THREE SCHEMA ARCHITECTURE and DATA INDEPENDENCE.

Important characteristics of Database approach

->insulation of data and programs

-> support four multiple user views

->use of catalog to store the data description (schema)

Three schema architecture - " ANSI/SPARC architecture".
$\downarrow$ std for DBMS

main goal of Three schema architecture is to seperate user application and physical database.

Internal level - This level has internal schema-

↳This describes physical storage structure. This describes complete details of data storage.

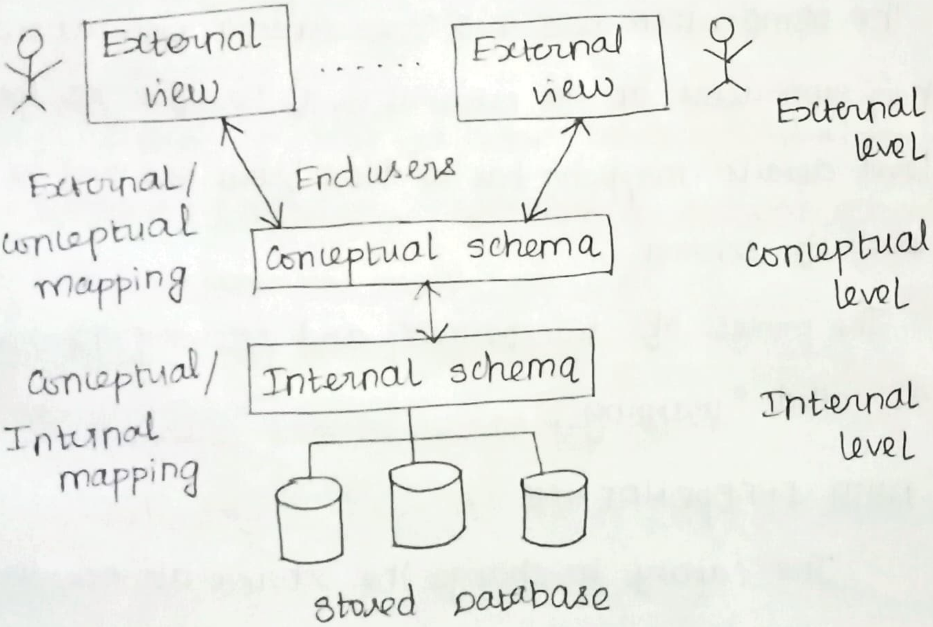


Fig: THREE SCHEMA ARCHITECTURE

Conceptual level: - This level has conceptual schema.

↳ This describes structure of whole database for community of users. This hides the inner details of physical storage structures.

↳ describes entities, data types, relationship, user operations.

External or view level- This describes user view

This describes part of database in which the user is interested. hides rest of database

All the above defined three levels are very important for designing the database DBMS.

Conceptual level information are represeted using relational DBMS like "Oracle uses SQL, Universal Database (UDB), a DBMS from IBM".

⊗

This three schema architecture describes only the description of data. The stored data actually present in the physical level of DB.

The DBMS which uses this three schema architecture, has their own users at an external level. To refer the conceptual level details, mapping has to done from one level to other level of schema.

The process of transforming and receiving the request is called "mapping".

DATA INDEPENDENCE

The capcity to change the schema at one level of a database without having to change the schema at next higher level is called "Data independence".

Types:-

↳ logical Data independence

↳ Physical Data independence.

Logical Data Independence.

↳have to change the conceptual schema without having to change external schema's or application programs

Physical Data Independence

↳have to change the internal schema without having to change the conceptual schema. hence external schema also need not change.

change in internal schema includes file reorganization for enhancing the performance of Data retrieval.

DBMS Language:-

once the DBMS design is completed, then DBMS language is chosen to implement the database.

There is need to mapping of conceptual and internal schemas.

DDL- Data Definition Language

↳used by DBA (Database Administrator) and by DB designers to define conceptual and internal schemas.

↳DDL Compiler process the DDL.

SDL- Storage Definition Language.

↳used to specify internal schema

↳It defines how and where relevant data is stored.

DDL- Describes data and relationships in DB.

VDL- view Definition Language.

↳to specify user's view and their conceptual mapping.

↳In DBMS, SQL plays a role of VDL.

DML- Data Manipulation Language

↳This is used four Data retrieval, insertion, deletion and modification of the data.

-> high level DML- used to specify complex operation

-> low level DML - used to retrieve individual record.

DBMS Interfaces:-

->Menu Based interface far web clients our browsing

-> Forms Based interface

-> Graphical user interface

->Natural language interface

->Speech input and output

->Interface four parametric users

->Interface four DBA.

Note ⓧ

UDB- Universal DataBase - web enabled RDBMS four data warehousing and transaction processing.

DBMS from IBM -

↳IBM DB2 -family of DBMS products, Servers which is developed by IBM.

Database Architecture

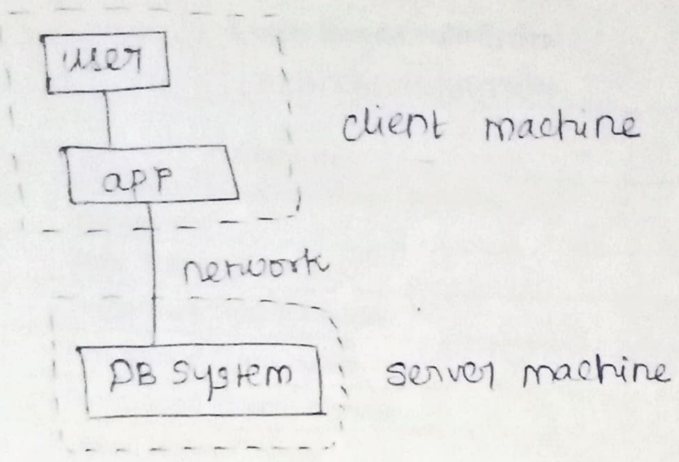
DB architecture can be seen as single tier our multi tier. Yet, there is two and three tier architecture

Two tier: same as client-server model.

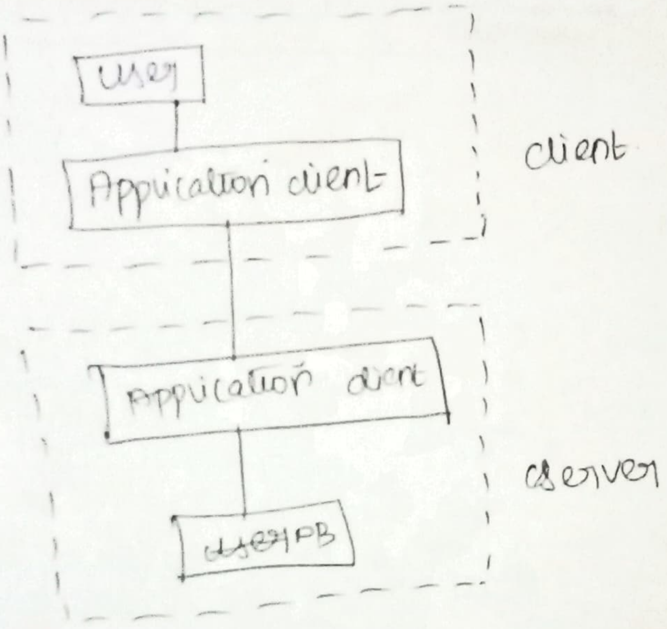
->user i/f and application programs run on client side.

->Server side provides functionalities such as query processing, transaction management.

Two tier architecture



Three tier architecture



-> This layer contains another layer b/w the client and server. here, client cannot able to communicate with the server.

-> Application on client end interacts with an application server which further communications with DB system.

-> here the end user does not have idea of the existence of the application server.