Book → (1) Henry F. Korth
(2) Bipin Desai
(3) Elmasri, S. Navate

Page No. 01
Date: / /

ADBMS [Advance Database Management System]

Transaction :- ACID [Atomicity Consistency Isolation Durability]

Transaction → A transaction is a unit of program execution that access and possibly updates various data items.

Properties of ACID

① Atomicity :→ The database system keeps track of the whole values of any data on which a transaction performs a right operation and if the transaction does not complete its execution, the database system restore the old values to make it appear as though the transaction never executed.

② Consistency :→ Execution of a transaction in isolation that is with no other transaction executing concurrently preserve the consistency of database.

③ Isolation :→ Even though multiple transaction may execute concurrently. The system guarantees that for every pair of transaction Ti and Tj, it appears to Ti that either Tj finish execution before Ti started or Tj start execution after Ti finish.

(4) Durability → Once a transaction complete successfully, all the updates i.e. carried out database persist, even if there is a system failure after the transaction complete execution.

UNIT-1 :→ Database System Concept & Architecture

- → Transactions
- → DBMS & its characteristics
- → DBMS vs Flat File
- → Advantages & Disadvantages
- → Architecture [3 types]

DBMS [Database Managment System]

Database :→ A database is a collection of relative data which represents some aspects of real world. A database system is designed to be build and populated with data for a certain task.

DBMS :→ (1) DBMS is a software for storing and retriving [access] user data considering appropriate security measures.

(2) It consists of a group of programs which manipulate the database.

(3) DBMS accepts the user request for data from an application & instructs the operating system to provide the specific data.

④ It provides an interface between data & software application.

⑤ Help to create own database as per requirement

Characteristics of DBMS

① Provides Security
② Remove Redundancy [Duplicate]
③ Support multiple view of data
④ Sharing of Data
⑤ Multiuser transaction processing
⑥ Allows entities and relation among them to form tables.
⑦ Follows ACID Concept [Atomicity Consistency Isolation Durability]
⑧ Support multi-user environment that allows user to access & manipulate data in parallel.

DBMS vs Flat File

	DBMS	Flat File Management System
①	Multi user access.	It doesn't support multi-user access
②	Design to fulfill the need for small and large business.	It is only limited to smaller DBMS system.
③	Remove Redundancy and Integrity	Redundancy & Integrity issues.
④	Expensive [but in long term total cost of ownership is cheap]	It is cheaper.
⑤	Easy to implement complicated transactions.	No support for complicated transactions.

Advantages Of DBMS

1. DBMS offer a variety of technique to store and retrieve data.
2. DBMS serves as an efficient handler to balance the need of multiple application using the same data.
3. Uniform Administration procedure for the data.
4. Application programmers never exposed to details of data representation and storage.
5. A DBMS uses various powerful functions to store and retrieve the data.
6. DBMS offer data security and integrity.
7. A DBMS schedule concreate access to the data in such a manner that only one user can access the same data at a time.
8. Reduces application development time.

Disadvantages Of DBMS

1. Cost of Hardware and Software is quite high which increase the budget of your organisation.
2. Most database management system are often complex system, so the training for the user to use the DBMS is required.
3. In some organisation, all the data is integrated into a single database which can be damage because of electric failure or database is corrupted on the storage media.

4. Use of the same program at a time by many users sometimes lead to the loss of data.

A DATABASE ARCHITECTURE

1. It is a representation of DBMS design.
2. It helps to design, develop, implement & maintain the DBMS.
3. A DBMS architecture allows dividing the database system into individual components that can be independently modified, changed, replaced, altered.
4. It also helps to understand the components of database.
5. A database stores critical information & helps to access data quickly & securely. Therefore, selecting the correct architecture of DBMS helps in easy & efficient data management.

→ Types of DBMS Architecture

1. 1-tier Architecture
2. 2-tier Architecture
3. 3-tier Architecture

1. 1-tier Architecture → It is the simple architecture of database in which the client, server and database all reside [exist] on the same machine.

Single-tier Architecture

(A) In this architecture, the database is directly available to the user. It means that the users can directly sit on the DBMS and use it.

(B) Any changes done here will directly be done on the database itself. It doesnot provide a handy tool for end users.

(C) The 1-tier architecture is used for development of local applications, where the programmers can directly communicate with ~~data~~database for quick response.

2. 2-Tier Architecture :→

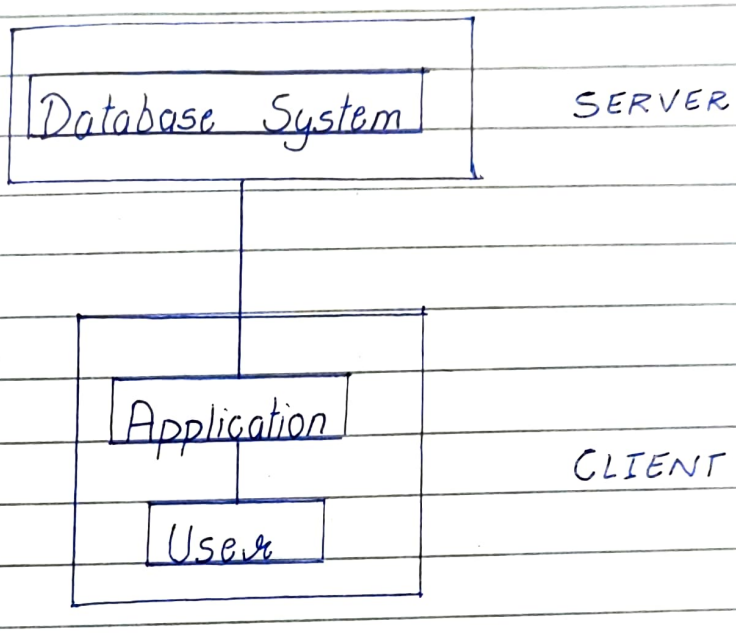
(A) It is similar to basic client server.

(B) Applications on the client end can directly communicate with the database at the server side. For this interaction, API's like ODBC [Object Database Connectivity], JDBC [Java Database Connectivity]

(C) The user interface & application programs are run on the client-side.

(D) The server side is responsible to provide the functionalities like — ① Query Processing
② Transaction Management

(E) To communicate with DBMS, client side application establishes a connection with the server side.



2 Tier - Architecture

3. 3 - Tier Architecture :→

(A) It contains another layer between the client & server.

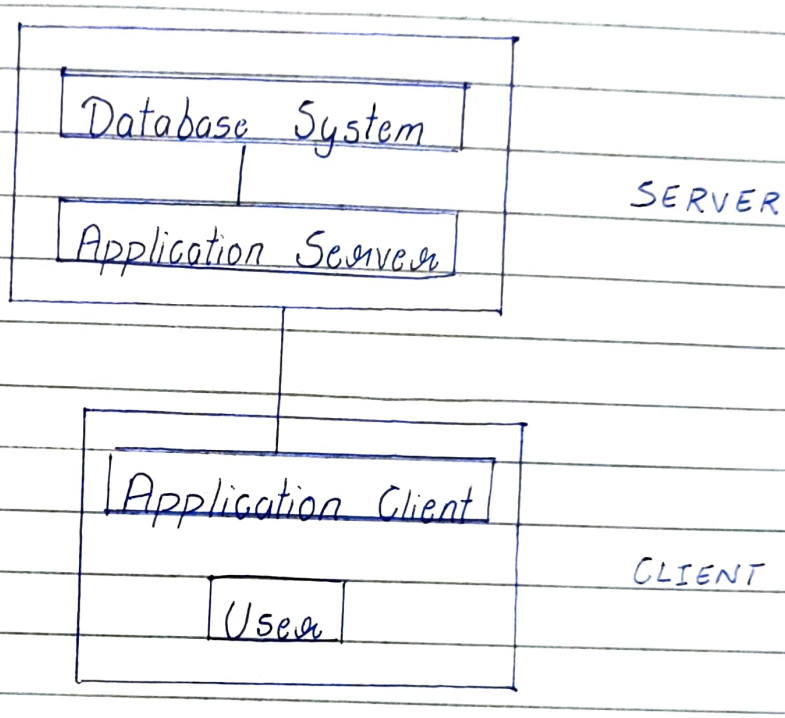
(B) Client cannot directly communicate with the server.

(C) The application on the client-end interacts with an application server which further communicates with the database system.

(D) End user has no idea about the existance of the Database beyond the application server.

(E) The Database also has no idea about any other user beyond the application.

(F) It is used in case of large web applications.



3-Tier Architecture

THREE-SCHEME ARCHITECTURE

→ It is also called three-level architecture.
→ This framework is used to describe the structure of a specific database system.
→ It is also used to seperate the user application & physical database.
→ It contain three-levels. It breaks the database down into 3 different categories-

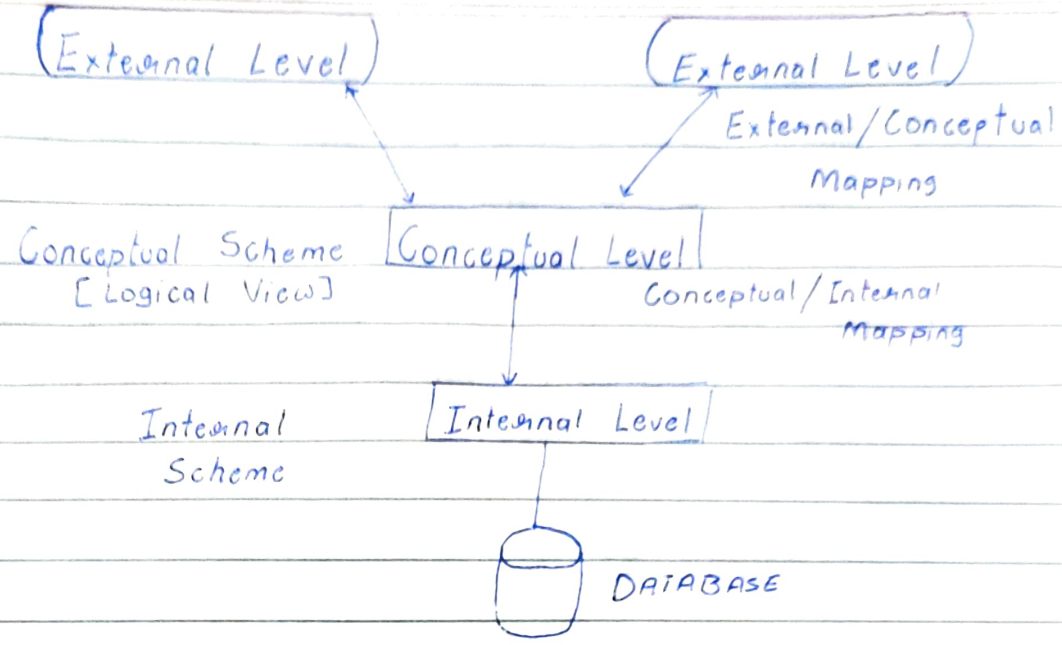


Fig:- Three-Scheme Architecture

1. Internal Level → Internal Level has an internal scheme which describes the physical storage structure of the database.
   → It is also known Physical Scheme.
   → It uses the physical data model. It is used to define that how the data will be stored in a block.
   → The physical level is used to describe complex low level data structure in detail.

2. Conceptual Level

   → The conceptual level describe the design of database. at the conceptual level. It is also known as logical level.
   → The conceptual level describe the structure of whole database.
   → The conceptual level describes what data are to be

stored in database and also describes what relationship exists among those data.

→ In the conceptual level, internal details such as an implementation of data structure are hidden.

→ Programmer and database administrator work at this level together.

3. External Level

→ At the external level, a database contain several schemes that sometimes known as sub-schemes. The sub-schemes is used to describe the different view of database.

→ It is also called view scheme.

→ Each view scheme describe the database path that a particular user group is interested and hides the remaining database from that user group.

→ The view scheme describe the end user interaction with database system.

DATA MODELS

→ It is the modelling of the data description, data semantics, and consists constraints of data. (Restrictions)

→ It provides the conceptual tools describing the design of a database, at each level of data abstraction.

→ There are 4 models used for understanding the structure of the database.

DATA MODELS

- Relational Data Model
- Entity Relationship Data Model
- Semi-structured Data Model
- Object based Data Model

1. Relational Data Model

→ It design the data in the form of rows and columns within a table.
→ It uses table for representing data and in-between relationship.
→ Tables are also called Relations.
→ This model was initially described by Edgar F. codd in 1969.
→ It is widely used by commercial data processing application.

2. Semi-structured data model

→ It allows the data specifications at places where the individuals data items of the same type may have different attribute sets.
→ XML [Extensible Mark-up language] is used for representing it.
→ JSON, CSV, XML, files are example of semi-structured data.

3. Entity - Relationship data Model

→ It is a based on perception of a real world which consist of a set of basic objects.
→ Introduced by P.P. chen
→ Components :- Entity
Attributes
Relationship → one-to-one Relationship
→ one to-many Relationship
→ Many-to many Relationship
key attributes

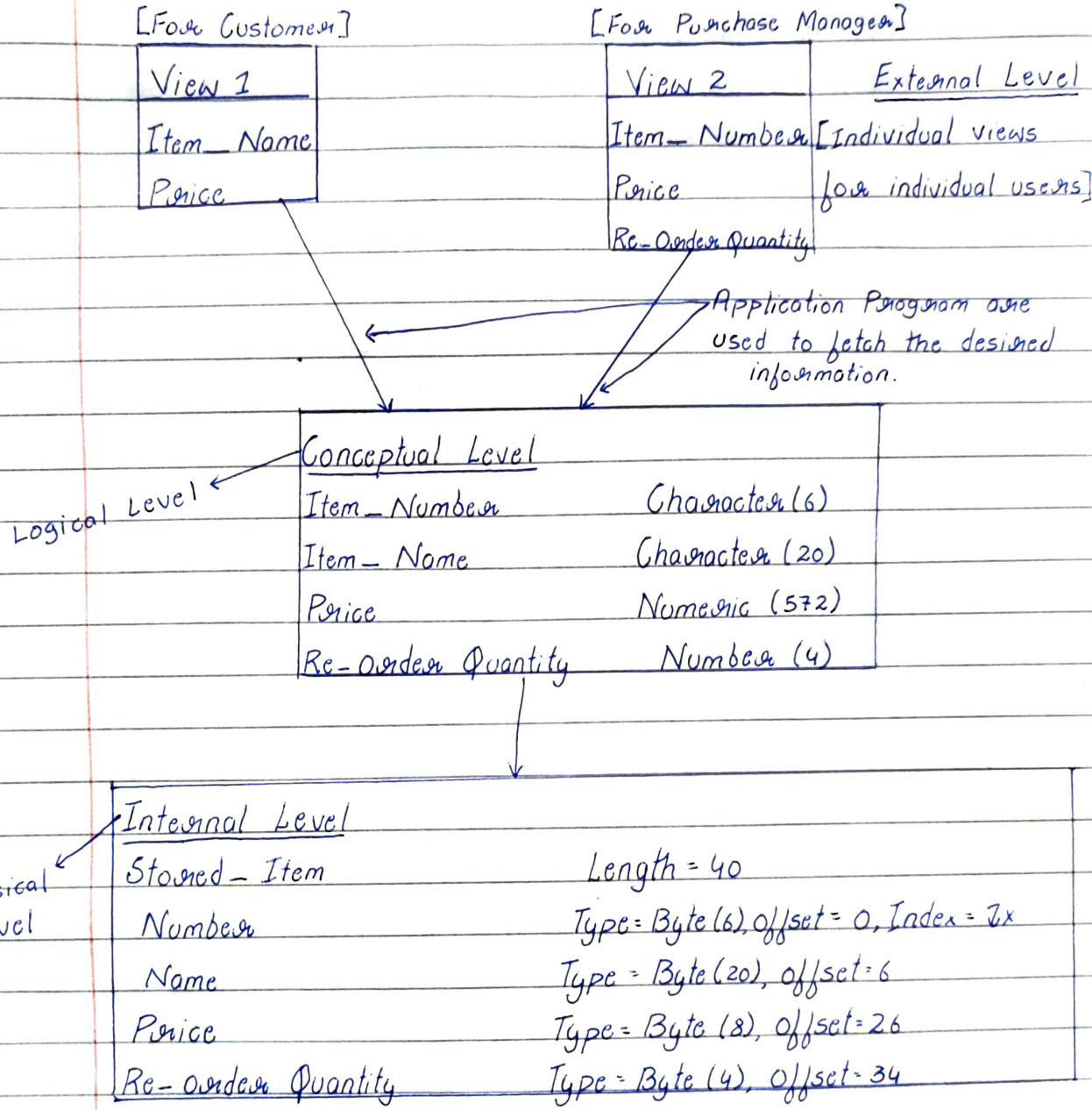
→ It is represented by symbols → Rectangle
Oval
Diamond
Line

4. Object - Based Data Model

→ It is an extension of E-R Model with notions of
① Functions ② Encapsulation ③ Object Identity

→ It support structured & collection types.
→ Objects are data carrying its properties

DATA INDEPENDENCE



Data Independence → The ability to modify a schema definition in one level without affect a schema definition in the next higher level is called DI.

A database is viewed through any level 3 level of abstraction. Any change at any level may affect other

level schema [structure of Database]. As the Database keeping growing, there may be changes made at some level. However, this should never lead to redesiging and re-implementation of Database.

There are two levels of Data Independence

1. Physical Data Independence [PDI]
2. Logical Data Independence. [LDI]

1. PDI → A. It refers to the ability to modify the schema followed at the physical level without affecting the schema followed at conceptual level.
   B. That is, the application programs remain the same even though the schema at physical level get modified.
   C. Modification at the physical level are occasionally necessary in order to improve the performance of system.

2. LDI → A. It refers to the ability to modify the conceptual schema without causing any change in the schema followed at view level.
   B. The LDI ensures that the application programs remain the same.
   C. Modification at conceptual level are necessary whereever logical structure of the database get altered because of some unavoidable reasons.

NOTE → It is more difficult to achieve logical data independence than the physical data independence. The reason being that the application program are heavily dependent on the logical structure of database.

DATABASE ADMINISTRATOR [DBA]

→ Database Administration is implemented by a person or group of persons under the supervision of a knowledge person called as Administrator.

→ This person known as Database Administrator.

→ He is responsible for supervising → Creation, Modification, Maintainence } of Database

→ DBA controls the Database Structure & Set-up the definition for physical as well as logical implementation of the Database.

Functions of DBA / Roles of DBA

(1) Scheme Definition → The original DataBase schema is created by writing a set of definitions which are translated by DDL [Data Definition Language] compiler to a set of tables that are permanently stored in the data dictionary.

(2) Storage Structure & Access Method Definition → Appropriate storage structure and access methods are created by writing a set of definitions which are translated by Data Storage and Definition Language Compilers.

(3) Scheme & Physical-Organization Modification → The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization or to alter the physical organization to improve the performance.

(4) Granting of Authorization for Data Access

A. DBA also maintains that the Database is not accessible to un-authorized users.
B. The DBA is also responsible for granting permission to use the Database and store the profile of each user of a Database.
C. The profile describes the permissible activities of a user on that portion of the Database which is authorized to him.

(5) Routine Maintainence

A. The DBA is also responsible for defining procedure to recover data from failure due to humans, natural or hardware malfunctioning with minimum loss.

B. This recovery procedure would enable an organization to continue working with the available intact portion of Database.

C. He also ensures that enough force disk space is available for normal operations and upgrading this space as requirement.

DATABASE LANGUAGES

Database Languages are used to read, update and store data in a database.

Types:- ① DDL → Data Definition Language
② DML → Data Manipulation Language
③ DCL → Data Control Language
④ TCL → Transaction Control Language

① DDL → (A) It is used to specify the database schema.

B. It is used for creating tables, schema, indexes, constraints in Database.

C. The operation it can perform are –
(a) To create the Database instance – CREATE
(b) To alter the structure of Database – ALTER
(c) To rename Database instance – RENAME
(d) To drop objects from Database such as tables – DROP
(e) To comment – COMMENT

(2) DML → A. It is used for accessing & manipulating data in Database.

B. The operation it can perform are —

(a) To read records from tables — SELECT

(b) Insert records into the tables — INSERT

(c) Update the data in the tables — UPDATE

(d) Delete all the records from the table — DELETE

(3) DCL → A. Used for granting and revoking user access on a Database.

B. The operation it can perform are —

(a) To grant access to user — GRANT

(b) To revoke access from user — REVOKE

(4) TCL → A. The changes in the Database that we made using DML are either performed or rollback using TCL.

B. The operations it can perform are —

(a) To persist the changes made by DML commands in Database — COMMIT

(b) To rollback the changes made to the Database — ROLLBACK.