**Data Base System**
**Week 3rd Topics**

- Schemas, Instances, and Database State
- ANSI SPARC Architecture
- Mapping
- Data Independence
- CENTRALIZED AND CLIENT/SERVER ARCHITECTURES FOR DBMSS
- CLASSIFICATION OF DATABASE MANAGEMENT SYSTEMS

# Data Base System
# Week 3rd Topics

- **Database Schema**

➢The structure which contains description of objects created by the user e.g. views, tables etc

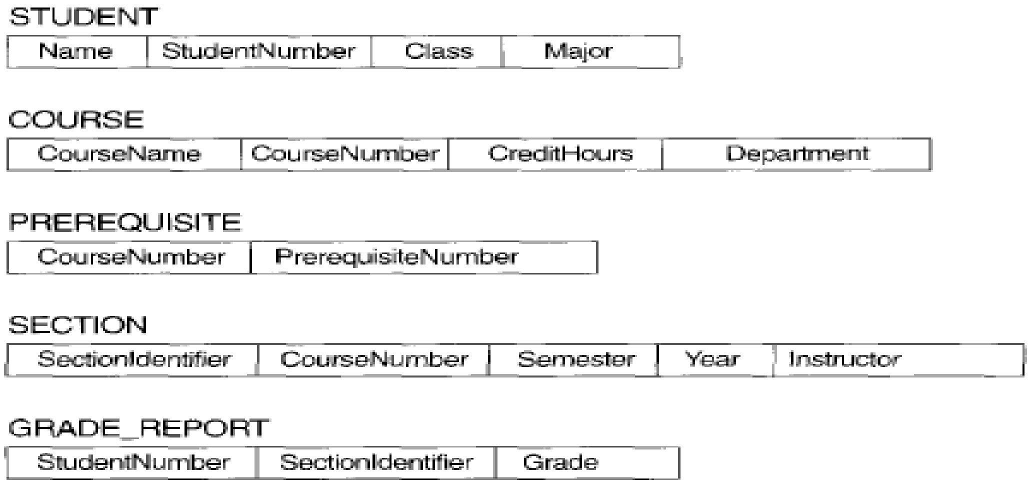
➢Which is specified during database design and is not expected to change frequently

- **schema diagram**

the diagram displays the structure of each record type but not the actual instances of records

- **schema construct**

We call each object in the schema-such as STUDENT or COURSE



STUDENT

| Name | StudentNumber | Class | Major |
|---|---|---|---|

COURSE

| CourseName | CourseNumber | CreditHours | Department |
|---|---|---|---|

PREREQUISITE

| CourseNumber | PrerequisiteNumber |
|---|---|

SECTION

| SectionIdentifier | CourseNumber | Semester | Year | Instructor |
|---|---|---|---|---|

GRADE_REPORT

| StudentNumber | SectionIdentifier | Grade |
|---|---|---|

**FIGURE 2.1** Schema diagram for the database in Figure 1.2.

# Data Base System
# Week 3rd Topics

- The permanent structure of DB is called ***intension of DB*** or ***database schema***
- The data stored at a given time is called extension of ***DB or DB instance***
- The extension of DB is performed after the intension of DB has been finalized
- The extension of DB is performed according to the rules defined in the intension

➢**Initial State**

We get the initial state of the database when the database is first populated or loaded with the initial data

➢**current state**

every time an update operation is applied to the database, we get another database state. At any point in time, the database has a current state

➢**valid state**

a state that satisfies the structure and constraints specified in the schema

The schema is sometimes called the ***intension***, and a database state an ***extension*** of the schema.

➢**schema evolution**

For example, we may decide that another data item needs to be stored for each record in a file, such as adding the DateOfBirth to the STUDENT schema

**Types**

1. **External Schema**
2. **Conceptual Schema**
3. **Internal Schema**

# Data Base System
# Week 3rd Topics

## ANSI SPARC Architecture

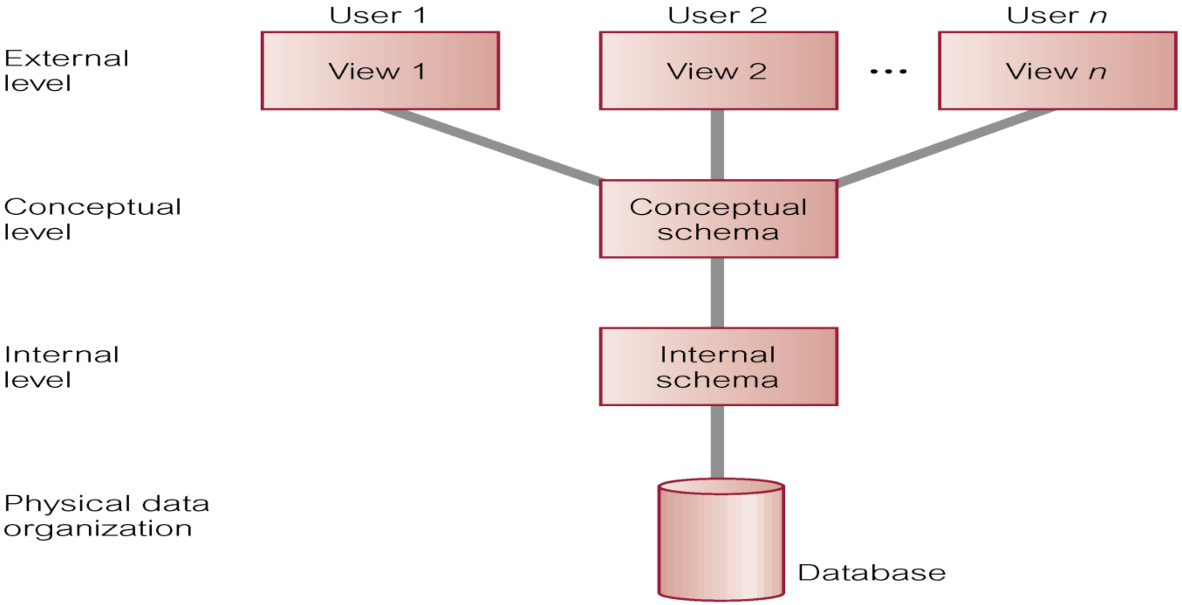
- The Standard Planning and Requirements Committee of American National Standards Institute committee on Computers and Information Processing -1975
- The basis for modern database architecture i.e. DB can be viewed at 3 levels depicted by three models \ schemas

## Objectives of Three-Level Architecture

- All users should be able to access same data.
- A user's view is immune to changes made in other views.
- Users should not need to know physical database storage details.
- DBA should be able to change database storage structures without affecting the users' views.
- Internal structure of database should be unaffected by changes to physical aspects of storage.
- DBA should be able to change conceptual structure of database without affecting all users

# Data Base System
# Week 3rd  Topics

Three-Level Schema



# Data Base System
# Week 3rd Topics

## ANSI SPARC Architecture

1. **External Level**
   - Users’ view of the database.
   - Describes that part of database that is relevant to a particular user.
   - Also Called User View/ Human Interface
   - User will be either application programmer or end user
   - All user’s will see only a part of DB not whole DB
   - User’s views are combined, defined and formatted and forwarded to conceptual level for processing
2. **Conceptual Level**
   - Also called Global or Community view of the database.
   - Describes what data is stored in database and relationships among the data
   - This Level contains logical structure of DB as seen by DBA
   - No description of storage and space consideration e.g. no bytes occupied
   - Constraints on data e.g. name will not be null
   - Semantic info about data e.g. admission before birth
   - Security and integrity e.g. what users are authorized for DB

# Data Base System
# Week 3rd Topics

## ANSI SPARC Architecture

### 3. Internal Level

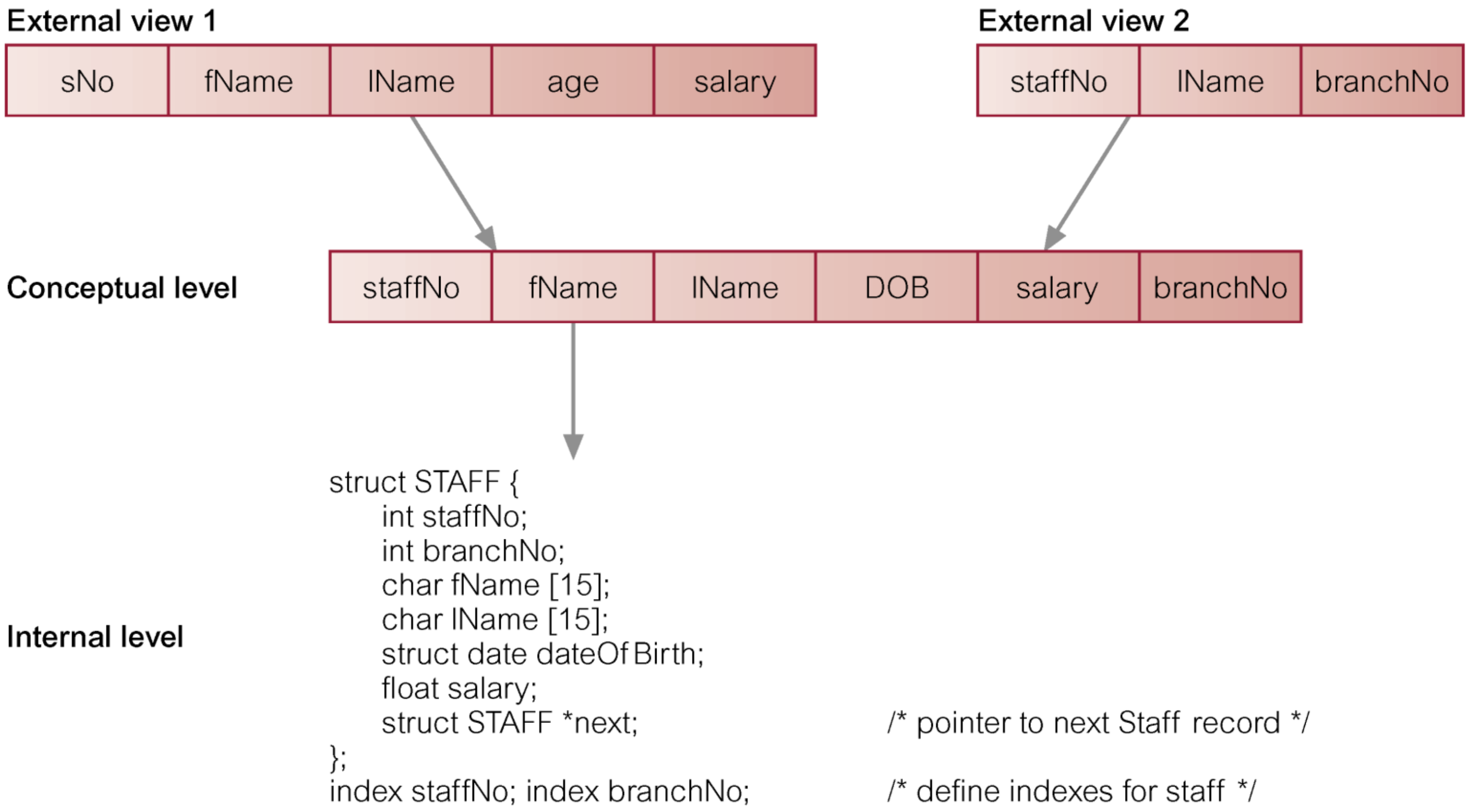
- Responsible to store data in the database.
- Describes how the data is stored in DB by DBMS on computers
- File organization, access methods, record placement, data compression and encryption all techniques are provided by internal level
- defines the methods by which the records are linked i.e. relationship among entities
- The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database

**Physical Data Model**

- Last Level of three level architecture
- Description of physical devices used for grouping of devices
- Addressing techniques manage by OS under the direction of DBMS

# Data Base System
# Week 3rd Topics

## ANSI SPARC Architecture



External view 1

sNo | fName | lName | age | salary

External view 2

staffNo | lName | branchNo

Conceptual level

staffNo | fName | lName | DOB | salary | branchNo

Internal level

```
struct STAFF {
      int staffNo;
      int branchNo;
      char fName [15];
      char lName [15];
      struct date dateOf Birth;
      float salary;
      struct STAFF *next;                  /* pointer to next Staff record */
};
index staffNo; index branchNo;             /* define indexes for staff */
```

# Data Base System
# Week 3rd Topics

**Mapping**

- The process of performing request and retrieving results between levels are called mapping

- These mappings may be time-consuming, so some DBMSs-especially those that are meant to support small databases-do not support external views

**Types**

1. **External / Conceptual Mapping**
2. **Conceptual / Internal Mapping**

**Data Independence**

- The separation of data and application program is called Data Independence

- It is an important feature of DBMS and advantage of three level architecture

- Accomplished when the schema is changed at some level the schema at the next higher level remains unchanged

- Enable the user to change the structure of DB without changing application program

# Data Base System
# Week 3$^{rd}$ Topics

### Types of Data Independence

1. **Logical Data Independence**

Enables the user to change the conceptual level without having to change the external level or application program i.e. to expand the database

- Adding or removal of entities
- Adding a file to DB
- Adding new field to the file
- Changing the type of a field etc
- to change constraints

2. **Physical Data Independence**

➢ Enables the user to change the internal level without having to change the conceptual level or application program or external schemas

- Changing the organization or storage structure
- Using different storage devices
- by creating additional access structures-to improve the performance of retrieval or update

# Data Base System
# Week 3rd Topics

## CENTRALIZED AND CLIENT/SERVER

## ARCHITECTURES FOR DBMSS

### 1. Centralized DBMSS Architecture

▪ database systems used these computers in the same way as they had used display terminals, which were connected to the central computer via various types of communications networks

▪ so that the DBMS itself was still a centralized DBMS in which all the DBMS functionality, application program execution, and user interface processing were carried out on one machine

### 2. Basic Client/Server Architectures

▪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, and other equipment are connected via a network

▪e.g. file server that maintains the files of the client

▪printer server by being connected to various printers; thereafter, all print requests by the clients are forwarded to this machine

▪The **client machines** provide the user with the appropriate interfaces to utilize these servers, as well as with local processing power to run local applications

▪A **server** is a machine that can provide services to the client machines, such as file access, printing, archiving, or database access.

# Data Base System
# Week 3rd Topics

### a) Two-Tier Client/Server Architectures for DBMSS

▪In such a client/server architecture, the user interface programs and application programs can run on the client side.

▪When DBMS access is required, the program establishes a connection to the DBMS (which is on the server side)

▪once the connection is created, the client program can communicate with the DBMS

▪A standard called Open Database Connectivity (ODBC) provides an application Programming interface (API), which allows client-side programs to call the DBMS,

▪client program can actually connect to several RDBMSs and send query and transaction requests using the ODBC API, (Start menu → control panel → Administrative tools →ODBC (32 /64))

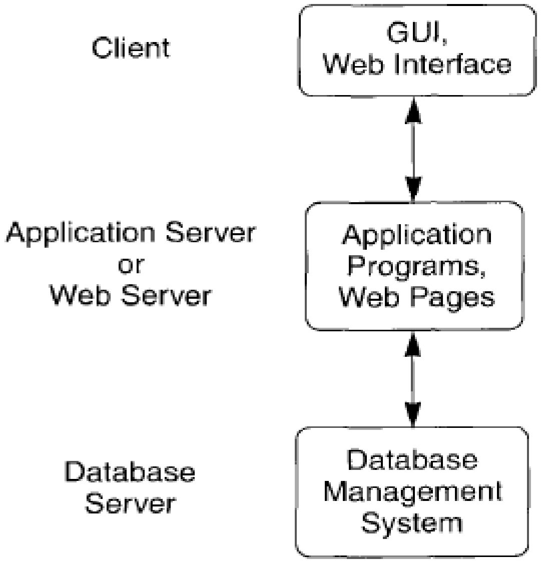
▪which are then processed at the server sites. Any query results are sent back to the client program, which can process or display the results as needed

▪In such a client/server architecture, the server has been called a data server, because it provides data in disk pages to the client.

# Data Base System
# Week 3rd Topics

**b) Three-Tier Client/Server Architectures for Web Applications**

- Many Web applications use an architecture called the three-tier architecture
- Which adds an intermediate layer between the client and the database server,
- This intermediate layer or middle tier is sometimes called the application server and sometimes the Web server, depending on the application


Client
GUI, Web Interface
Application Server or Web Server
Application Programs, Web Pages
Database Server
Database Management System

# Data Base System
# Week 3rd Topics

- This server plays an intermediary role by storing business rules (procedures or constraints) that are used to access data from the database server.
- It can also improve database security by checking a client's credentials before forwarding a request to the database server
- Clients contain GUI interfaces and some additional application-specific business rules.
- The intermediate server accepts requests from the client, processes the request and sends database commands to the database server, and then acts as a conduct for passing (partially) processed data from the database server to the clients, where it may be processed further and filtered to be presented to users in GUI format.

## CLASSIFICATION OF DATABASE MANAGEMENT SYSTEMS

**1. data model**

- ➢ relational data model (main)
- ➢ object data model (not widespread)
- ➢ hierarchical and network data models (older)

**2. number of users**

- ➢ **Single-user systems** support only one user at a time and are mostly used with personal computers.
- ➢ **Multiuser systems**, which include the majority of DBMSs, support multiple users concurrently.

# Data Base System
# Week 3rd Topics

## CLASSIFICATION OF DATABASE MANAGEMENT SYSTEMS

**3. number of sites**

➢ A DBMS is **centralized** if the data is stored at a single computer site

➢ A **distributed DBMS** (DDBMS) 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 multiple sites

➢ A recent trend is to develop software to access several autonomous preexisting databases stored under **heterogeneous** DBMSs.

**4. cost of the DBMS**

DBMS packages cost between $10,000 and $100,000

**5. types of access path**

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

**Finally**

➢**special purpose.:** when performance is a primary consideration, a special-purpose DBMS can be designed and built for a specific application

➢Many airline reservations and telephone directory systems developed in the past are special purpose DBMS