Databases Environment

Chapter #2
Imran Khan
FCS, IBA

Lecture 2 - Objectives

- Purpose of three-level database architecture.
- Contents of external, conceptual, and internal levels.
- Purpose of external/conceptual and conceptual/internal mappings.
- Meaning of logical and physical data independence.
- Distinction between DDL and DML.
- A classification of data models.

Lecture 2 - Objectives

- Purpose/importance of conceptual modelling.
- Typical functions and services a DBMS should provide.
- Software components of a DBMS.
- Meaning of client–server architecture and advantages of this type of architecture for a DBMS.
- Function and importance of the system catalog.

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.

Objectives of Three-Level Architecture

- 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.
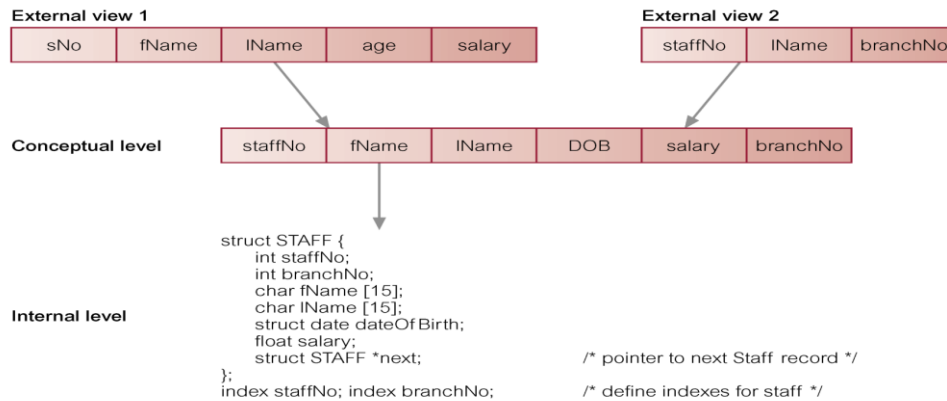
ANSI-SPARC Three-level Architecture
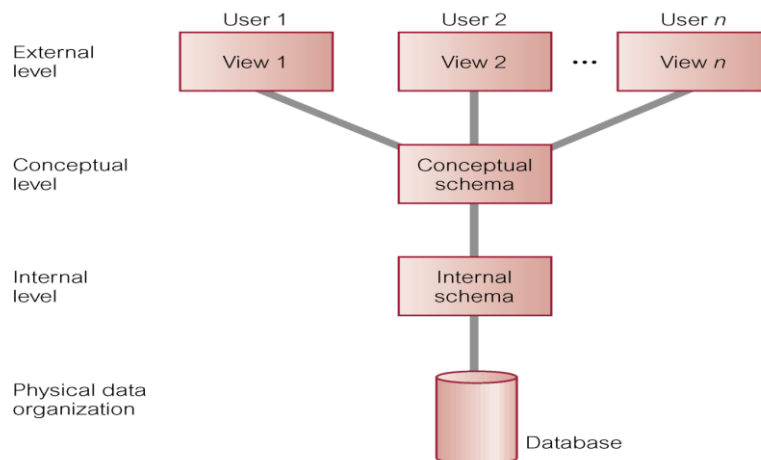
ANSI-SPARC Three-level Architecture

- External Level
  - Users' view of the database.
  - Describes that part of database that is relevant to a particular user.
- Conceptual Level
  - Community view of the database.
  - Describes what data is stored in database and relationships among the data.

ANSI-SPARC Three-level Architecture

- Internal Level
  – Physical representation of the database on the computer.
  – Describes how the data is stored in the database.

Differences between Three Levels of ANSI-SPARC Architecture

External view 1						External view 2		
sNo	fName	lName	age	salary		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 Independence

- Logical Data Independence
  - Refers to immunity of external schemas to changes in conceptual schema.
  - Conceptual schema changes (e.g. addition/removal of entities)
  - Should not require changes to external schema or rewrites of application programs.

Data Independence

- Physical Data Independence
  - Refers to immunity of conceptual schema to changes in the internal schema.
  - Internal schema changes (e.g. using different file organizations, storage structures/devices).
  - Should not require change to conceptual or external schemas.

Data Independence and the ANSI-SPARC Three-level Architecture
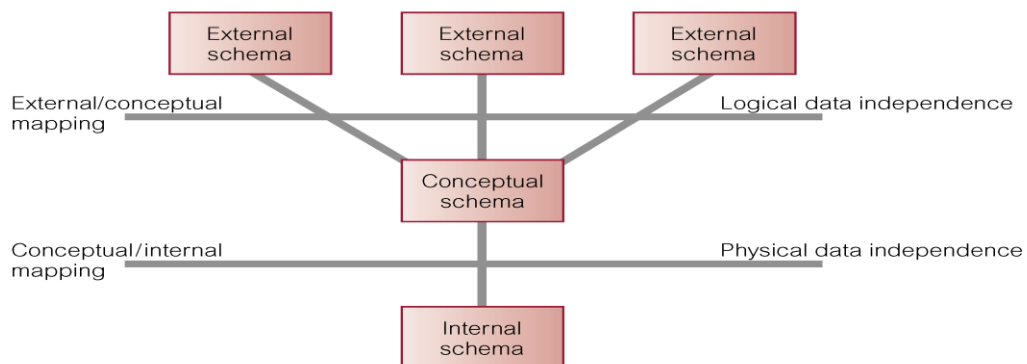
Database Languages

- Data Definition Language (DDL)
  - Allows the DBA or user to describe and name entities, attributes, and relationships required for the application
  - plus any associated integrity and security constraints.

Database Languages

- Data Manipulation Language (DML)
  - Provides basic data manipulation operations on data held in the database.
- Procedural DML
  - allows user to tell system exactly how to manipulate data.
- Non-Procedural DML
  - allows user to state what data is needed rather than how it is to be retrieved.

Database Languages

- Fourth Generation Language (4GL)
  - Query Languages
  - Forms Generators
  - Report Generators
  - Graphics Generators
  - Application Generators

Data Model

- Integrated collection of concepts for describing data, relationships between data, and constraints on the data in an organization.

- Data Model comprises:
  - A structural part
  - A manipulative part
  - Possibly a set of integrity rules

Data Model

- Purpose
  - To represent data in an understandable way.

- Categories of data models include:
  - Object-based
  - Record-based
  - Physical

Data Models

- Object-based Data Models
  - Entity-Relationship
  - Semantic
  - Functional
  - Object-Oriented
- Record-based Data Models
  - Relational Data Model
  - Network Data Model
  - Hierarchical Data Model
- Physical Data Models

Conceptual modelling

- Conceptual schema is the core of a system supporting all user views.
- Should be complete and accurate representation of an organization's data requirements.
- Conceptual modelling is process of developing a model of information use that is independent of implementation details.
- Result is a conceptual data model.

Functions of a DBMS

- Data Storage, Retrieval, and Update.

- A User-Accessible Catalog.
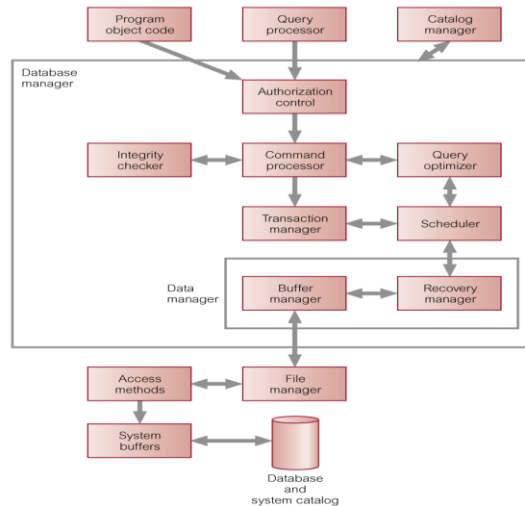
- Transaction Support.

- Concurrency Control Services.

- Recovery Services.

Functions of a DBMS

- Authorization Services.

- Support for Data Communication.

- Integrity Services.

- Services to Promote Data Independence.
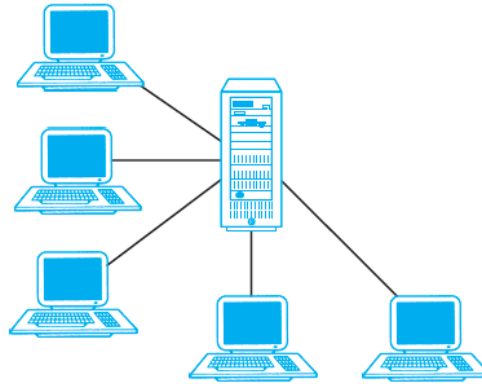
- Utility Services.

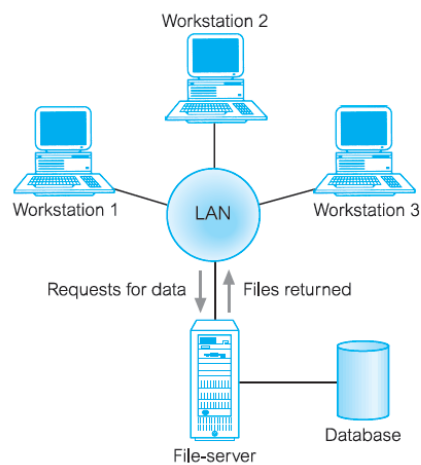# Components of a DBMS
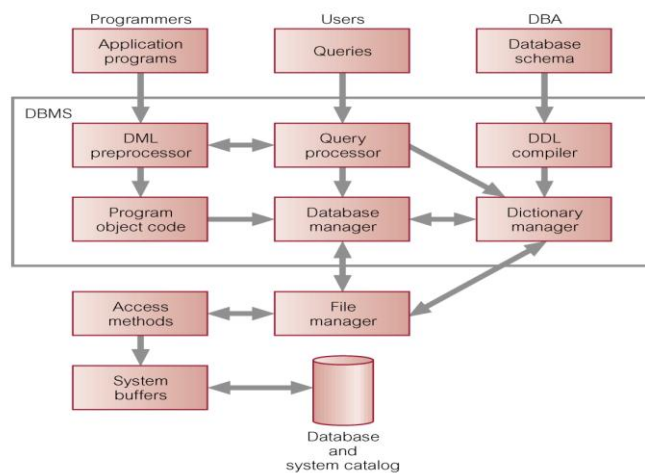
Components of Database Manager (DM)

Multi-User DBMS Architectures

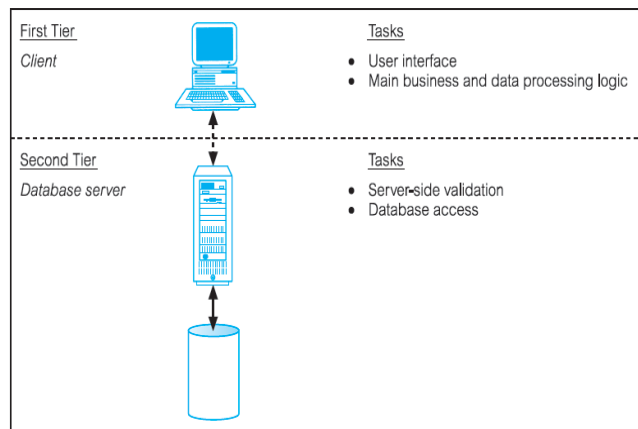- Teleprocessing

- File-server

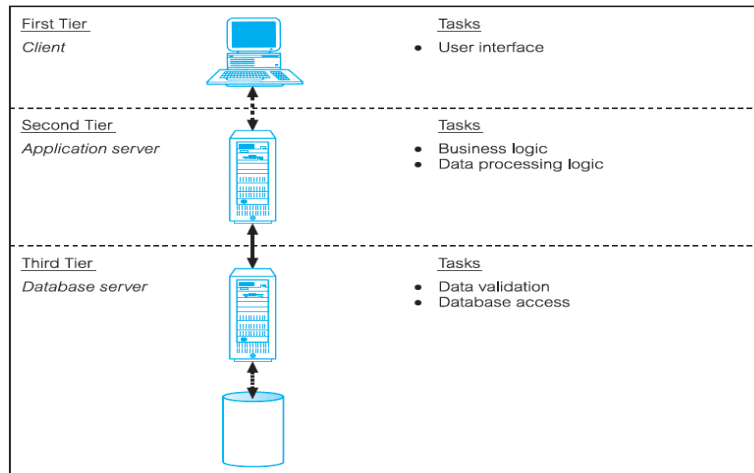- Client-server

Teleprocessing

File Server

Client-server

- Server holds the database and the DBMS.

- Client manages user interface and runs applications.

- Advantages include:
    - wider access to existing databases
    - increased performance
    - possible reduction in hardware costs
    - reduction in communication costs
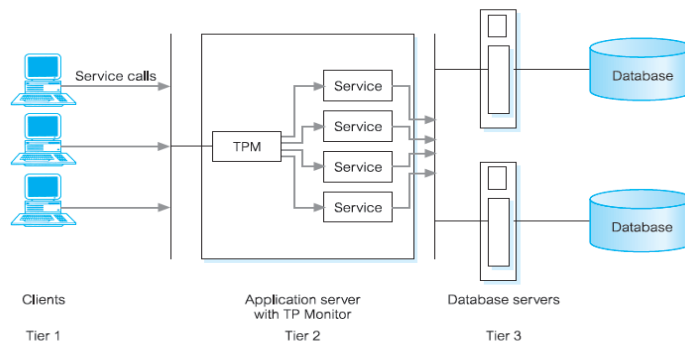   - increased consistency.

Client-server Architecture
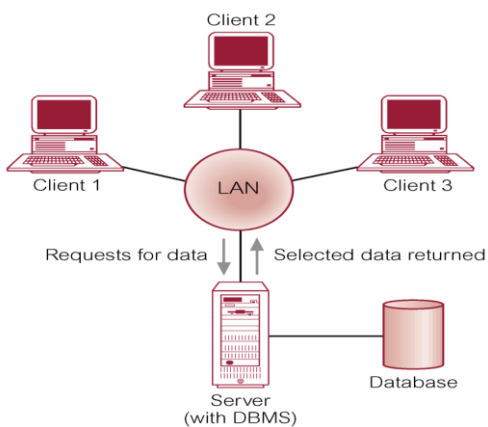
Client-server Architecture

Transaction Processing Monitors

- A program that controls data transfer between clients and servers in order to provide a consistent environment, particularly for online transaction processing (OLTP).

System Catalog

- Repository of information (metadata) describing the data in the database.
- Typically stores:
    - Names of authorized users.
    - Names of data items in the database.
    - Constraints on each data item.
    - Data items accessible by a user and the type of access.
- Used by modules such as Authorization Control and Integrity Checker.