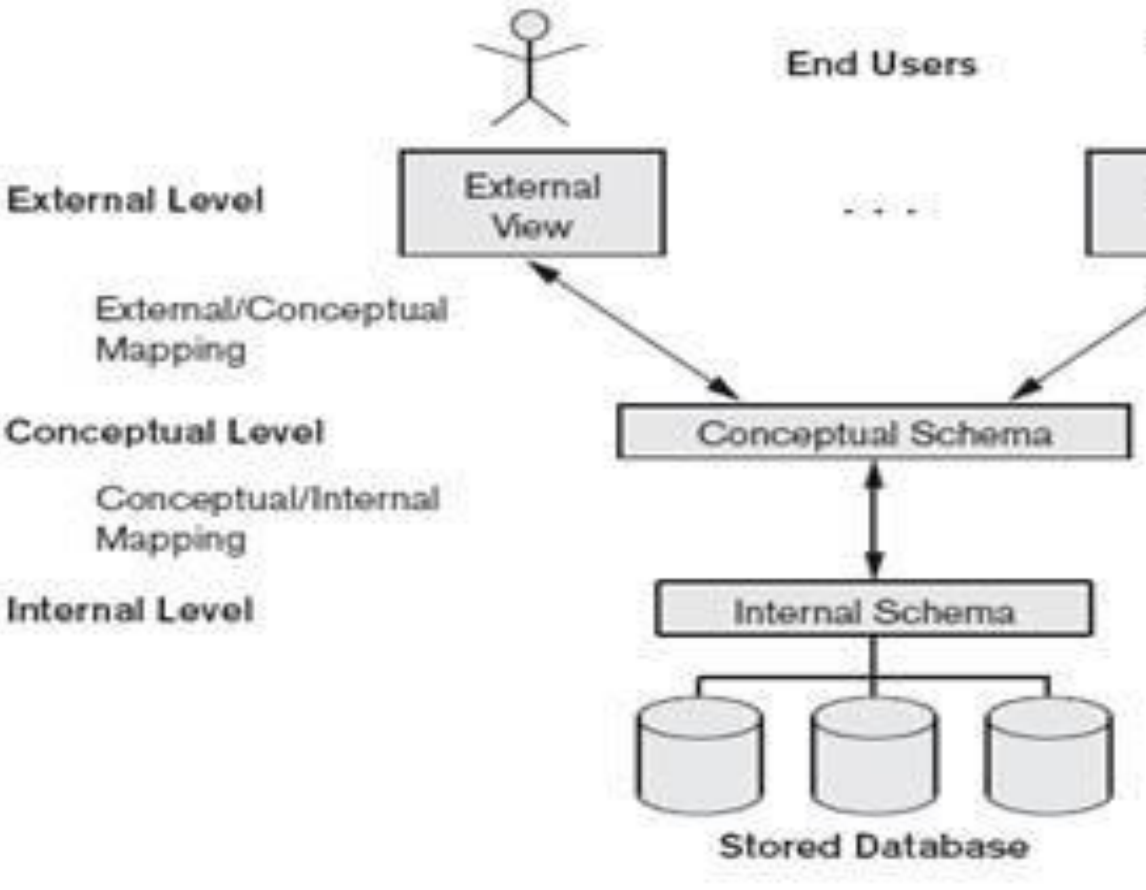
- **Database Schema**: The *description* of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- **Schema Diagram**: A diagrammatic display of (some aspects of) a database schema.
- **Schema Construct**: A component of the schema or an object within the schema, e.g., STUDENT, COURSE.
- **Database Instance**: The actual data stored in a database at a *particular moment in time*. Also called **database state** (or **occurrence**).

Database State

- **Database State:** Refers to the content of a database at a moment in time.
- **Initial Database State:** Refers to the database when it is loaded
- **Valid State:** A state that satisfies the structure and constraints of the database.
- **Note:**
  - The **database schema** changes *very infrequently*. The **database state** changes *every time the database is updated.*
  - **Schema** is also called **intension**, whereas **state** is called **extension**.

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Three Schema Architecture Diagram



3 Levels - 3 Schemas

- **Level 1 : Internal schema** at the internal level to describe physical storage structures and access paths. Typically uses a *physical* data model.
- **Level 2 : Conceptual schema** at the conceptual level to describe the structure and constraints for the *whole* database for a community of users. Uses a *conceptual* or an *implementation* data model.
- **Level 3 : External schemas** at the external level to describe the various user views. Usually uses the same data model as the conceptual level.

Data Independence

- **Mappings** among schema levels are needed to transform requests and data. Programs refer to an external schema, and are mapped by the DBMS to the internal schema for execution.

- When a schema at a lower level is changed, only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence. The higher-level schemas themselves are *unchanged*. Hence, the application programs need not be changed since they refer to the external schemas.

Data Independence Types

- **Logical Data Independence**: The capacity to change the conceptual schema without having to change the external schemas and their application programs.
- **Physical Data Independence**: The capacity to change the internal schema without having to change the conceptual schema.

DBMS Languages

- **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).
- In some DBMSs, separate **storage definition language** (**SDL**) and **view definition language** (**VDL**) are used to define internal and external schemas.

DBMS Languages

- **Data Manipulation Language (DML)**: Used to specify database retrievals and updates.
  - DML commands (**data sublanguage**) can be *embedded* in a general-purpose programming language (**host language**), such as COBOL, C or an Assembly Language.
  - Alternatively, *stand-alone* DML commands can be applied directly (**query language**).

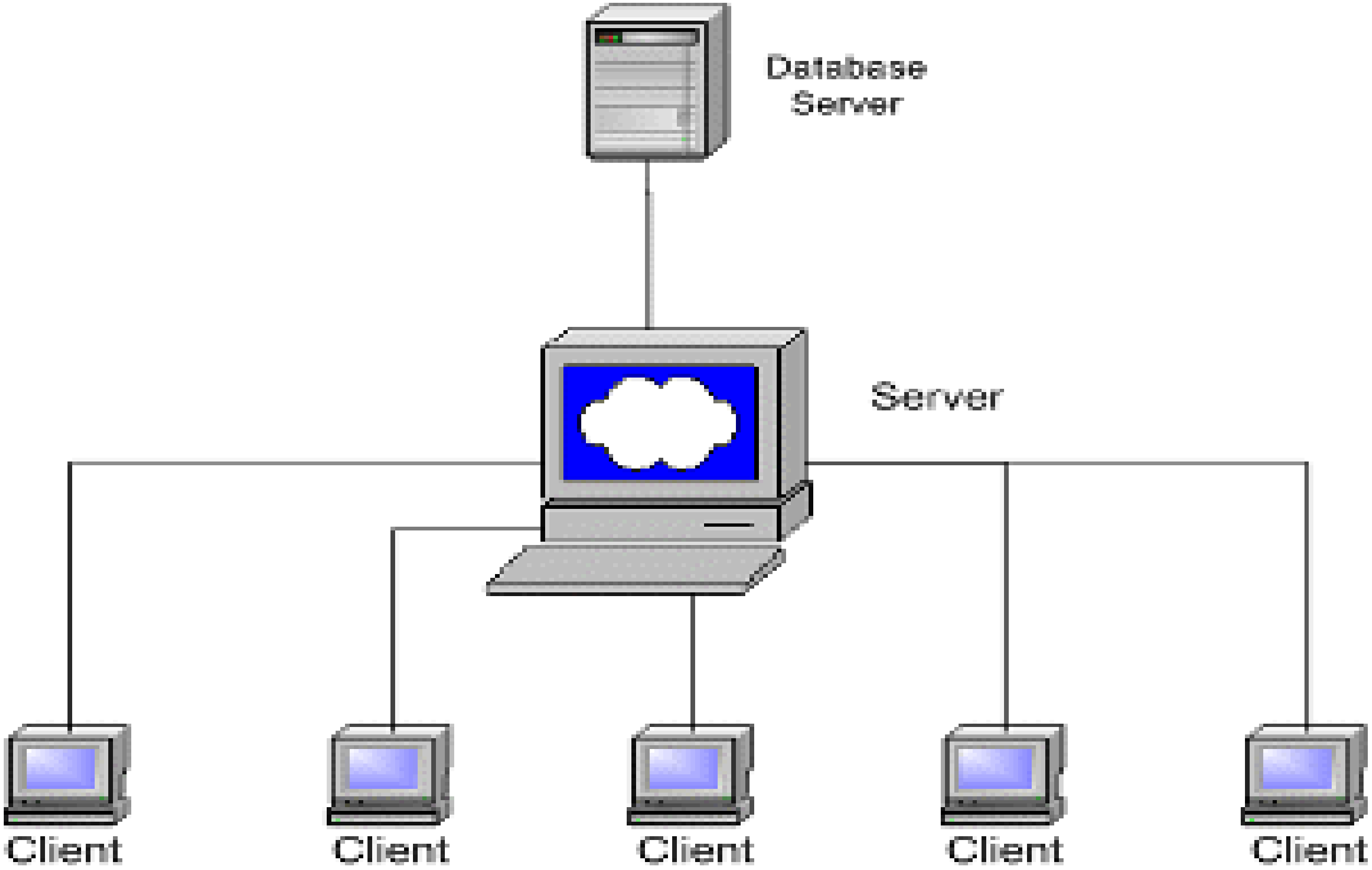
DBMS Languages

- **High Level** or **Non-procedural Languages:** e.g., SQL, are *set-oriented* and specify what data to retrieve than how to retrieve. Also called *declarative* languages.
- **Low Level** or **Procedural Languages:** record-at-a-time; they specify *how* to retrieve data and include constructs such as looping.

Client-Server Architecture

Client-server architecture can be considered as a network environment that exchanges information between a server machine and a client machine where server has some resources that can be shared by different clients.

Client-Server Architecture



Basic Client-Server Architectures Includes :

- **Specialized Servers with Specialized functions**
- **Clients**
- **DBMS Server**

Specialized Servers with Specialized functions:

- File Servers
- Printer Servers
- Web Servers
- E-mail Servers

Clients:

- Provide appropriate interfaces and a client-version of the system to access and utilize the server resources.
- Clients maybe diskless machines or PCs or Workstations with disks with only the client software installed.
- Connected to the servers via some form of a network. (LAN: local area network, wireless network, etc.)

DBMS Server

- Provides database query and transaction services to the clients
- Sometimes called query and transaction servers

Example

Take an example of a file server to understand the process of a client/server network, the file server acts as a storage space on the network for the files, spreadsheets, databases, etc. Instead of storing these records on every individual computer, the file server allows the clients to store their files on one central computer and make them sharable. The client-server architecture is beneficial in reducing the multiple iterations of a single file and allowing the organization to have one centralized point for every computer to access the same file.

The Database Administrator (DBA)

DBA is the person who provides the necessary technical support for implementing the decisions and strategies. He is responsible for the overall control of the DB system at a technical level.

DBA responsibilities

- Defining the conceptual schema
- Defining the internal schema
- Interacting with the users
- Defining security & integrity constraints.
- Defining restore schemas
- Monitoring performance & responsibility to changing requirements

Thank You