DBMS UNIT-1 written notes

Database Management Systems (SRM Institute of Science and Technology)

Database Management System

UNIT-1

Introduction :

Database :

It is a collection of data for describing activities of one (or) more organisations.

DataBase Management System (DBMS):

It is a collection of interrelated data and a set of programs to access those data.

(or)

It is a software which is used to maintain (or) align large collection of data.

→ The main goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.

→ The Database System must ensure the safety of the information stored, from system crashes (or) attempts at unauthorized access.

Database System Applications :

Databases are widely used in :

* Enterprise Information

→ Sales : for Customer, Product & Purchase info

→ Accounting : For Payments, receipts, account balances, assets & other accounting info

→ Human Resources : Employees info, Salaries, Payroll taxes & benefits, Generation of Paychecks.

→ Manufacting : For mngmt of Supply chain, tracking production of items in factories, inventories of items & orders for items

→ Online Retailers : for Sales data , Recommendation lists , maintenance of online product- evaluation.

* Banking and Finance

→ Banking : Customer info, accounts , loans , transactions .

→ Credit card transactions : For Purchases on credit cards & generation of monthly stmts.

→ Finance : Storing info about holdings , sales , & purchases of financial instruments i.e stocks , bonds

* Universities

For Students info , Cours Registrations , grades

* Airlines

For reservations and schedule info.

* Telecommunication :

For keeping records of calls made , generating monthly bills , balances on Prepaid Calling Cards , Storing info about Comm. nlw .

I Purpose of Database Systems :

→ In 1960's the data will be stored in the Operating System files .

Fo) ex : University data contain all info about all instructors , Students , Dept's and course offerings

→ In 1960's this kind of info stores in OS files . To allow users to manipulate the info , the System has a no. app prgms which manipulates (Changes) the files , i.e to

* Add new students , instructors & courses
* Register student
* Assign grades , GPA , (or) Results.

→ For ex: To create new Dept. for university, a new permenant files to record info about all instructors, Students, etc. should be created

→ New App.Pgm should be created & this is a time taking process, the system acquires more files. This is nothing but "file-processing System"

There are some Disadvantages in the File-Processing Systems they are:

① Data redundancy and inconsistency:

→ Diff. Pgmers create the files so, the same info may be duplicated in several files (places) Ex: student data

→ Due to this redundancy leads to higher storage and access cost.

→ For ex: Redundant data change at one file but not elsewhere in the system may cause Inconsistency Ex: student address is located at two files in which the changes made at only one file then it cause inconsistency.

② Difficulty in accessing data

Ex: University clerks needs to find out the name of all students who live within a Particular Postal code area

→ clerks asks to data-Processing Dept. to generate the list. If there is no original app. Pgmer then it is difficult to find the list.

→ The solutions to new Programmer is: List out manually and

(or) create new app. Prgm.

③ Data Isolation:

writing new app. Prgm to retrieve the data from diff. files is difficult

④ Integrity problems:

→ The Data values stored in the Databases must satisfy certain types of Consistency constraints.

Ex: University maintains account for each Dept. And the Balance amount of the Dept. should not be less than zero(0). so, this is one constraint, Prgmmers should write app. Prgm for this constraint.

⑤ Atomicity Problems:

Computer (or) any other device is subject to failure then the data be restored to the consistent state and cause total failure.

Ex: Consider to transfer Rs. 500 from account A to account B, if System failure occurs during the execution of Prgm then the amount 500 is removed from account A but not transferred to Account B.

This type of failures are called Atomicity Problems.

⑥ Concurrent - access anomalies:

Ex: Consider two persons joint account having balance of Rs 10,000. Both the Persons debit Rs 500, Rs 100 from the account. Then this cause anomaly.

If two Prgms run Concurrently, they may both read value Rs 10,000, & write back Rs 9500, Rs 9900.

Depending on which one writes the value last the account balance of persons contain either 9500, 9900 rather than 9400 or

⑦ Security Problems:

→ Every user in Database System should not access all the data. But, incase of File-Processing System application Pgms are in the adhoc manner. So, it is difficult to follow this constraint.

Ex: In University info, Payroll personnel need to see only financial info.

II View of Data

A Major purpose of a Database system is to provide users with an abstract view of data. System hides certain data i.e

(i) Data Abstraction
(ii) Instances & Schemas
(iii) Data Models.

(i) Data Abstraction:

→ System must retrieve data efficiently, in which developers hide the complexity from users through several levels of abstractions:

* Physical level: This is lowest level of abstraction describes "how" the data actually stored.

* Logical level: The next-higher level of abstraction describes "what" data are stored in the DB, & what relationship exist among data.

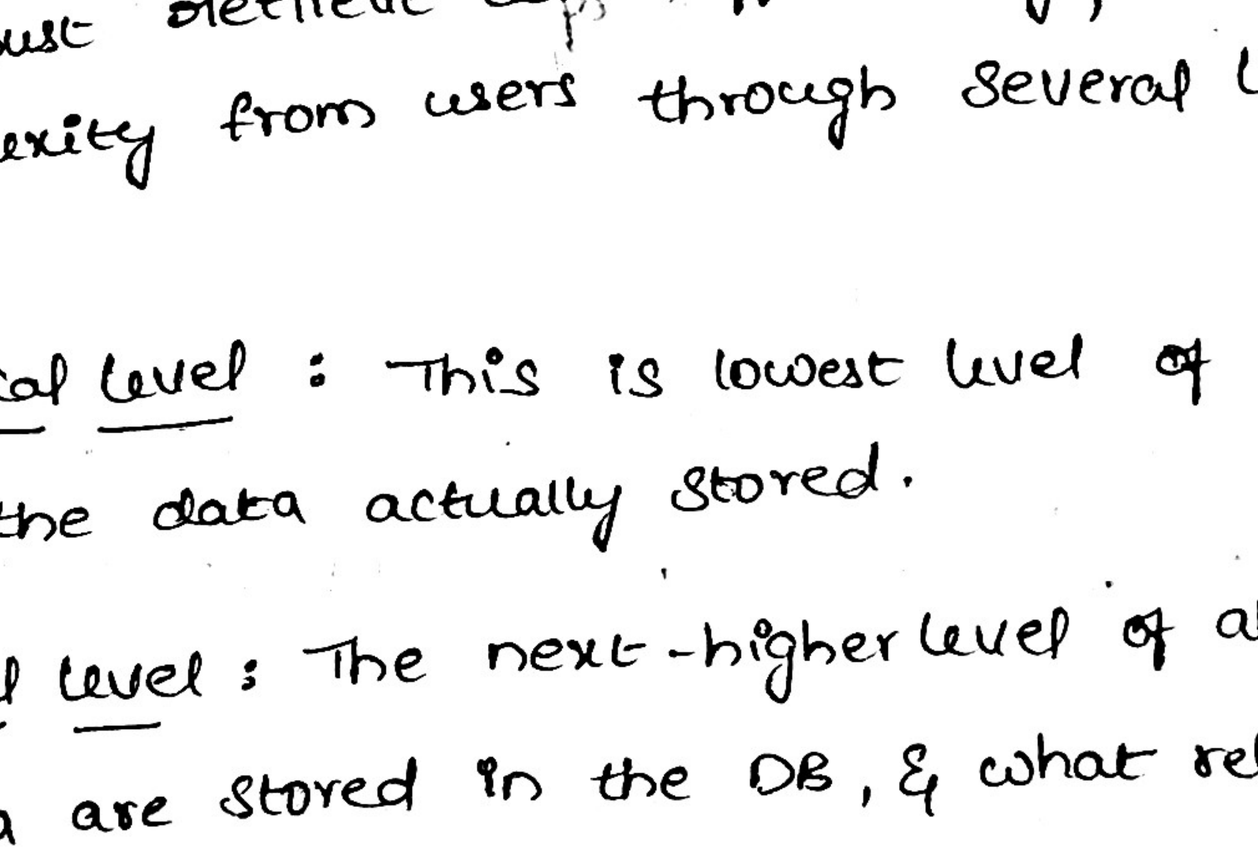

Fig: The 3 levels of data Abstraction

* View level : the highest level of abstraction describes only part of the entire DB.

→ Complexity of the stored DB is not required for the users.

Ex: University Info

→ department, with fields dept_name, dept_code and budget

→ Course, with fields course_id, title, dept_name and credits

→ Student, with fields ID, name, dept_name

→ At Physical level, department, course, Student record will be described. compiler, DB system hides this level from Programmers.

→ DB Administrator will be aware of all the details of Physical level.

→ At logical level, the type definition of each record is described. Prgmers using Programming language at this level.

→ At View level, several views of DB are defined, & DB users see all of the views.

(ii) Instances and Schemas :

→ The DB's change over time when info added (inserted) & deleted. The collection of info stored in the DB at a particular moment is called "Instances"

→ The overall design of the Database is called "DB Schema"

→ In Programming language, schema can be defined as (ex) corresponds to the variable declaration (int a) Ex:

Physical schema : The DB design at Physical level

Logical schema : " " " " Logical level

Sub schemas : " " " " View level

(iii) Data Models :

→ Data model is defined as a collection of conceptual tools for describing data, data relationships, data semantics & consistency constraints.

The no. data models are :

* Relational Model : It uses a collection of tables to represent the data and relationship among them.

→ Tables contain columns and each column has distinct name. So, Tables are also called "Relations".

* Entity - Relationship Model : It is a collection of entities and relationship among them.

• Entity : The thing (or) Object which exist that is distinguishable from other things (or) Object.

* Object - Based Data Model : Objects are created in C++, Java & C#, in which this object-oriented programming has become dominant in s/w development.

* Semistructured Data Model : This is contrast to Data models. The individual data items of same type may have diff. set of attributes.

Ex: XML (Extensible Markup language).

IV Database languages :

A DB System Provides : (1) Data-Definition language
(2) Data - Manipulation language

(1) Data - Manipulation language :

In which that enables users to access (or) manipulate data as organized by the appropriate data model.

Type to access Data : * Retrieval of info * Delition of info
* Insertion of new info * Modification of info

Two types of DML :

* Procedural DML : It requires a user to specify what data is needed & how to get those data.

* Declarative (or) Non-procedural DML : Requires a user to specify what data are needed without specifying how to get those data.

→ NO? Procedural DMLs are easier than Procedural DML's. Because just specifying the what data is needed but not how to get data.

Query: Stmt which is requesting for retrieval, the portion of a DML that involves into retrieval is called "Query language".

2) Data-Definition language

→ DB schema with set of definitions expressed by a special language called "Data-Definition language"

→ Storage structure & access methods used by set of stmts in a special type called "Data storage" & "Definition language"

Integrity constraints that can be tested with minimal overhead :

* Domain Constraints : Domain is nothing but possible values associated with attributes. Declaring an attribute to be particular domain acts as constraint.

* Referential Integrity : The values that appear in one relation for a given attributes also appear in another relation.

Ex: Dept_name is an attribute that appears in both Department, student relation.

* Assertions : An Assertion is a condition that DB must satisfy. Domain, Referential constraints are special forms of assertions.

* Authorization : Differentiation which makes among the users. Those differentiations are expressed in terms of authorization. i.e. : read authorization, insert, update, delete authorization.

V. Database Architecture:

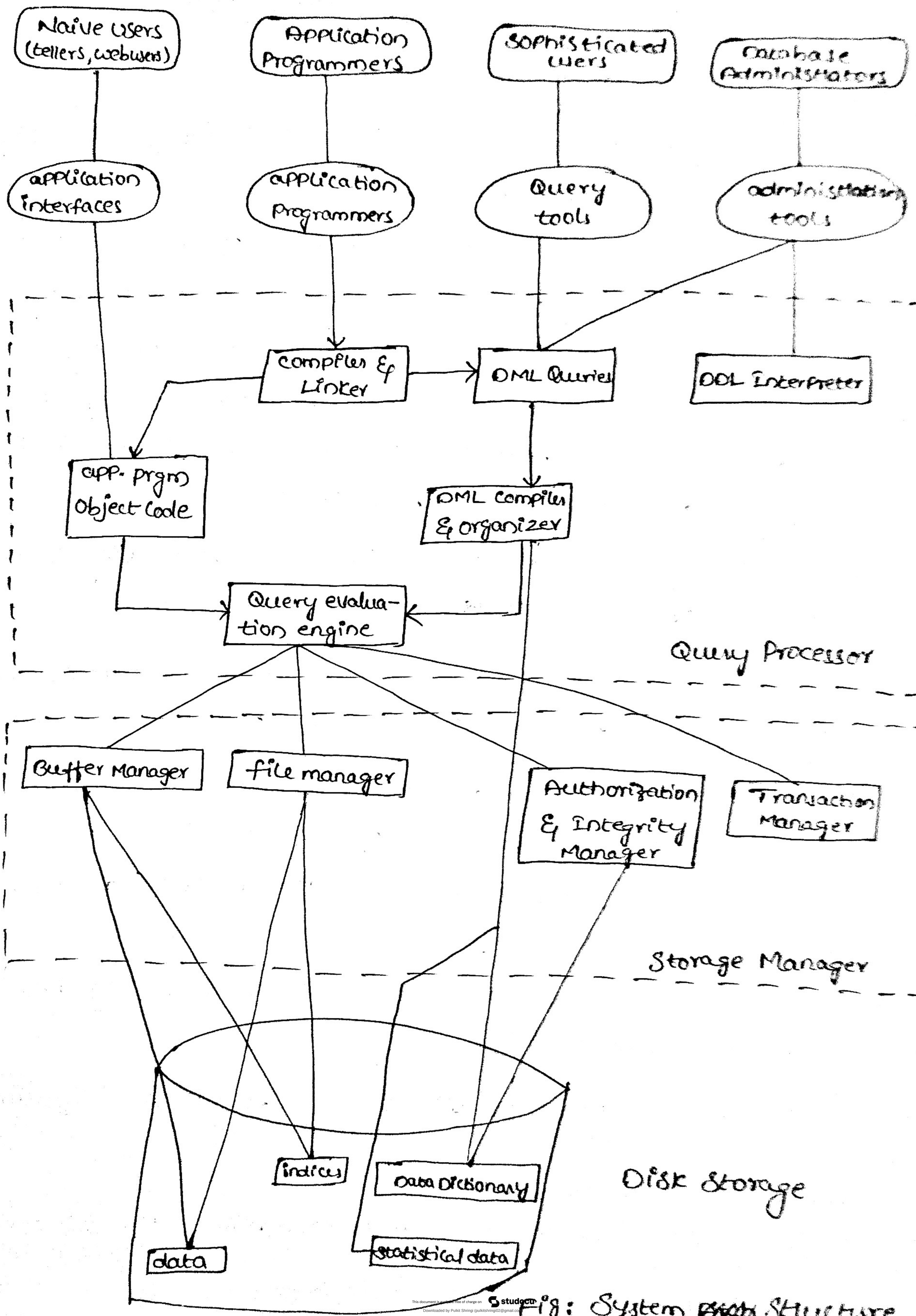

Fig: System Structure

→ The architecture of a Database System is greatly influenced by the underlying computer system on which DB System runs.

→ The system structure is clearly shown before.

(i) Data Storage & Querying:

→ Storage manager is important because database typically requires a large amount of storage space.

→ It is difficult to store huge amount of data in main memory (i.e 100's of GB). So, the information is stored in disks.

→ Data are moved b/n disk storage and main memory.

→ Storage manager is the component of a Database System that provides the interface ~~beem~~ b/n low-level data stored in DB and the app. prgm's & Queries submitted to the System.

→ Storage manager is responsible for storing, retrieving, & updating data in DB. The components of storage manager:

* Authorization & Integrity Manager: Tests for satisfaction of integrity constraints and checks authority of users to access data.

* Transaction Manager: Ensures that the DB remains in a consistent (correct) state despite system failures.

* File Manager: which manages the allocation of space on disk storage.

* Buffer Manager: which is responsible for fetching data from disk storage into main memory.

Storage Manager implements disk storage as a Physical System implementation:

* Data files: Which stores DB

* Data Dictionary: Which stores metadata about the structure of DB in particular schema

* Indices: Which can Provide fast access to data.

(ii) Query Processor:

→ It helps the Database System to simplify and facilitate access to data.

→ Query processor allows Database users to obtain good Performance. The components of Query processor:

* DDL Interpreter: Which interprets DDL stmts and records the information in the Data dictionary.

* DML Compiler: which translates DML stmts in a Query language into evaluation plan consisting of low-level instructions.

* Query Evaluation Engine: Which executes low-level instructions generated by DML compiler.

(iii) Database Users and Administrators:

A primary goal of a DB System is to retrieve info from & store new info into the DB. The Users and Administrators are categorized as:

Database Users and Users Interface: There are 4 different types of users:

* Naive users: They are unsophisticated users who interacts with System by using application programs.

Ex: Google - gmail registration, users wishes to register to gmail by using web interface.

* Application Programmers: They are computer Professionals who write the Prgm's. Application prgrammer's choose many tools from Rapid Application development (RAD) to write App. Prgm.

* Sophisticated users: Interacts with system by using DB Query language (or) by using tools such as data analysis s/w.

* Specialized users: They are sophisticated users who write

specialized DB applications. The applications are knowledge base & expert systems, computer-aided design systems, systems that store with complex data types (audios, graphics data are ex)

Database Administrator:

→ DBMS should have central control of both the data & program that access those data. A person who has central control over system is called "Database Administrator (DBA)"

Functions of DBA:

* Schema Definition: DBA creates original DB schema by executing set of data definitions.
* Storage Structure & access-method definition
* Schema & physical-organization modification: The DBA carries out changes to the schema and physical-organization to improve performance
* Granting of authorization for data access: By granting different types of authorization, the DBA manages which users can access DB.
* Routine maintenance: the activities of routine maintenance are:
  * periodically Backing up (System) Database
  * Ensuring that enough free disk space.
  * " " Performance is not degraded

→ Now adays the DB is connect with the networks. So, DB application can be in the form of: (i) Two-tier Architecture & (ii) Three-tier Architecture.

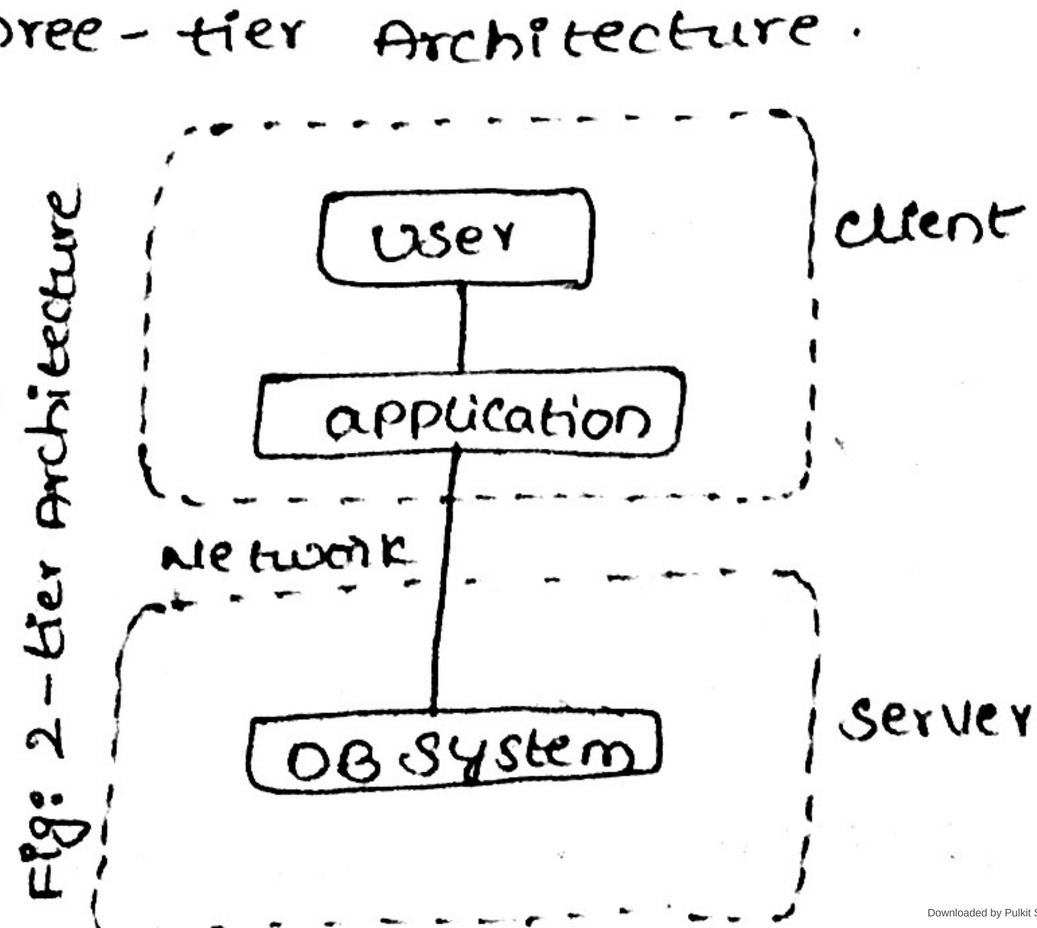

Fig: 2-tier Architecture

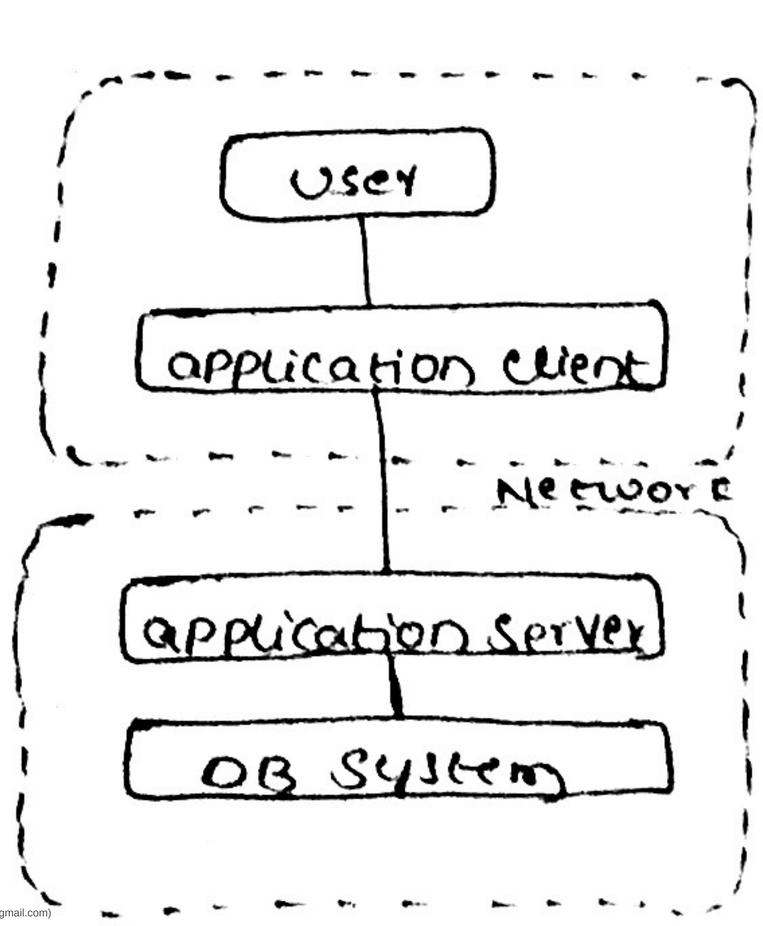

Fig: 3-tier Architecture

→ In two-tier architecture the client invokes DB System at the server through Query language sentiments. The standards like ODBC and JDBC are used bln client & server.

→ In three-tier architecture, the client end communicati with an application server, usually through a form interface. These architecture more appropriatly used in large applications, i.e WWW (world wide web)

VI History of Database Systems:

Information Processing provides growth of computers from early days. The techniques for data storage and processing have evolved over the years:

* 1950s & Early 1960s:

→ Magnetic tapes were developed for data storage. Payroll tasks were automatid, with data stored on tapes.

→ Read the data from one (or) more tapes and writing data to a new tape.

→ Data could i/p from punched cards & o/p from printers.

Ex: salaries raises over the years, then the new tape should be taken to change the new salary. The new tape acts as master tape.

* Late 1960s and 1970s:

→ Widespread use of Hard disks in the late 1960 changed the data processing.

→ The position of data on disk was immaterial, we can access data from any location on disk in just tens of milliseconds.

→ In 1970's Codd defined the relational model in a non-Procedural way of querying data in relational model.

* 1980s

→ The relational model was not used in practice, because the relational database could not match the performance of existing nlw.

→ IBM Research. that developed for the construction of an efficient relational database system.

→ The fully functional SystemR prototype led to IBM's first relational database Product, "SQL/DS".

→ At the same time Ingres System was also developed. Initially the relational DB systems are IBM DB2, Oracle, Ingres, and DEC Rdb.

* Early 1990s:

→ The SQL language was designed for decision support application, which was a Query approach.

In 1990's: The major event that, there is rapid growth of WWW (world wide web). DB provide the support very high trasanction-Processing rates, as well as next reliability & availability.

* 2000s:

→ In early 2000s XML came into existence. which is used for exchanging and storing data and it is associated with XQuery.

→ "autonomic /auto-admin" are also saw significant growth and at the same time "PostgreSQL & MYSQL" also came into existence.

→ The DB Systems are used to built to handle the data management requirements of very large webSites such as Amazon, Facebook, Google, Microsoft and Yahoo.