# Week 03 Chapter Two Database System Concepts and Architecture

### **Outline**

- Data Model
- Schema and Instances
- Three-Schema Architecture and Data Independence
- Classification of DBMS
- Database Languages and Interfaces
- Database Systems Environment

## **Data Model**

 A collection of concepts that can be used to describe the structure of the database.

## **Categories of Data Model**

- Conceptual/High Level Data Model
- Representational/ Implementation Data Model
- Physical / Low Level Data Model

## **Self Describing Data Model**

```
C:\test.xml
                                            0 + 0
 <?xml version="1.0"?>
- <List>
   - <Student id="101">
        <Name>John</Name>
        <Place>London</Place>
     </Student>
   - <Student id="102">
        <Name>Peter</Name>
        <Place>Sydney</Place>
     </Student>
   - <Student id="103">
        <Name>Mary</Name>
        <Place>Frankfurt</Place>
     </Student>
 </List>
```

## Not Only SQL Data Model

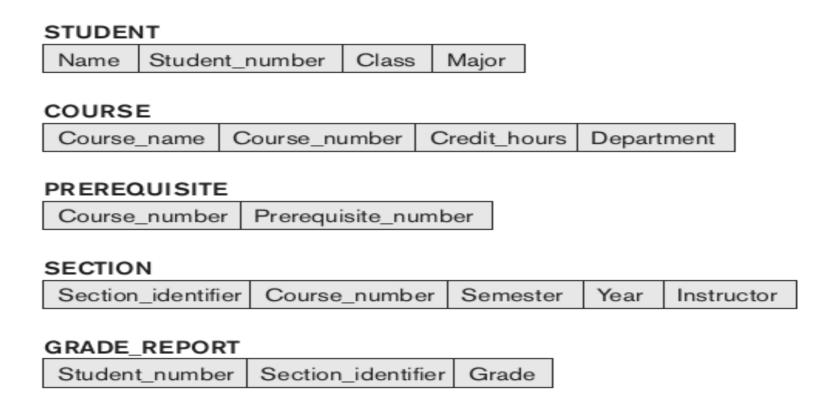
#### Relational Database

Student_Id	Student_Name	Age	College
1001	Chaitanya	30	Beginnersbook
1002	Steve	29	Beginnersbook
1003	Negan	28	Beginnersbook

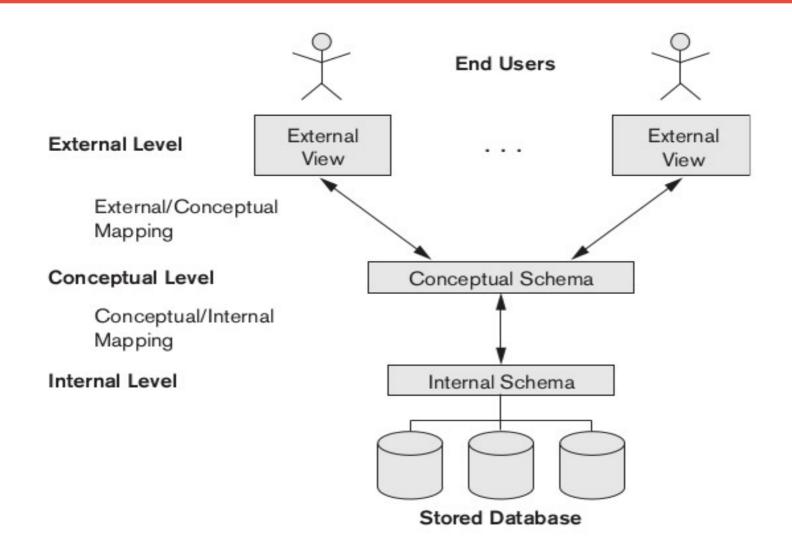
```
MongoDB
"_id": ObjectId("....."),
"Student_Id": 1001,
"Student_Name": "Chaitanya",
"Age": 30,
"College": "Beginnersbook"
"_id": ObjectId("....."),
"Student_Id": 1002,
"Student_Name": "Steve",
"Age": 29,
"College": "Beginnersbook"
"_id": ObjectId("....."),
"Student_Id": 1003,
"Student_Name": "Negan",
"Age": 28,
"College": "Beginnersbook"
```

## Schema, Instances, and Database State

Description of a database



## **Three-Schema Architecture**



## **Three Schema Architecture**

Defines DBMS schemas at **three** levels:

**Internal schema** at the internal level to describe physical storage structures and access paths (e.g indexes).

Typically uses a **physical** data model.

**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.

**External schemas** at the external level to describe the various user views.

Usually uses the same data model as the conceptual schema.

## **Data Independence**

#### **Logical Data Independence:**

The capacity to change the conceptual schema without having to change the external schemas and their associated application programs.

#### **Physical Data Independence:**

The capacity to change the internal schema without having to change the conceptual schema.

For example, the internal schema may be changed when certain file structures are reorganized or new indexes are created to improve database performance

## **DBMS Languages**

Data Definition Language (DDL)

Data Manipulation Language (DML)

High-Level or Non-procedural Languages: These include the relational language SQL May be used in a standalone way or may be embedded in a programming language Low Level or Procedural Languages:

These must be embedded in a programming language

## **Reading Assignment**

Topic 2.3

## **Database Systems Environment**

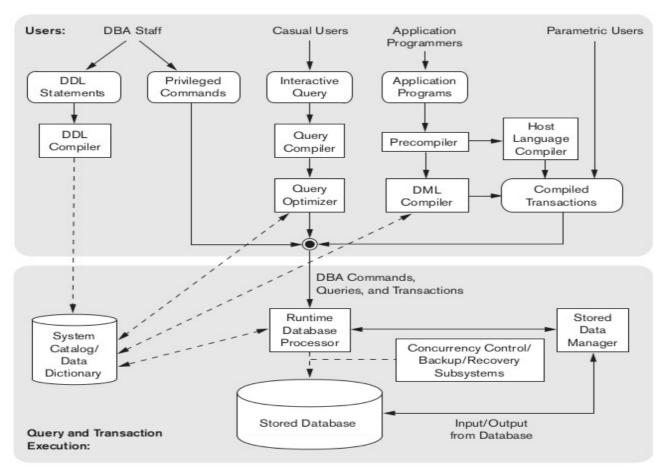
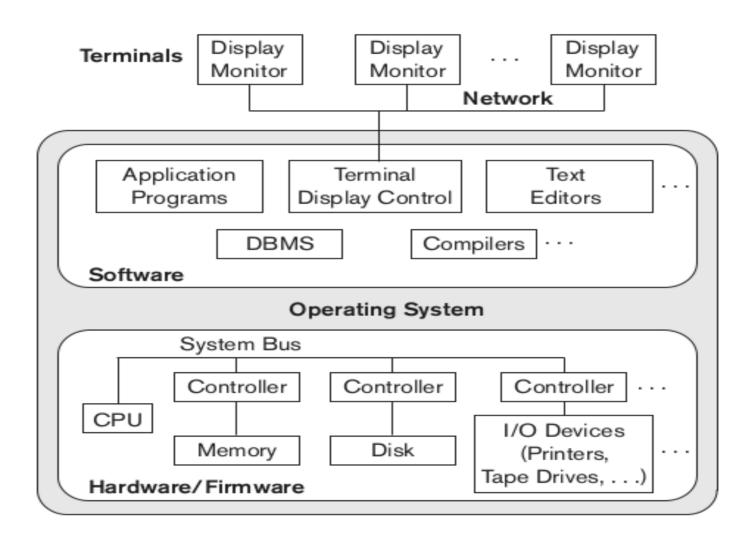


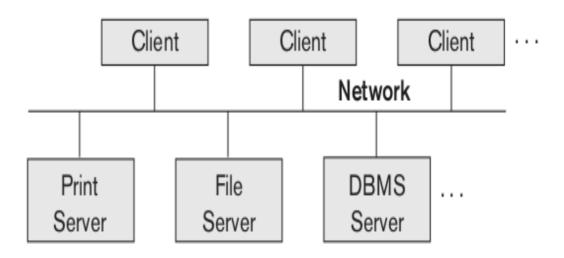
Figure 2.3
Component modules of a DBMS and their interactions.

## **Centralized DBMSs Architecture**



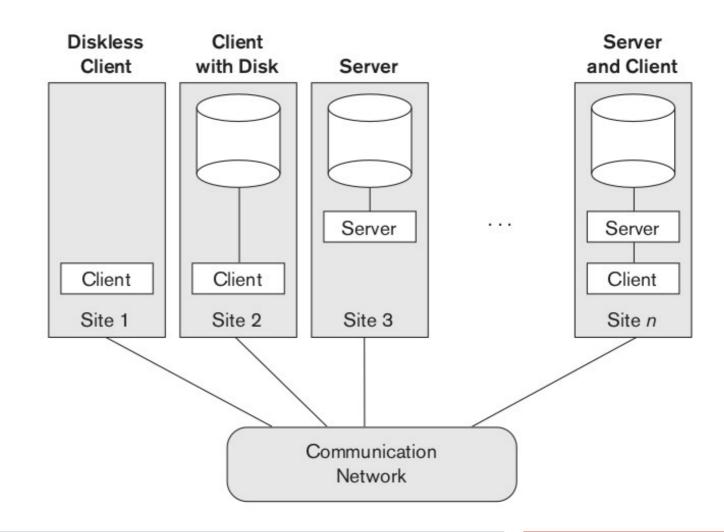
## **Client-Server Architectures**

## Figure 2.5 Logical two-tier client/server architecture.

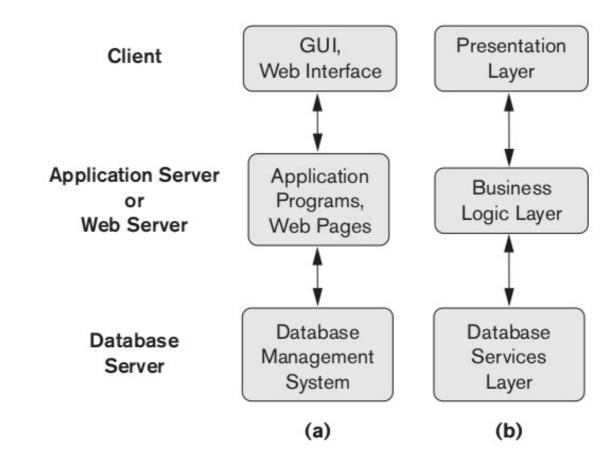


## **Client-Server Architectures**

## Figure 2.6 Physical two-tier client/server architecture.



## **Client-Server Architectures**



#### Figure 2.7

Logical three-tier client/server architecture, with a couple of commonly used nomenclatures. Questions ???