Lecture 3

Database Management System

Er. Shiva Kunwar

Lecturer, GU

Lesson 1: Introduction to DBMS (5hrs)

1. Overview of Database and DBMS
2. Characteristics and Applications
3. **Data Abstraction and Independence**
4. **Database Users and Administrator**
5. **Application Architecture**
6. Basics of Database Language (DDL, DML, DCL)

Data Abstraction

- Data abstraction is the technique of hiding the complexity of the database to its users.
- Data abstraction simplifies users' interactions with the system.
- There are three levels of data abstraction which are discussed below.
  - **Physical Level or Internal Level**
  - **Logical Level or Conceptual Level**
  - **View Level or External Level**
  - These are known as three-schema architecture

Physical Level or Internal Level

- It is the lowest level of abstraction.
- It describes how the data in the database are actually stored.
- This level describes complex low-level data structures in detail and is concerned with the way the data is physically stored.
- Data only exists at physical level

Logical Level or Conceptual Level:

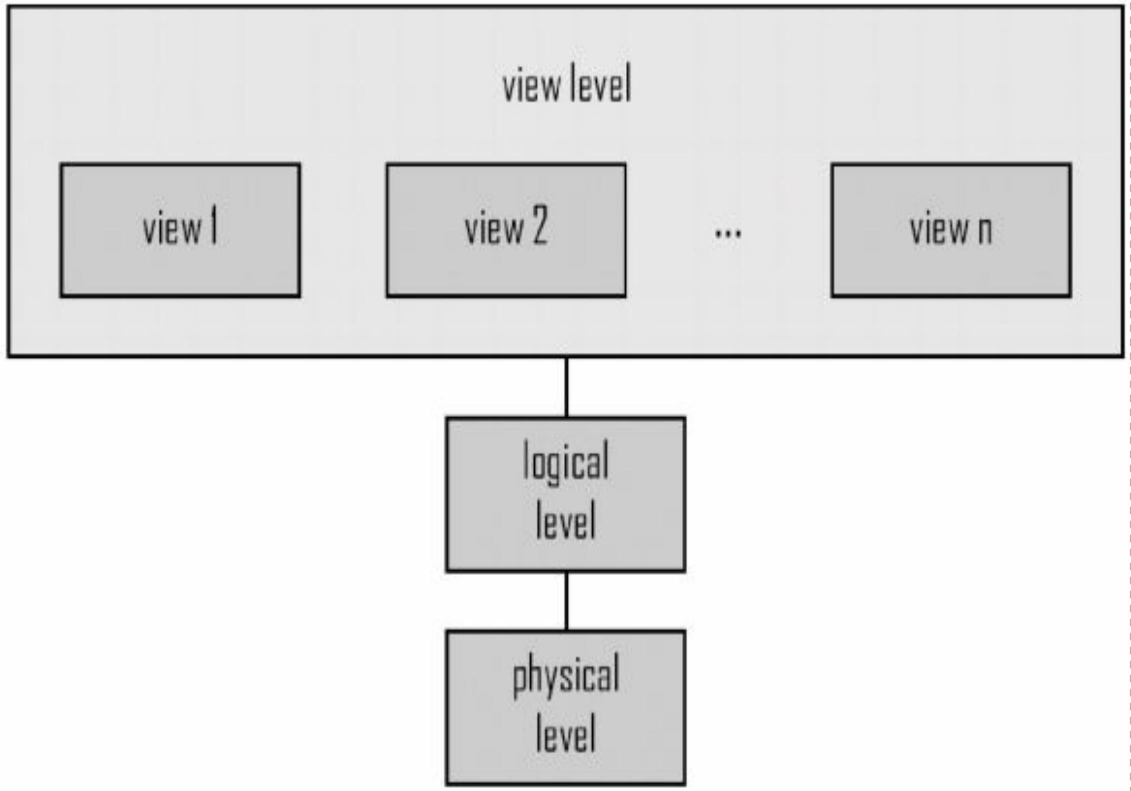
- This is the next higher level of abstraction and describes what data are stored in the database, and what relationships exist among those data.
- It describes the structure of whole database and hides details of physical storage structure.
- It concentrates on describing entities, data types, relationships, attributes and constraints.
- All of the views must be derivable from this conceptual schema.
- Database administrators, who must decide what information to keep in the database, use the logical level of abstraction

View Level or External Level

- It is the highest level of abstraction and is concerned with the way the data is seen by individual users.
- This level simplifies the users’ interaction with the system.
- It includes several user views and hence is guided by the end user requirement.
- It describes only those part of the database in which the users are interested and hides rest of all from those users.
- Each user group refers to its own external schema.
- The overall design of the database which is not expected to change frequently is called the database schema.

Data Abstraction contd.

- The DBMS must transform a request specified on an external schema into a request against the conceptual schema,
- and then into a request on the internal schema for processing over the database.
- The process of transforming requests and results between levels is called mapping.



Data Abstraction Example

- *view level*
  - View result
  - View student information
- *logical level: entire database schema*
  - Courses (CourseNo, CourseName, Credits, Dept)
  - Student (StudentID, Lname, Fname, Level, Major)
  - Grade (StudentID, CourseNo, mark)
- *physical level:*
  - how these tables are stored, how many bytes it required etc.

Data Abstraction Example

- *view level*
  - View student information
- **CREATE VIEW StudentInformation AS**
- **SELECT StudentID, Lname, Fname, Level, Major**
- **FROM Student;**
- Users can query this view without needing to know the detailed structure of the "Student" table.
- **SELECT * FROM StudentInformation;**

Data Abstraction Example

- *logical level: entire database schema*
  - Courses (CourseNo, CourseName, Credits, Dept)
  - Student (StudentID, Lname, Fname, Level, Major)
  - Grade (StudentID, CourseNo, mark)
- **SELECT Student.StudentID, Student.Lname, Student.Fname, Courses.CourseName, Grade.Mark**
- **FROM Student**
- **JOIN Grade ON Student.StudentID = Grade.StudentID**
- **JOIN Courses ON Grade.CourseNo = Courses.CourseNo;**

Data Abstraction Example

- *physical level:*
  - how these tables are stored, how many bytes it required etc.
  - **Student Table: Each row may require, for example, 100 bytes.**
  - **Courses Table: Each row may require 80 bytes.**
  - **Grade Table: Each row may require 60 bytes.**
  - If there are 1,000 students, 50 courses, and 20,000 grades:
  - Storage for "Student" table: 100 bytes/row * 1,000 rows = 100,000 bytes.
  - Storage for "Courses" table: 80 bytes/row * 50 rows = 4,000 bytes.
  - Storage for "Grade" table: 60 bytes/row * 20,000 rows = 1,200,000 bytes.

Data Independence

- It can be defined as the capacity to change the schema at one level of a database system without having to change schema at the next higher level.
- There are mainly two types of Data Independence

1. **Logical data independence**
2. **Physical data independence**

Data Independence

- **Logical data independence**
  - The ability to change the Logical schema (Conceptual) without changing the External schema (User View) is called Logical Data Independence.
  - For example, the addition or removal of new entities, attributes, or relationships to the conceptual schema should be possible without having to change existing external schemas or having to rewrite existing application programs.

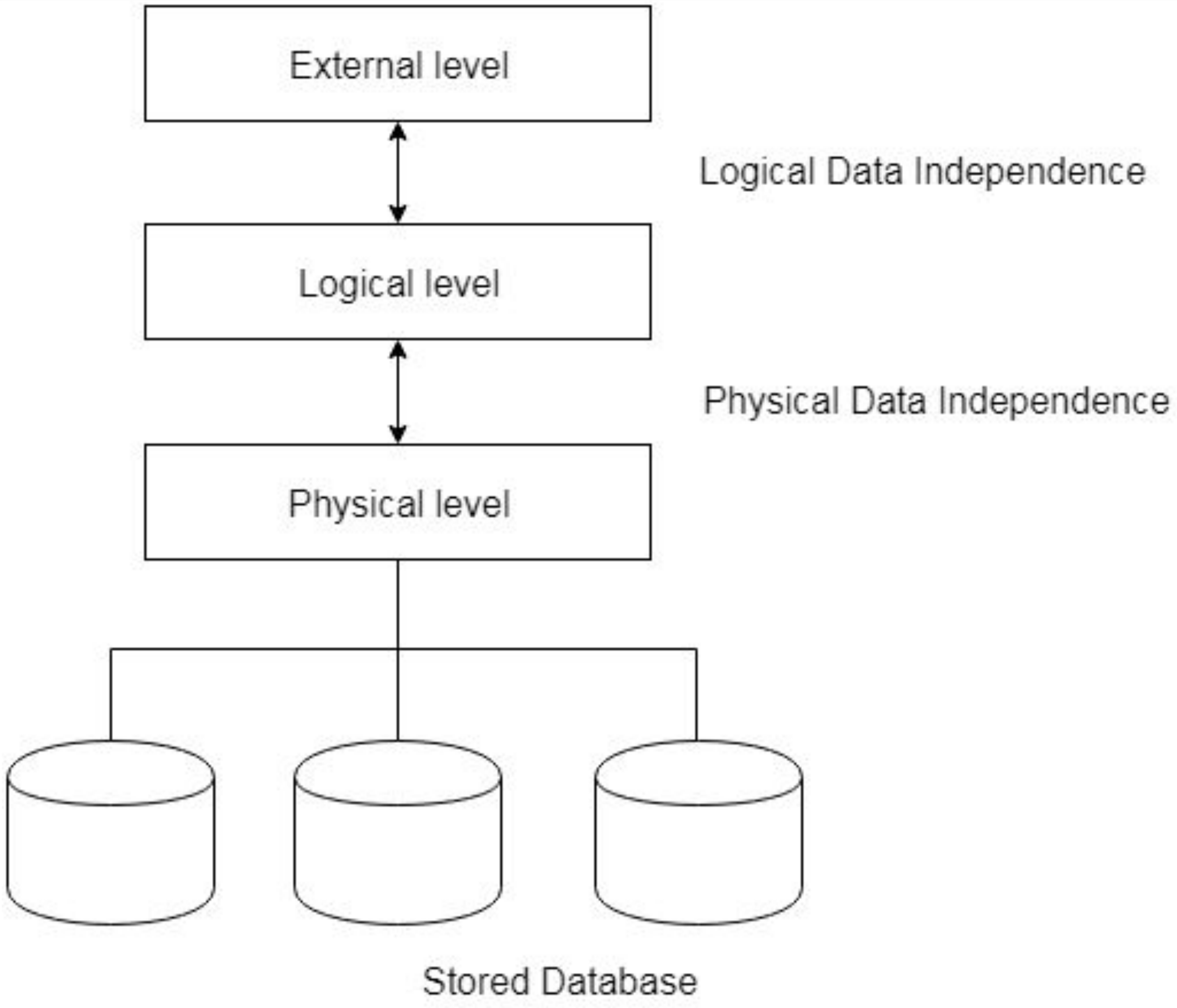
Data Independence

- **Physical data independence**
  - The ability to change the Physical schema (Internal) without changing the Logical schema (Conceptual) is called Physical Data Independence.
  - For example, a change to the internal schema, such as using different file organization or storage structures, storage devices, or indexing strategy, should be possible without having to change the conceptual or external schemas.

Data Independence Example

- **Logical Data Independence**: Adding a new column to a table or creating a new table to store additional information.
  - When a new column is added to the table, the existing application programs won't be aware of this change.
  - They can still retrieve data from the original columns without any modification.
- **Physical Data Independence**: Changing the storage system from one type of disk to another or modifying the indexing strategy for better performance.

Data Independence



Database Users

- People who work with a database can be categorized as database users or database administrators.
- There are four different types of database-system users, differentiated by the way they expect to interact with the system.
- Different types of user interfaces have been designed for the different types of users.
  1. **Naïve Users**
  2. **Application Programmers**
  3. **Sophisticated Users**
  4. **Specialized Users**

Database Users

1. **Naïve Users**
   - Naïve users are the simple users who just uses the application that have been built previously. For example, a customer who uses ATM simply invokes the program which checks his username, password and balance: and finally allows to withdraw certain amount from his account
2. **Application Programmers**
   - These are the computer professionals who write application programs. They build user interfaces to interact with the database and hence it makes naïve users easy. Application programmers uses many application development tools like RAD for the quick development of the application.

Database Users

3. **Sophisticated Users**
   - Sophisticated users do not interact with the database through the application programs and hence they generate their own query in a database query language (DML). Analyst who submit queries to explore data in the database fall in this category.
4. **Specialized Users**
   - Specialized users are the sophisticated users. These are highly advanced users and write specialized database applications. It could be computer-aided design systems, knowledge based and expert systems that store data with complex data types. Scientists, researchers fall in this category.

Database Administrator

- The person who has control over both data and the program that accesses those data are called database administrator(DBA). The functions of database administrator are:
- **schema definition:**
  - The DBA creates the original structure or schema of the database by using DDL.
- **schema and physical organization modification:**
  - The changes needed in any organization is analyzed and then appropriate change is made in the database schema by the DBA.

Database Administrator

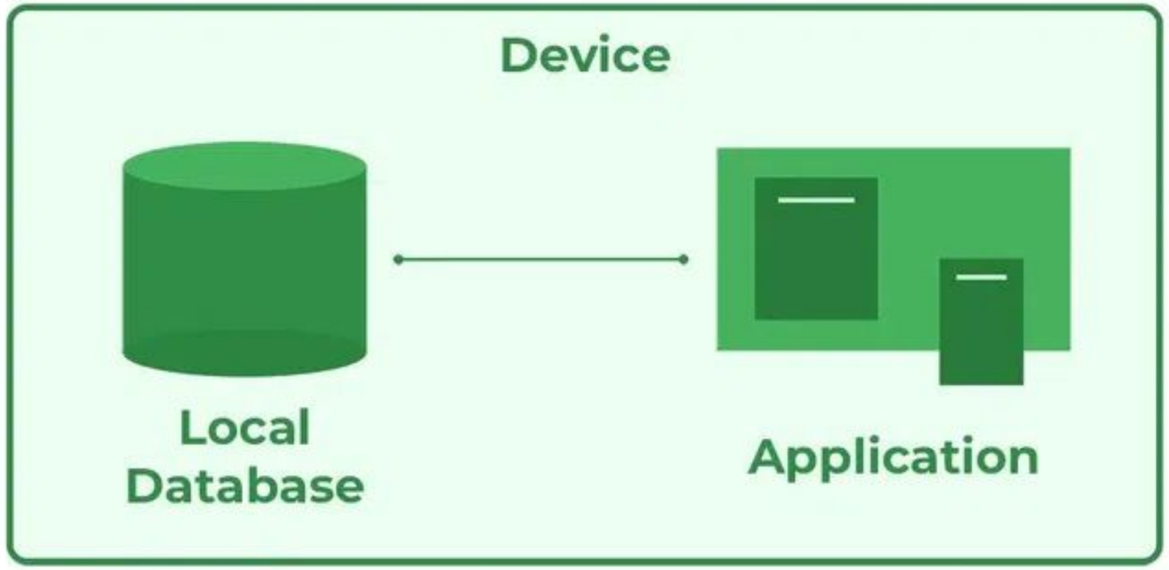
- **granting of authorization:**
  - By granting different types of the authorization, DBA can regulate which part of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system.
- **runtime maintenance:**
  - DBA periodically backup the database, ensures that enough free disk space is available for normal operation. Monitoring jobs running on the database and ensuring that performance is not degraded too much with expensive tasks.

Application Architecture

- Types of DBMS Architecture are
- **1-Tier Architecture**
- **2-Tier Architecture**
- **3-Tier Architecture**

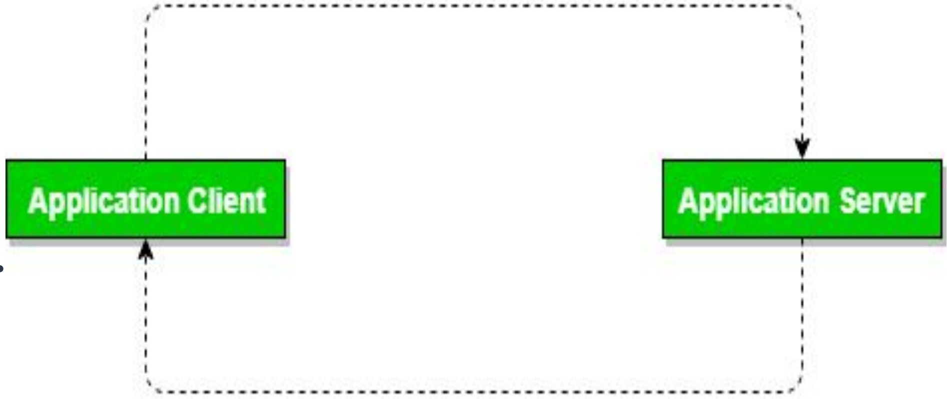
1-Tier Architecture

- The database is directly available to the user.
- The user can directly sit on the DBMS and use it that is, the client, server, and Database are all present on the same machine.
- For Example: to learn SQL we set up an SQL server and the database on the local system.
- This enables us to directly interact with the relational database and execute operations.



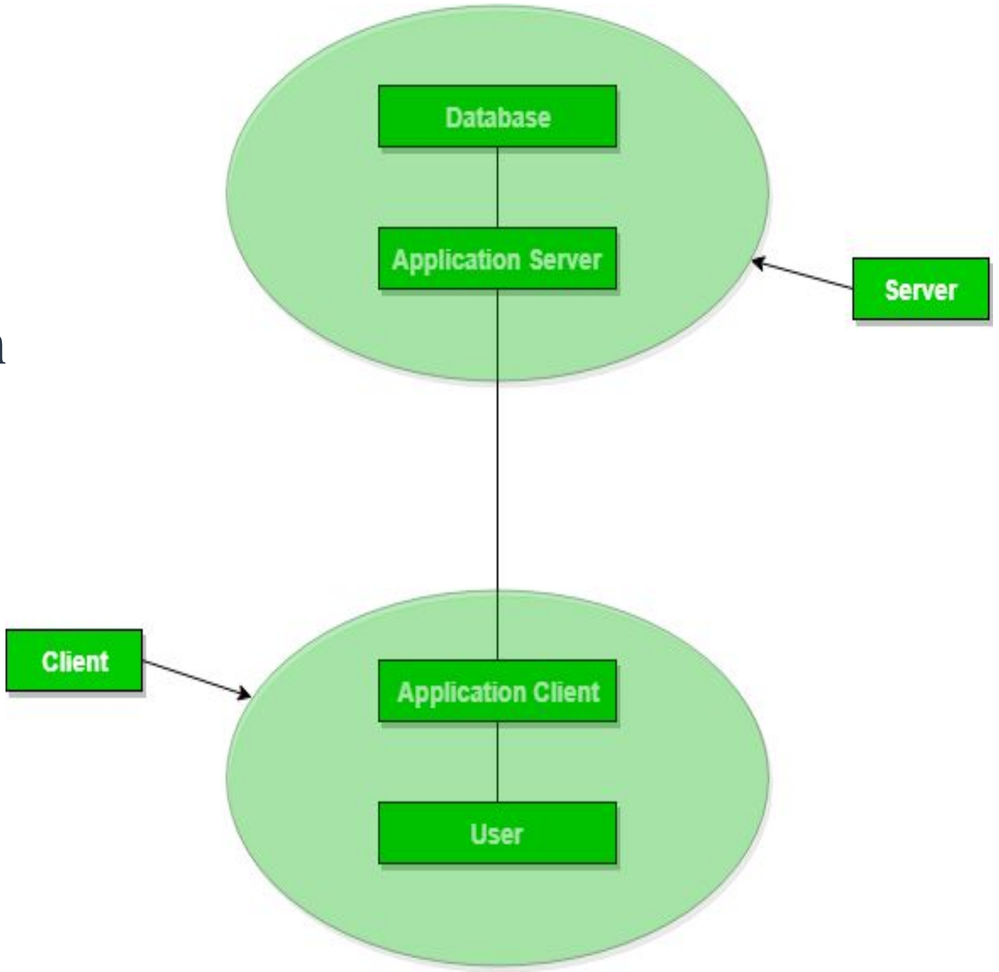
2-Tier Architecture

- The 2-tier architecture is similar to a basic client-server model.
- The application at the client end directly communicates with the database on the server side.
- The server side is responsible for providing query processing and transaction management functionalities.
- On the client side, the user interfaces and application programs are run.
- The application on the client side establishes a connection with the server side to communicate with the DBMS.



3-Tier Architecture

- The client does not directly communicate with the server.
- Instead, it interacts with an application server which further communicates with the database system and then the query processing and transaction management takes place.
- This intermediate layer acts as a medium for the exchange of partially processed data between the server and the client.
- This type of architecture is used in the case of large web applications.



END OF LECTURE 3

- SHIVA.KUNWAR@HOTMAIL.COM
- +977-9819123654

Google classroom code : drbzdcf

PREVIEW FOR LECTURE 4

BASICS OF DATABASE LANGUAGE

(DDL, DML, DCL)

LAB SESSION