Unit-1

Page No.:
Date: / /

Q.1) What is Mean by DBMS and what is purpose of the DBMS ?

→

A database Manggement System (DBMS) is a collection of inter-related data and set of programs to access those data.

DBMS Contains information about a particolar enterprise.

- Collection of inter-related data.
- Set of programs to access the data.
- An environment that is both convenient and efficient to use.

Database Application:

- Database is used in the banking transaction.
- Airlines :- reservations, Shedules.
- Universities :- registration, grades.

- Sales :- Customers, products, purchase.
- Online ~~retrainer~~ retrailer :- order tracking, customized recommendations
- Manufacturing :- Production, inventory, orders, Supply chain.
- Human Resources :- employee records, Salaries, tax deductions.

Database can be very large to Store the Data in size of Memory.

Example

This is the example of University Database.

- Application Program Example.
  - Add new students, instructors, and courses.
  - Register students for courses and generate class rosters.
  - Assign grades to students, compute grade point average (GPA) and generate transcripts.
- In early days, database application were built directly on top of file system.

Q.2) Drawbacks of File System to Store the data. and what is Data Redundancy explain it in brief.

→

• Data Redundancy :-

Data Redundancy refers to the duplicate of data, ~~test~~ Sto

• Data Redundancy :-

This can occur in file system because each System could have different formats and Constraints, Meaning the same piece of data could be Stored differently in Seprate Systems. For instance the data could be Stored in "dd/mm/yyyy" format in one System and "mm/dd/yyyy" in another.

• Difficulty in Accessing data :-

File System does not typically have Sophisticated querying capability making it difficult to access data. They usually provide Simple read/write operation on every file basis and do not provide a convenient means to retrive Subparts of a file.

• Data Isolation :-

Data Stored in a file System is not easily integrated with other data. Files can be Stored in various format like text, binary and on disparate System which can make it challenging to combine data from different Source

• Automicity of Updates :-

Automicity refers to the concept that an operation like updating file either completely Success or completely fails with no partial Success.

If a system crash or power failure happens during an update the file can be left in an inconsistent state.

- Integrity Problem :-

Integrity constraints (eg. account balance > 0) become 'buried' in program code rather than being stated explicitly.

Hard to add new constraints or changes existing once.

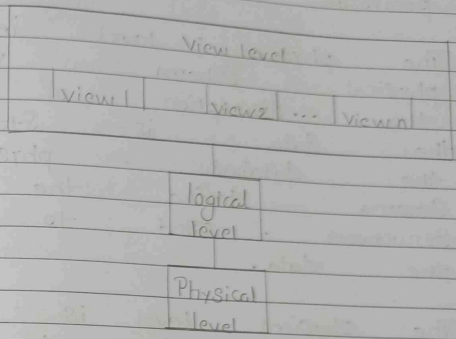
- Concurrent access by multiple users :-

Concurrent access needed for performance

Uncontrolled concurrent access can lead to inconsistencies

Example :- Two people reading a balance (Say 100) and updating it by withdrawing money (Say 50) at each same time.

Q3) Explain levels of Abstractions or view of Data.



Each level has its own purpose and functions and they are connected to each other through mapping.

1) Physical level :-

This is the level of abstraction that defines how the data is physically stored and organized in the database. It includes the details of how the data is stored on disk, the access method used to retrive the data.

and the algorithm used to perform operation on data.

The physical level includes the physical Schema, which is the discription of how the data is stored in the database. physical Schema define the data Structures used to store the data.

The physical level is designed to provide a low-level view of the database that is optimized for performance and efficiency. It allows database Management Systems to store and access data in the most efficient way possible, without worrying about the needs of the users or applications.

example of physical level is abstraction would be sequential file Organization due to the ~~contigeous~~ continuous storage of records.

2) Logical level :-

logical level is the level of abstraction that defines the overall Structure of the database. It defines the relationship between the data elements, the constraints on data, and the operations that can be performed on the data.

The logical level is designe to provide high-level view of database that is independent of any specific implementation. It allows to user define the Structure of the database in a way that is meaningful.

example of the logical level is defining tables and their attributes and specifying relationships between them. A table named 'class' may have different attributes like Student_name, Roll-no, StudentID, and Marks.

3) View level

View level is the level of abstraction that is closest to the end user. It defines how the data is viewed by the users or applications that access the database. It provides high-level view of the database that is tailored to the specific need of each user or application.

For Example

interacting with system using a graphical user Interface (GUI) to access an application features. Here GUI is the View level and the user does not known how and what data is exactly stored hiding the details from the user.

Page No.:
Date: / /

Q.4) What is instances and Schema Explain it in brief.

→ Shema:-

DBMS Schema means designing the database. Database Schema it is the defination of the database or it is defined as the description of the database.

Schema rarely change. the correspond to the variable declaration of a programming language.

Schema defines the basic structure of database i.e. how the database will be stored in the database Schema is same for whole database. It does not change very frequently.

Schema is further divided into three types. There are as follows.

1. Logical Schema.
2. View Schema
3. Physical Schema.

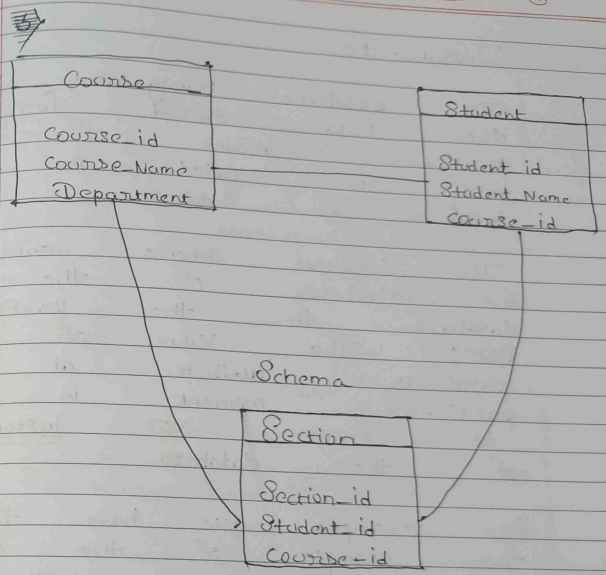
The Schema defines the logical view of the database. It provides some knowledge about the database and what data needs to go where.

1) Physical Schema :-

In the physical Schema, the database is designed at the physical level. At this level, the Schema describe how the data block is stored and how the storage is managed.

2) Logical Schema :-

In the logical Schema, the database is designed at logical level. At this level, the programmer and data administrator perform their work. Also at this level, a certain amount of data is stored in structured way But the internal implementation data are hidden in the physical layer for the security purpose



This diagram shows the relationship between the Section and the course. Schema is only type of structured view of the database.

Instance :-

In Database Management System the Data Stored for a particular amount of time and is called an instance of database.

The database Schema defines the attributes of the database in the particular DBMS. The value of the particular attribute at a particular moment in time is known as instance of the database

For example we have taken the example of the attribute of the Schema. In this Diagram each table :- Contains two row or two records. In the above Schema of the table, the table employee has some instances because all the data stored by the table have some instances.

Q. 5) Explain Physical Data Independence in detail.

→

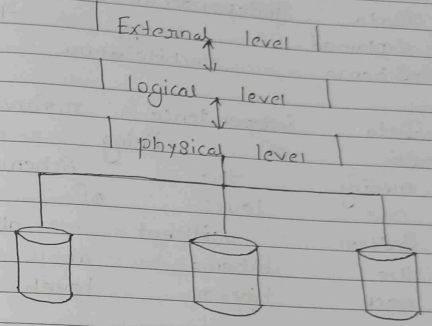


Fig. Data Independence.

physical Data Independence:-

physical Data Independence can be defined as the ability to change the physical level without affecting the logical level.

physical Data independence gives us the freedom to modify the - Storage devices, file structure, location of the database etc. without changing the definition of view level.

Data Independence

- Data Independence can be explained using the three-Schema architecture.
- Data independence refers characteristic of being able to modify the Schema at one level of database System without altering the Schema at the next higher level.

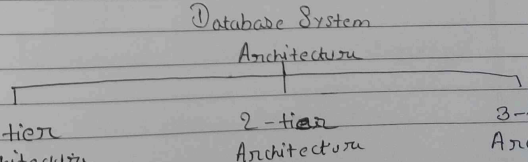
Q.6) Draw and explain the Database System Architecture with all its Component.

→ The Architecture of a database systems is greatly influenced by the underlying computer System on which the database is running.

- Centralized.
- Client-Server
- Parallel (Multi-processor)
- Distributed.

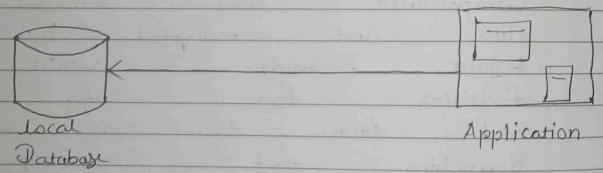
In DBMS architecture, in RDBMS, the database an its components are stored in disk and mangged by the RDMS Software, which provide facilities for adding, updating, and delete data, as well as Searching and retreiving data based on Specific condition.

Types of Database System Architecture



1) 1-Tier Architecture

The simplest form of DBMS architecture is the 1-tier design, where all DBMS components reside on a single server or platform allowing direct access to the database by end user. This direct connection result in rapid response time, making it popular choice among programmers seeking to enhance local applications.



2) 2-tier Architecture.

2-tier Architecture in DBMS refers to a client-server architecture where the user interface and the application logic are seperated into two separate component. The client component is typically the user interface and the server component is responsible for handling the data and buisness logic. In this architecture, the client component communicates directly with the server component to request data and perform actions.

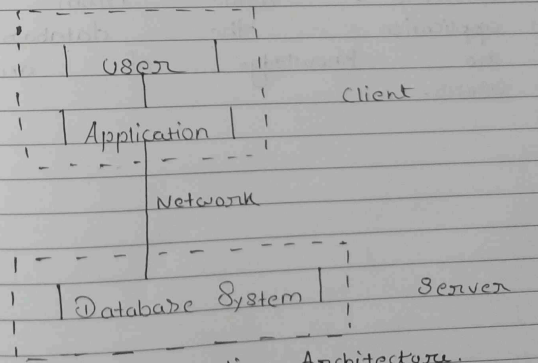


fig. two-tire Architecture.

3) 3-tier Architecture

Another level layer exists between the client and Server in the 3-tier Architecture. The client cannot Communicate directly with the Server with this design.

On the client side, the program communicates with an application Server, which then communicates with database System.

Beyond the application Server the end-user has no knowledge of the database's existence. Aside from this application, the database has no knowledge of any other users.

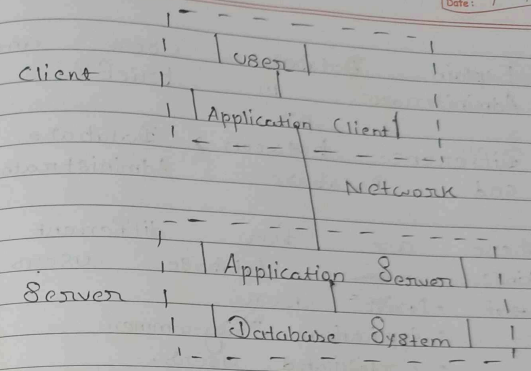


fig. 3-tier Architecture.

Q. 7) Explain Database User and Administrator in brief?
or
Difference between Database User and Database Administrator

→

There are four different kinds of database users:
- Naive User
- Application Programmer
- Sophisticated Users.
- Specialized Users.

1) Naive User:-

Naive users are people who don't knowin much about computers and use the system by calling up one of the application programs that have been written.

For example, A bank teller uses a program to move the 15000 from Account A to Account B. This program asked the teller for the amount of money that needs to be moved,

the account that the money is comming from and the account that the money is going to.

2) Application Programmers:-

These are people who work in computers and write programs for applications.

Application programs have lot of tools to choose from when making user interfaces.

3) Sophisticated Users:-

These people knows how to use the system without writing programs. They write their request in a language for talking to database.

They send each query to a query processor whose Job is to turn DML Statement into instructions that the Storage Manager can understand

4) Specialized Users:-

Specialized users are advanced users who write database program that doe, don't fit into the traditional way of processing data.

Among these applications are computer Aided design System knowledge base and expert System, Systems that store data with complex data types.

* DataBase Administrators:-

One of the Main reasons to use DBMS is to have centralized control over the data and the programs that access the data. The person incharge of the System as a whole is termed the database Administrator (DBA).

Following are some duties of DBA:-

1) Defining Schema:-

The DBA makes the original database Schema by writing a set of definitions. The DDL compiler translates these definitions into a set of tables which are then stored into the data directory.

2) Defining Access Method:-

The DBA makes the ~~write~~ right Storage Structures and access Method by writting a set of definitions that the data-Storage and data-defination language compiler translate.

3) Modification of Schema and physical Organization:-

The DBA make changes to the Schema and physical organization to reflect how the organization needs change or to change the physical organization to make it run better.

4) Giving Permission to Access Data:-

The database Administrator (DBA) can control which part of database different users can access by giving different types of permission.

5) Integrity - Constraints Specification:-

The values of data stored in the database must meet certain consistency requirement. The DBA must tell the database about this Constraints.

The integrity constraints are kept in a special structure that the database system checks every time it makes a change.

Q.8) Explain Database engine in brief.

→ A database engine is the core software component that powers database system. It simplifies creating, reading, updating and deleting (CURD) data stored in a database, ensuring data integrity, security and efficient access.

Database engine manage data storage, retrieval and manipulation. They interact with data files, indexes and system memory to execute SQL commands, which make sure that query performance and transaction management are optimized.

Database Engine Example.

- Oracle :- A leading commercial relational database engine.
- MongoDB :- A popular NoSQL Database engine catering to document-oriented storage.

Database Engine Advantages

- Ensure that data remains consistent and secure through automic transaction and encryption.
- Uses indexing, caching, and query optimization techniques to retrive data faster.
- Multiple user can interact with the database simultaneously without the data conflicts.

Database Engine Disadvantages.

- High performance engine are difficult to set up, tune, and Maintain.
- Large Database and high transaction volumes require significant hardware resource
- Different Databases engine can be challenging, resulting in dependency on a single vendor.

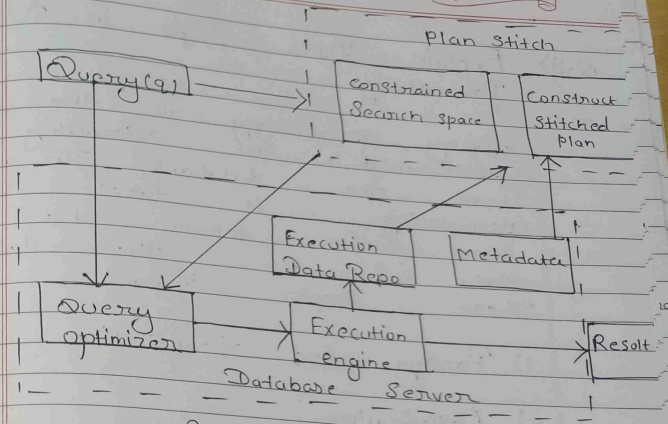


fig. Database Engine.

Q.9) Explain Basic Steps in Query Processing.

→ Query processing is the activity performed in extracting data from the database. In query processing, it takes various steps for fetching the data from the database. These steps involved.

1) Parsing and translation.

2) Optimization.

3) Evaluation.

1) Parsing and Translating.

In this step the user writes its request in Structured query language (SQL) an the DBMS convert it into machine understandable low level language. The query is first pick by the query processor which scans the queries and parses them into individual token. After that it examines the correctness of the query and it converts the token into trees graphs and relational expression.

These are the following checks performed by the parsing

- Syntax check
- Semantic check
- Shared Pool check.

Q→ SELECT * FROM EMPLOYEE;

2) Query Optimization.

This step analyze SQL queries and determines effective execution mechanism in query processing. optimizer uses Statistical Data.

i.e. information about the length of records, size of the table, the index created on table

The query optimizer produces one or more query plans and the most efficient is selected and used to execute query

3) Evaluation :-

They got many execution plans through query optimization Although they give the same output but differ in terms of space and time consumptions. Evaluation helps us to choose effective and less cost consuming execution plan to give the final result. by accessing the data from the database.

In SQL some of the factor considered to calculate the cost of evaluation plan by the Optimizer are.

- CPU time.
- Number of Operation.
- Numbers of tupples to be Scanned
- Disk access time.

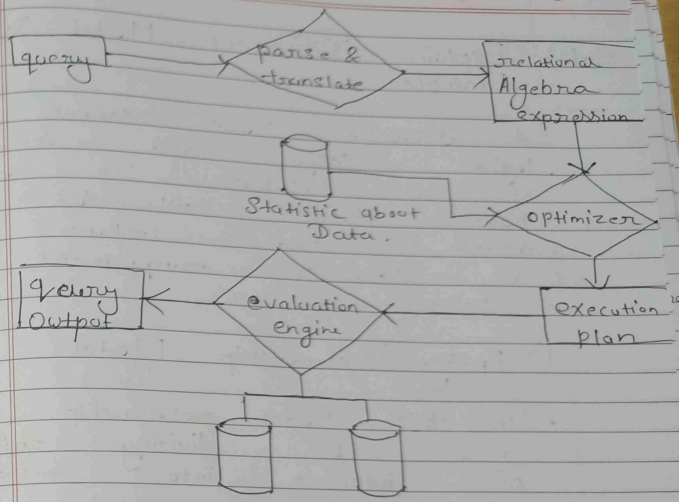


fig. 7 Query Processing.

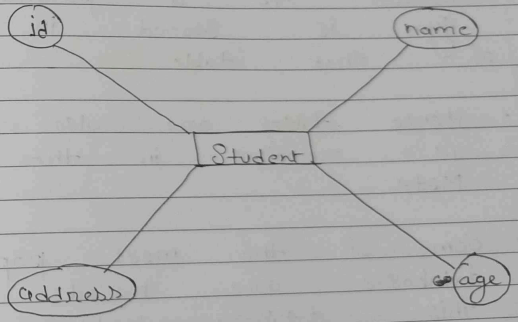
Q.10) Explain Different Model with Example.

→

A Database Model defines the logical design and structure of a database. It defines how data will be stored, access, and update in a database management System.

1) Entity - Relationship Model

- In this Model relationship are created by dividing object of interest into entities and their characteristics into attributes.
- Different entities are related using relationships.
- ER-Models are defines to represent the relationships in pictorial from & to make it easier for different Stake-holder to understand.
- This model is good to design a database, which can then be turned into tables in a relational model.
- Example if we have to design a School database, then the Student will be an entity with attributes name, age, id, address, etc.



2) Relational Model

- In this Model, data is organized in 2-dimensional table and the relationship is maintained by storing a common field.

- The basic structure of the data in relational Model is tables. All the information related to a particular type is stored in rows of that table.

- Hence, tables are also known as relations in the relational Model.

- Some of the most popular databases are based on this database Model. For example, Oracle, MySQL

Student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	19

Student_id	name	teacher
1	Java	Mr. J
2	Phython	Miss Py
3	Javascript	Ms. Ja
4	C#	Mr. C

Student_id	Subject_id	marks
1	1	98
1	2	97
2	1	76
3	2	88

3) No-SQL Model

- The No-SQL Model support an unstructured style of storing data.

- Data is stored as documents

- It provides flexible schema.

- It does provide features like indexing, relationship between data.

- The support for data querying is limited in the NoSQL Model.

: This database model is well-suited for Big data applications, real-time analytics etc. content management System.

Unit-2

Q.1) Explain Data Defination Language & Data Manipulation language in detail.

→

Data Defination Language :-

DDL or Data Definition language actualy consist of the SQL command that can be used to define the database schema. It is used to create and modify the structure of database object in the database.

DDL is the set of SQL command are ~~normally~~ used to create, modify and delete database structure but not data.

These command not used by a general user, who should be used / accessing via an application.