DBMS ARCHITECTURE

View of Data (Three Schema Architecture)

The major purpose of DBMS is to provide users with an abstract view of the data. That is, the system hides certain details of how the data is stored and maintained. To simplify user interaction with the system, abstraction is applied through several levels of abstraction. The main objective of three level architecture is to enable multiple users to access the same data with a personalized view while storing the underlying data only once.

Physical Level / Internal Level

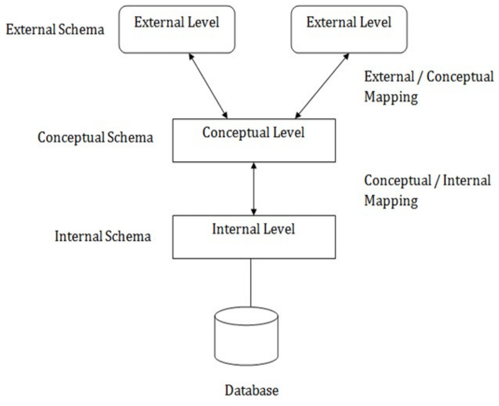
- The lowest level of abstraction describes how the data are stored.
- Low-level data structures used.
- It has Physical schema which describes physical storage structure of DB.
- Talks about: Storage allocation (N-ary tree etc.), Data compression G encryption etc.
- **Goal:** We must define algorithms that allow efficient access to data.

Logical Level / Conceptual Level:

- The conceptual schema describes the design of a database at the conceptual level, describes what data are stored in DB, and what relationships exist among those data.
- User at logical level does not need to be aware about physical-level structures.
- DBA, who must decide what information to keep in the DB use the logical level of abstraction.
- **Goal:** ease to use.

View Level / External Level:

- Highest level of abstraction aims to simplify users' interaction with the system by providing different view to different end-user.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- At views also provide a security mechanism to prevent users from accessing certain parts of DB.



Instances and Schemas

The collection of information stored in the DB at a particular moment is called an instance of DB. The overall design of the DB is called the DB schema. Schema is structural description of data. Schema doesn't change frequently. Data may change frequently. DB schema corresponds to the variable declarations (along with type) in a program. We have 3 types of Schemas: **Physical, Logical,** several **View Schemas** called Sub-Schemas.

Logical schema is most important in terms of its effect on application programs, as programmers construct apps by using logical schema. Physical data independence, physical schema change should not affect logical schema/application programs.

Data Models

Provides a way to describe the design of a DB at logical level. Underlying the structure of the DB is the Data Model; a collection of conceptual tools for describing data, data relationships, data semantics G consistency constraints.

E.g., ER model, Relational Model, object-oriented model, object-relational data model etc.

Database Languages

Data Definition Language (DDL) to specify the database schema. We specify consistency constraints, which must be checked, every time DB is updated.

Data manipulation language (DML) to express database queries and updates. Data manipulation involves: Retrieval of information stored in DB. Insertion of new information into DB. Deletion of information from the DB.

Practically, both language features are present in a single DB language, e.g., SQL language. DDL.

How is Database accessed from Application programs?

Apps (written in host languages, C/C++, Java) interacts with DB. E.g., Banking system's module generating payrolls access DB by executing DML statements from the host language. API is provided to send DML/DDL statements to DB and retrieve the results.

- Open Database Connectivity (ODBC), Microsoft "C".
- Java Database Connectivity (JDBC), Java.

Database Administrator (DBA)

A person who has central control of both the data and the programs that access those data.
Functions of DBA

- Schema Definition
- Storage structure and access methods.
- Schema and physical organization modifications.
- Authorization control.
- Routine maintenance: Periodic backups, Any upgrades.

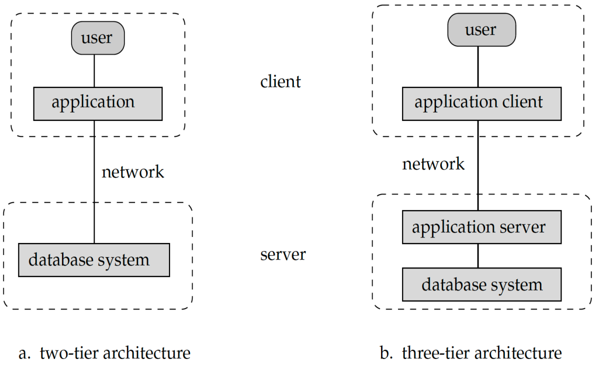
DBMS Application Architectures

Client machines, on which remote DB users work, and server machines on which DB system runs.

T1 Architecture: The client, server and DB all present on the same machine.

T2 Architecture

- App is partitioned into 2-components.
- Client machine, which invokes DB system functionality at server end through query language statements.
- API standards like ODBC G JDBC are used to interact between client and server.



a. two-tier architecture b. three-tier architecture

T3 Architecture

App is partitioned into 3 logical components. Client machine is just a frontend and doesn't contain any direct DB calls. Client machine communicates with App server, and App server communicated with DB system to access data. Business logic, what action to take at that condition is in App server itself. T3 architectures are best for WWW Applications.

Advantages

- **Scalability** due to distributed application servers.
- **Data integrity**, App server acts as a middle layer between client and DB, which minimize the chances of data corruption.
- **Security,** client can't directly access DB, hence it is more secure.