Introduction to DBMS

Ram Datta Bhatta

Data, Database, DBMS, Database Application?

Data and Database

- **Data:** raw facts, such as numbers, words, measurements, observations, or even just descriptions of things. For examples: **Ram** represents name of a person. Thus, it is also data.
- **Database:** the collection of data. It generally contains interrelated information. An employee database may contain personnel information of employees working in an organization, their duty, job description, family information etc.

DBMS

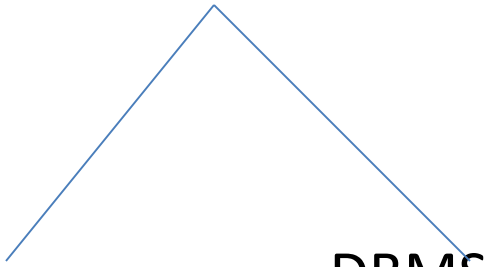
- ✓ DBMS is a software package/program designed to define, manipulate, retrieve, and manage data in a database.
- ✓ DBMS contains collection of interrelated data and a set of programs to access those data.
- ✓ A DBMS generally manipulates the data itself, the data format, field names, record structure, and file structure. Examples: MS-Access, MySQL, MS-SQL, Oracle, Informix, IMS, DB2, FoxPro etc.

DBMS

- DBMS is:
  - A collection of interrelated data.
  - A set of programs to access the data.
  - An environment that is both convenient and efficient to use.
- DBMS stores data in such a way that it becomes easier to retrieve, manipulate and produce information.

DBMS

Database System



Database

- Structured(University, Bank)
- Unstructured(Web Page)

DBMS

SQL server

Oracle 8i,9,11

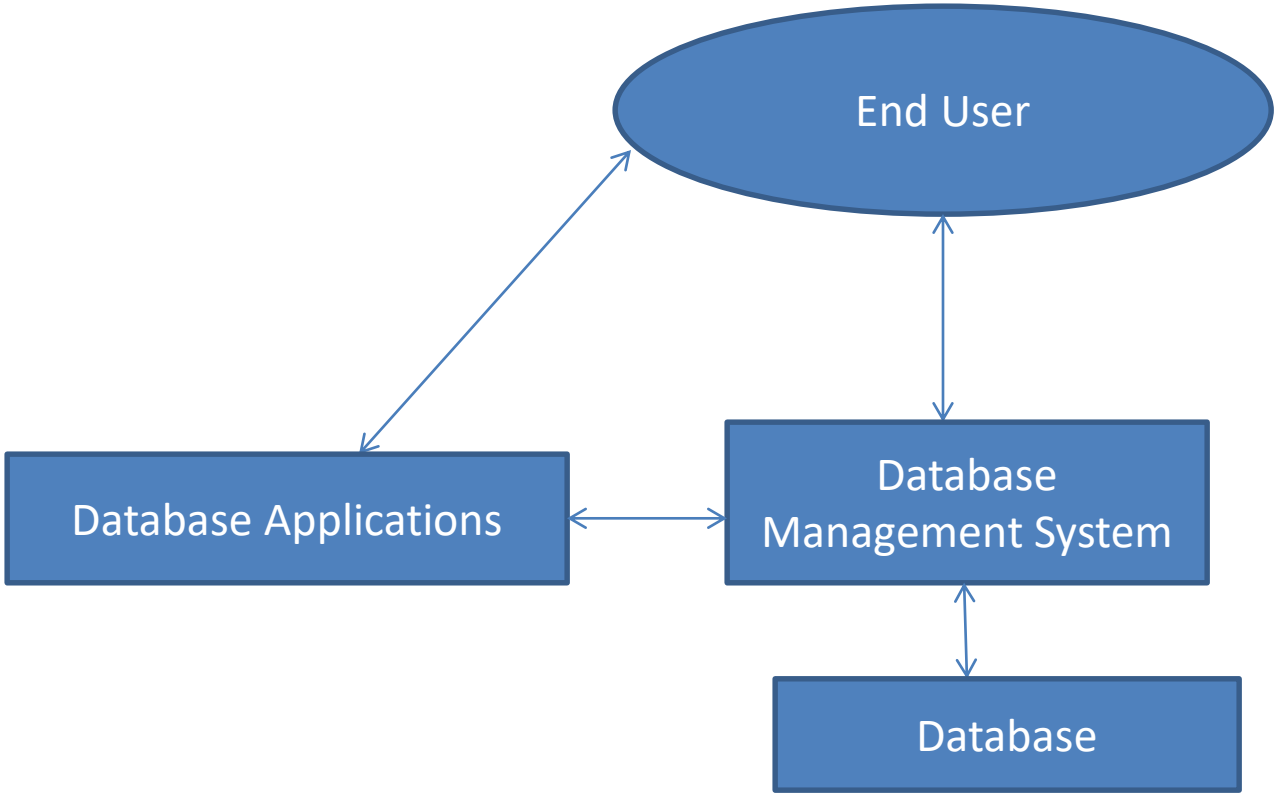
My SQL

DB2

Database Application/System

- ✓ A computer program interacts with the database by issuing an appropriate request (typically an SQL statement) to the DBMS is known as database application.
- ✓ It is a software/program which uses database (generally as backend) is known as database application.
- ✓ Let us consider an example of student management system implementing in a school. Let us suppose the management system has been developed using Java programming language and database oracle. Here, Java has been used for programming/processing purpose but actual information (data about student) is contained in the database. Here, Java is called front-end and oracle database is known as back-end.

DBMS


End User
Database Applications
Database Management System
Database

Use of DBMS (Applications)

- ✓ Banking : All transactions
- ✓ Airlines: Reservations, Schedules
- ✓ Universities: Registration, Grades
- ✓ Sales: Customers, Products, Purchases
- ✓ Manufacturing: Production, Inventory, Orders, Supply chain
- ✓ Human Resources: Employee records, salary, tax deductions.
- ✓ Online retailers: Order tracking.

Why DBMS? How does it differ from Traditional File System?

- ✓ Reduced Data redundancy and inconsistency.
- ✓ Easy in accessing data.
- ✓ Isolation of data and application
- ✓ Maintains Data Integrity
- ✓ Removes concurrent-access anomalies (Concurrency Control)
- ✓ Atomicity problems
- ✓ Multiple Views
- ✓ Enhanced Security

[For detail: see attached word document]

Difference between file system and DBMS

File System

- File system is a software that manages and organizes the files in a storage medium(HDD) within a computer.
- Redundant data can be present in the file system
- It does not provide backup and recovery of data if it is lost.
- there is no efficient query processing in file system
- There is less data consistency in file processing system.
- File system provides less security in comparison to DBMS.
- It is less expensive as compared to DBMs.

DBMS

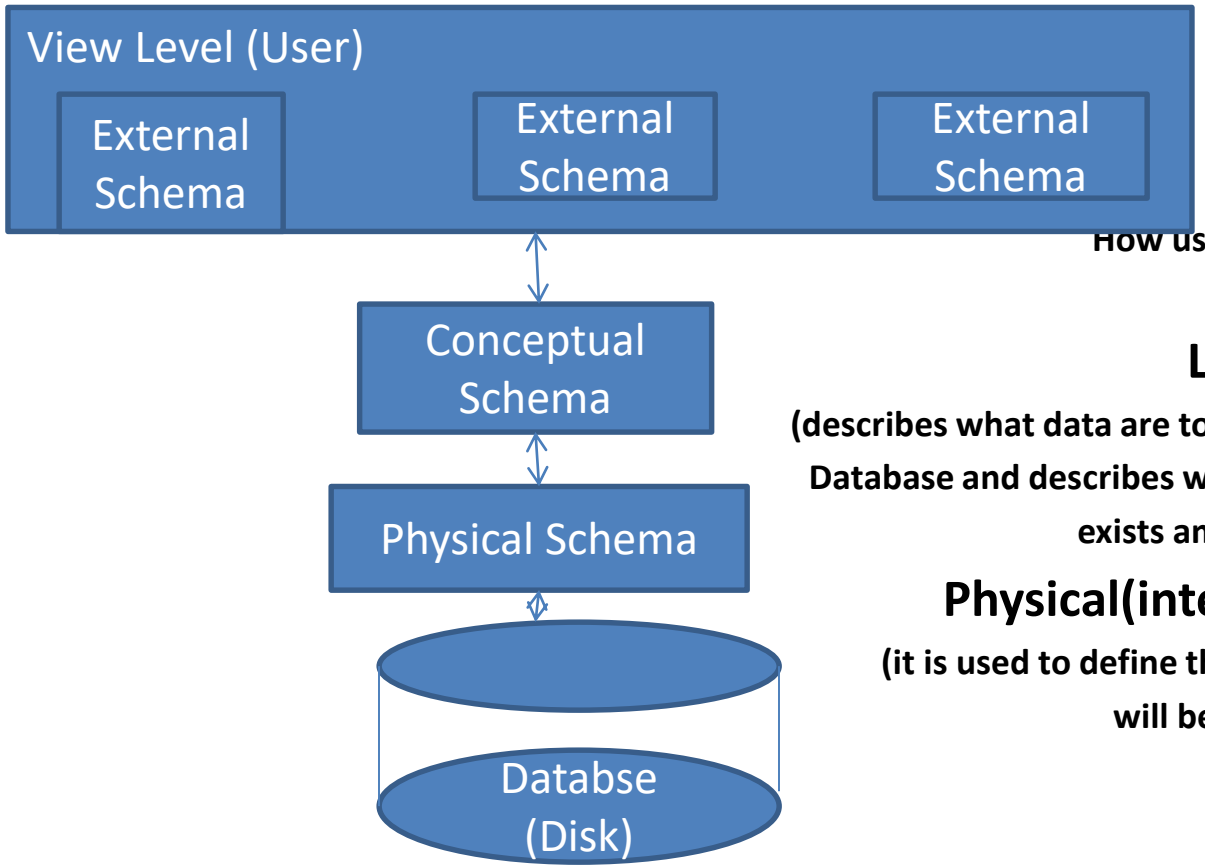
- DBMS is a software for managing the database.
- In DBMS, there is no redundant data.
- It provides backup and recovery of data even if data is lost.
- Efficient query processing is there in DBMS.
- There is more data consistency because of the process of normalization.
- DBMS has more security mechanism as compared to file system.
- It has comparatively higher cost than file system.

Levels of Abstraction,

- ✓ Abstraction: Hiding background detail
- ✓ To ease the user interaction with the database, the developers hide internal irrelevant details from users and provides abstract view of data to users. This process of hiding irrelevant details from user is called **Data Abstraction**

Three Levels of Data Abstraction or Three Schema Architecture

Three levels of data abstraction



View Level

(views describe How users see the data)

Logical Level

(describes what data are to be stored in the Database and describes what relation ship exists among those data)

Physical(internal) Level

(it is used to define that how the data will be stored in block)

Three level of data abstraction

1. Physical Level(Internal Level)

- This is the lowest level of data abstraction.
- It describes how data is actually stored in database. User can get the complex data structure details at this level.

2. Logical Level(conceptual level)

- It describe what type of data?
- It describes how data is stored in database and what relationships exist among those data. (pk, fk, relationships, constraints etc.)

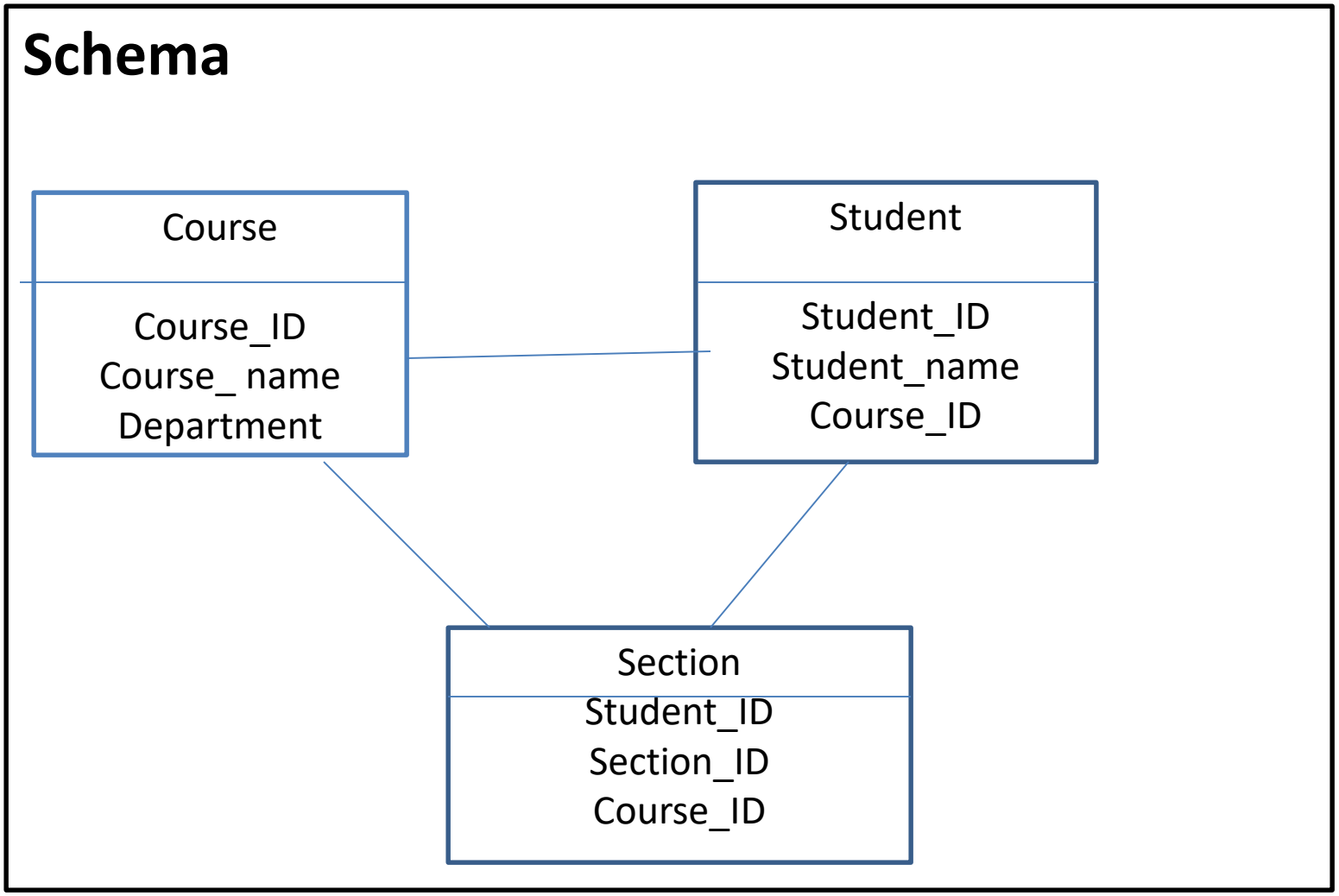
3. View Level

- ✓ Highest level of data abstraction.
- ✓ This level describes only part of the database i.e. it describes the user interaction with database system via application program that hide details of data types. At the View Level, user just interact with system with the help of GUI and enters the details at the screen, they are not aware of how the data is stored and what the data is stored, such details are hidden from them.

Schemas and instances

- ✓ Overall design of the database is called schema. E.g. given a schema shows the relationship between three tables : course, student, section
- ✓ The diagram only shows the design of the database, it does not show the data present in those tables. Schema is only the structured view(design) of a database.

Schema


Course
Course_ID
Course_ name
Department
Student
Student_ID
Student_name
Course_ID
Section
Student_ID
Section_ID
Course_ID

Types of Schemas

- **Physical Schema**
  - It describes database design at physical level i.e. physical structure of database.
  - How the data stored in blocks of storage is described at this level.
  - Database administrator works at this level.
- **Logical Schema**
  - It describes database design at logical level i.e. logical structure of database.
  - Database programmer works at this level.

Instances

- ✓ The data stored in the database at particular moment of time is called instance of database.
- ✓ Example: lets say we have a single table **Student** in the database. Today the table has 100 records, so today the **instance** of the database has 100 records. Lets say we are going to add another 100 records in this table by tomorrow so the instance of the database tomorrow will be 200 records in the table. In short, at a particular moment the data stored in database is called the **instance,** that changes over time when we add or delete data from the database.

Data independence

The ability to modify the schema at one level of the database system without the altering the schema at next higher level.

Two types of data independence

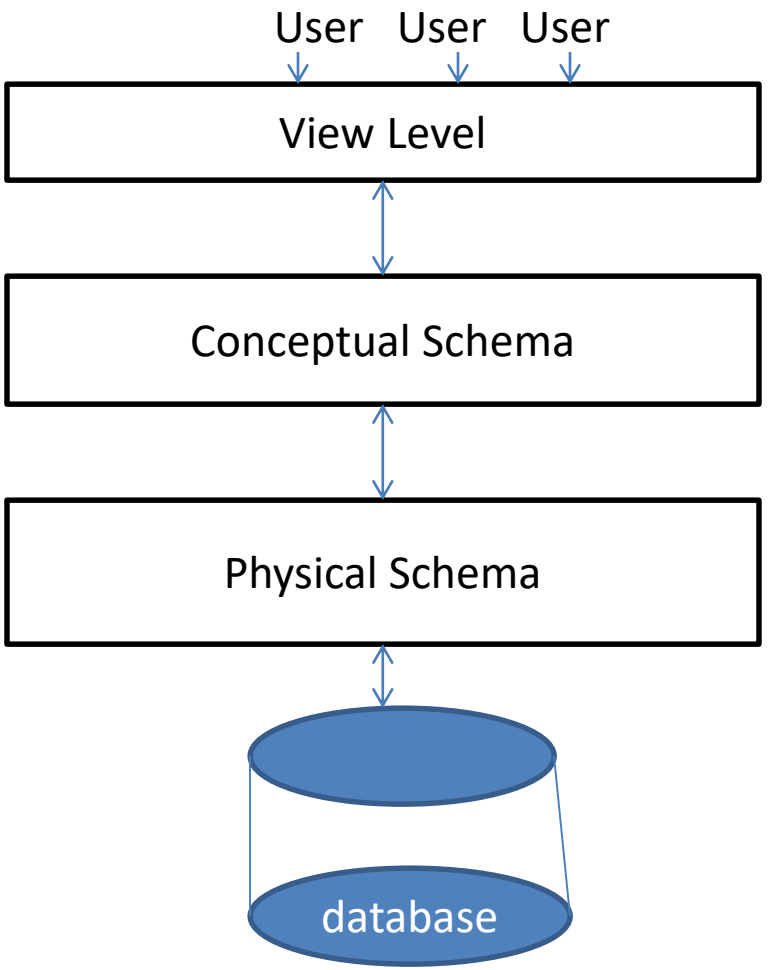
1. Logical Data Independence

Ability to modify the logical schema without changing the external view and application.

2. Physical Data Independence:

Ability to modify the physical schema without changing the logical schema.

Data independence

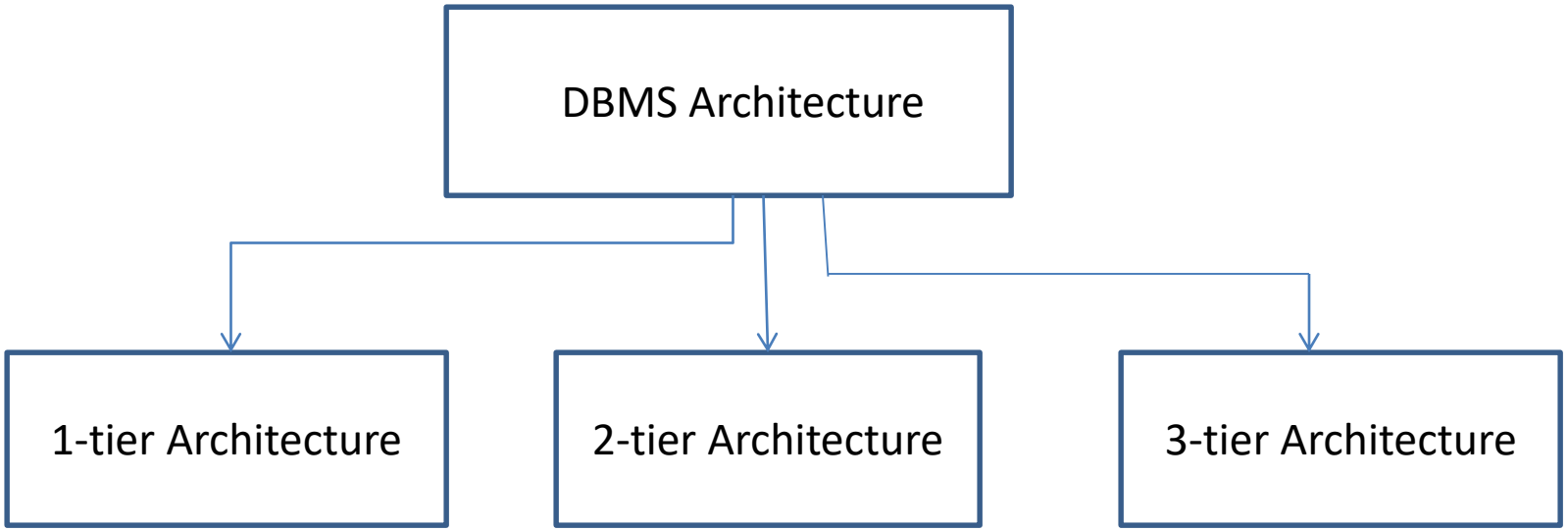


Logical Data Independence

Physical Data Independence

DBMS Architecture

- Database Architecture can be seen as a single tier or multitier.

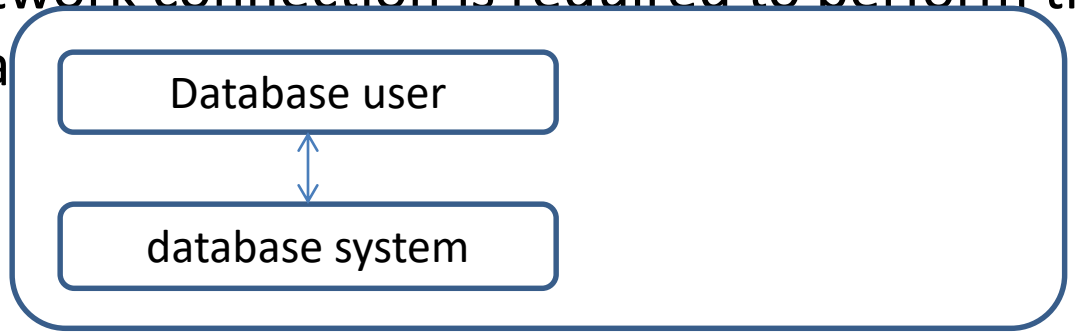


Single or 1-tier Architecture

- In a single or 1-tier architecture, the database is directly available to the user. The client, server and database all reside on the same machine i.e. the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It does not provide a handy tool for end users.
- No network connection is required to perform the action on the database.

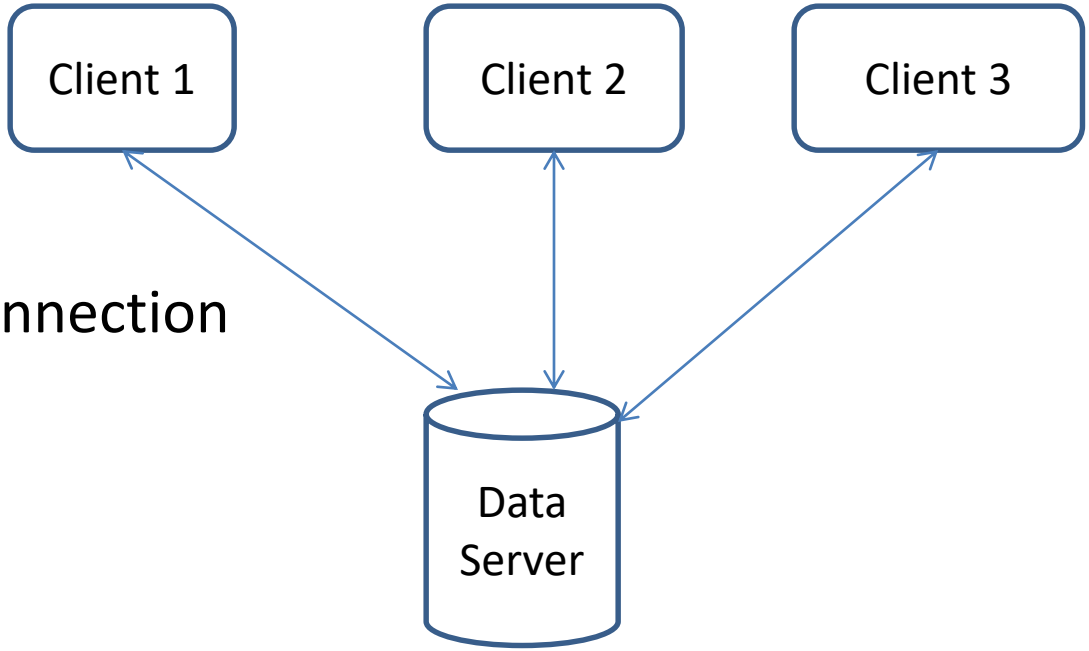
1-tier architecture is used

- ✓ In a single or 1-tier architecture, the database is directly available to the user. The client, server and database all reside on the same machine i.e. the user can directly sit on the DBMS and uses it.
- ✓ Any changes done here will directly be done on the database itself. It does not provide a handy tool for end users.
- ✓ No network connection is required to perform the action on the databa



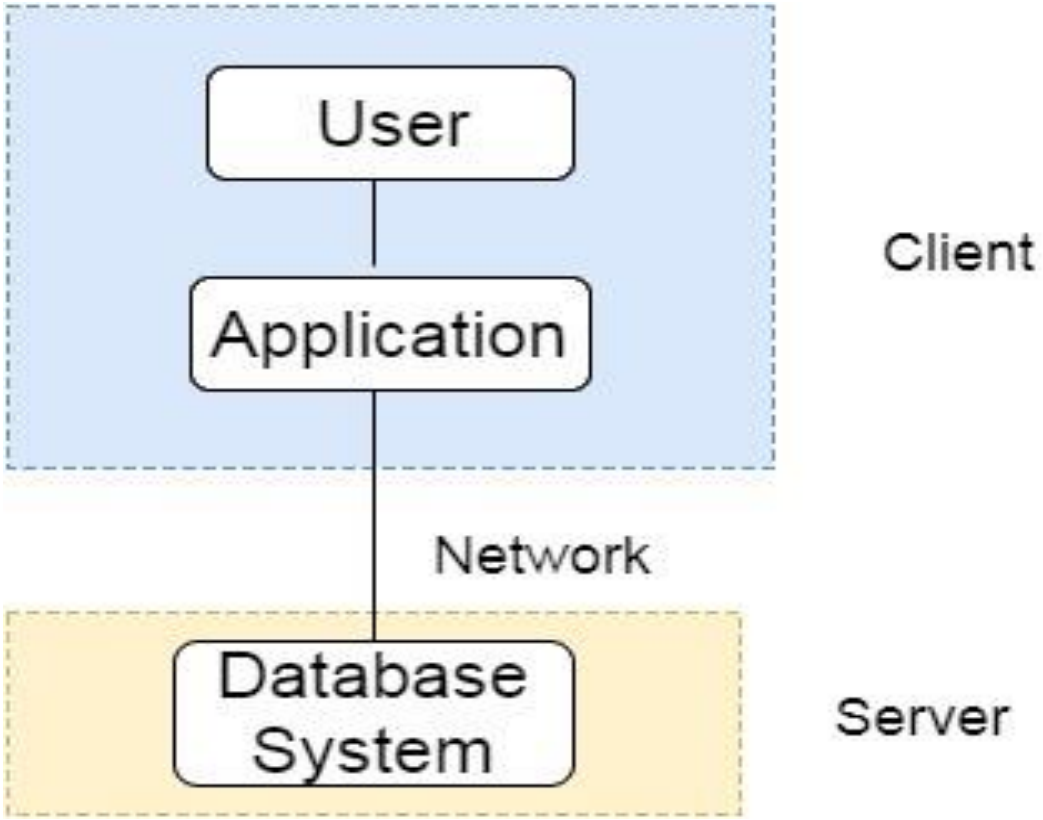
1-tier architecture

2-tier architecture



- It is same as the client server.

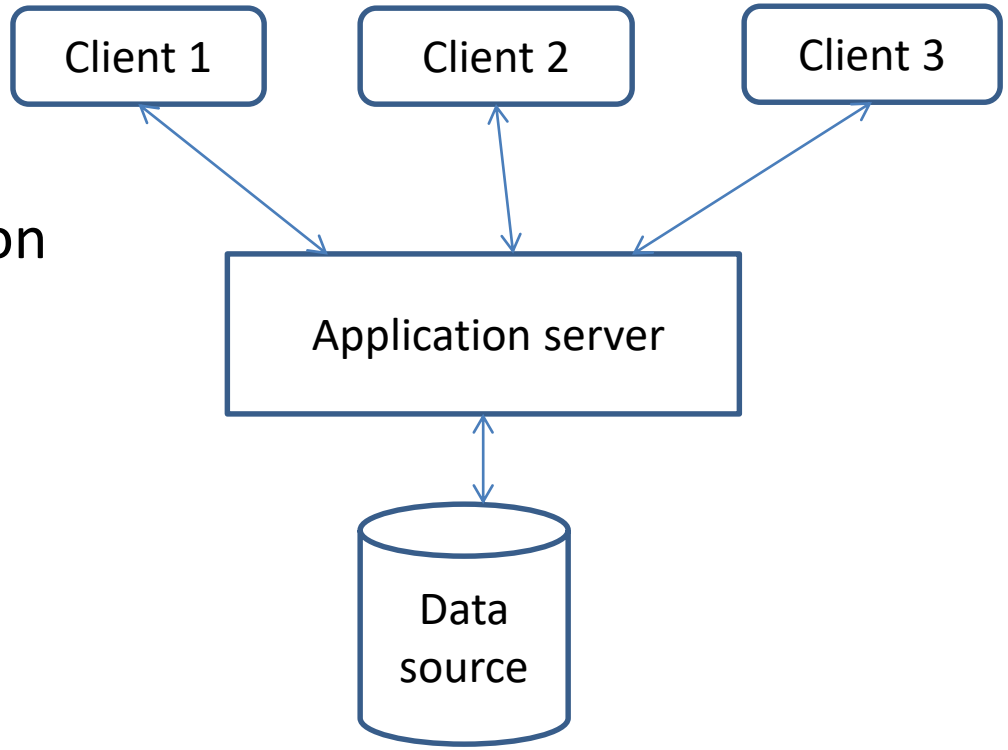
2-Tier Architecture



2-tier architecture

- ✓ In the 2-tier architecture, applications on the client end can directly communicate with the database at the server side.
- ✓ An Application Programming Interface(APIs) like ODBC(Open Database Connectivity) and JDBC(Java Database Connectivity) are used by client side programs to call the DBMS.
- ✓ The user interface and application programs are run on the client side.
- ✓ To communicate with the DBMS, client side application established a connection with the server side.
- ✓ The server side responsible to provide the functionalities like query processing and transaction management.

3-tier architecture



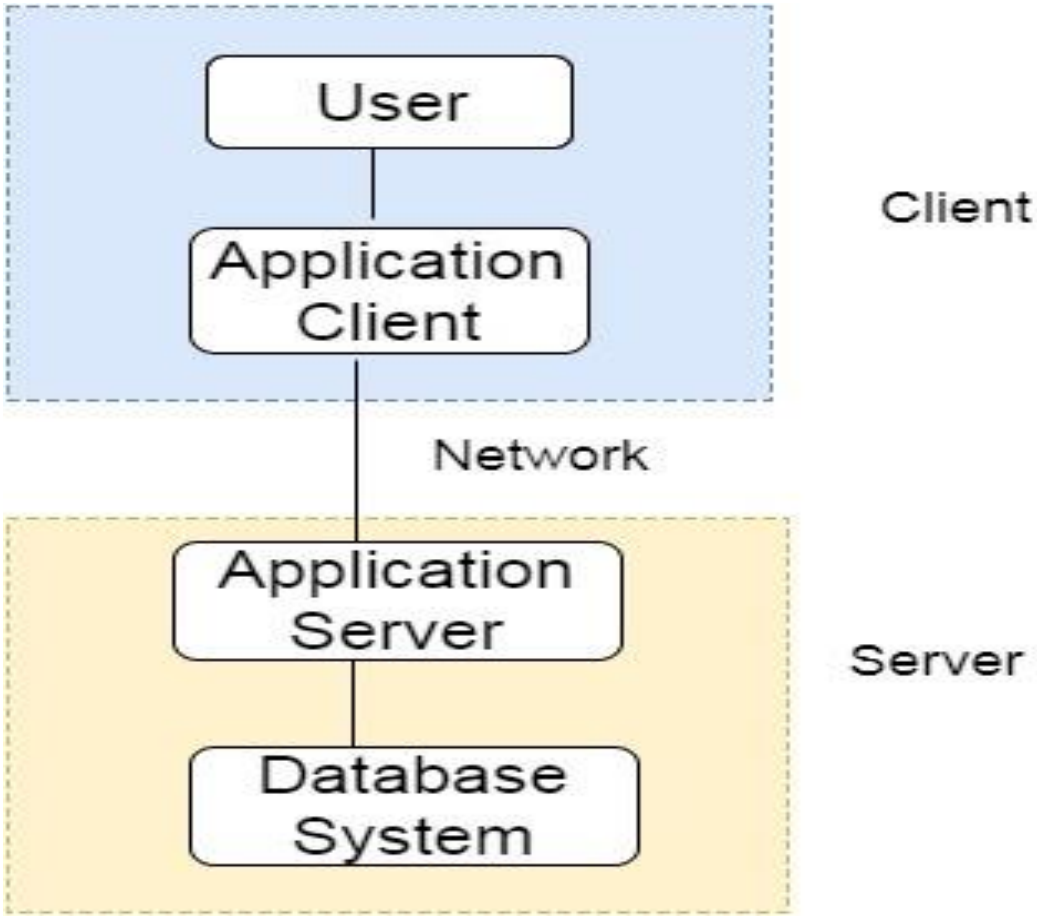
Client Layer

Client Application

Business Layer

Data Layer

3- Tier Architecture



3-tier architecture

- ✓ 3-tier architecture contains another layer of application server between the client and database server.
- ✓ In this architecture, client can't directly communicate with the database server.
- ✓ The application on the client end interacts with an application server which further communicates with the database server and then query processing and transaction management takes place.
- ✓ The intermediate layer of application server acts as a medium for exchange of data between the database server and client.

Thank you