ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University

(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM ACCESSIBILITY

B.Com. (Digital)

SEMESTER-VI

(BCDB33601T): DATA BASE MANAGEMENT SYSTEM

Course Coordinator: Dr. Monika Pathak

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

COURSE COORDINATOR & CONTENT WRITER:

Dr. Monika Pathak

Assistant Professor, School of Sciences and Emerging Technologies

Jagat Guru Nanak Dev Punjab State Open University, Patiala

COURSE OUTCOMES (COs)

After completion of the course, learners will be able to

1. Understand the fundamental elements of database management system
2. Understands the three level architecture of DBMS and mapping between these levels.
3. Familiar with the hierarchical model, network model, entity relationship model and relational model
4. Have knowledge of normalization technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divide larger tables into smaller tables and links them using relationships.
5. Familiar with MS ACCESS

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in December 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open University of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,

Dean Academic Affairs

Name of Programme: B.Com (Digital)

Name of Course: Data Base Management System

Course Code: BCDB33610T

Semester: 6th Semester

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 6
Pass Percentage: 40%

SECTION A

Introduction: Basic concepts of Database, Characteristics of a Database, Database System Environment. Roles in Database Environment: Database Administrators, Database Designers, End Users, Application Developers. Database Management System: Definition and Features, Traditional File System vs DBMS, Significance and Classification of DBMS, Architecture: Data Models, Types of Data Models Conceptual Data Models, Physical data Models, Representational Data Models, e.g. Object Based Models, Record Based Models, Database Schema and Instance, Three Schema Architecture, Data Independence – Physical and Logical data Independence. Entity-Relationship model: Concepts, Entities and Entity Sets, Attributes, Mapping Constraints, E-R Diagram, Weak Entity Sets, Strong Entity Sets, Aggregation, Generalization, Converting ER Diagrams to Tables.

SECTION B

Relational Data Model: Concepts and Terminology, Characteristics of Relational Database. Constraints: Integrity Constraints- Entity and Referential Integrity constraints, Keys- Super Keys, Candidate Keys, Primary Keys, Secondary Keys and Foreign Keys. Relational Algebra: Basic Operations, Additional Operations. Normalization: Functional Dependency, Full Functional Dependency, Partial Dependency, Transitive Dependency, Normal Forms– 1NF, 2NF, 3NF, Boyce-Codd NF, MS-ACCESS: introduction to MS-ACCESS, working with databases and tables, queries in Access, Applying integrity constraints, Introduction to forms, sorting and filtering, controls, Reports and Macro: creating reports, using Macros.

Reference Books:

1. H. F. Korth & Silverschatz, A., Database System Concepts, Tata McGraw Hill.
2. Hoffer, Prescott, Mcfadden, Modern Database Management, Paperback International.
3. Elmasri & Navathe, Fundamentals of Database Systems, Addison-Wesley.
4. Connolly & Begg, Database Systems, Pearson Education.
5. Ivan Bayross, SQL,PL/SQL The programming language of Oracle, BPBPublications.

Section A

UNIT 1: INTRODUCTION OF DATABASE MANAGEMENT SYSTEM

1. INTRODUCTION

1.1 DATABASE CONCEPTS

1.2 TRADITIONAL FILE MANAGEMENT SYSTEM

1.3 DATABASE

1.4 DATABASE MANAGEMENT SYSTEM (DBMS)

- **1.4.1 CHARACTERISTICS OF DATABASE MANAGEMENT SYSTEM**
- **1.4.2 OPERATIONS/FUNCTIONS OF DATABASE MANAGEMENT SYSTEM**
- **1.4.3 ADVANTAGES OF DATABASE MANAGEMENT SYSTEM**
- **1.4.4 DISADVANTAGES OF DATABASE MANAGEMENT SYSTEM**

1.5 COMPONENTS OF DATABASE SYSTEM

1.6 DBA (DATABASE ADMINISTRATOR)

1.7 COMPARISON OF FILE MANAGEMENT SYSTEM WITH DATABASEMANAGEMENT SYSTEM

1.8 CATEGORIES OF DBMS

- **1.8.1 CENTRALIZED DBMS**
- **1.8.2 PARALLEL DBMS**
- **1.8.3 DISTRIBUTED DBMS**
- **1.8.4 CLIENT/SERVER DATABASE SYSTEM**

1. INTRODUCTION

The exponential growth of information technology and its dependency in different sectors of society results in collection of hugs data. The large data collection has to be stored in such a way that is should be retrieved and processed as per the requirement of the user. Traditionally, data was manually maintained, stored in fields, updated and retried manually. The system was worked with very small amount of data which was isolate handled by single user. With the increase in size of data and access of multiple

users for single source of data, manually management of such data was nearly impossible in such a scenario, the concept of database management system was originated. The goal of the database management system is to store information in such a way so that it can be access with ease. The database management system is aimed to perform basic operation like: storing, retrieval, sorting, searching, and deletion of records in database. It plays a critical role in almost all area where computer systems are used for information processing like business, engineering, medical, defence, education, library etc. The database sheared among different users. Sometime, it is called mediator between user and data as it communicate between user and data. It responses to the user with results after processing query raised by the user. Before going into more technical detail of DBMS, let us go through the basic concepts:

1.1 DATABASE CONCEPTS

The conceptual understating of database is required to go through two elementary database concepts: Data and Information

I. ***Data: It is defined as representation of facts, concepts, and instruction in a from which is suitable for communication, interpretation by human or computer.***

- Data can be recorded and have meaning.
- Examples of data are: height, weights, prices, costs, names of things, marks, image, and sound.
- In a formal manner, data is suitable for understanding and processing's.

Data can be represented with different character set and format which are stipulated in table given below:

Representation of Data		
Sr. No.	**Format**	**Character set**
1	Alphabet	(A-Z a-z)
2	Digits	(0-9)
3	Special characters	(+,-,*,/,@,#)

4	picture	Picture in jpeg, Gif, Tiff, etc format
5	Sound	Sound in mp3, mp4 format

Table 1.1 Representations of Data (Format Character Set)

In a any information processing system like database management system, data us considered as raw material or figure as it itself is not significant. It requires to be processed to come up with suitable fact and figure which is called as information.

II. Information: It is defined as processed from of data which has significance in decision making or performing some action. In another words, information is data that has been converted into some useful form.

The information is a result of processing of data according to specific requirement. In hugs data collection user is asked to make query to fetch required information. The following logical diagram demonstrated the same concept:

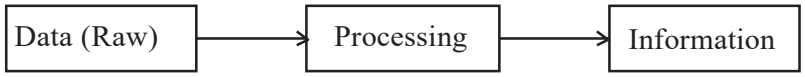


Fig. 1.1: Data Processing for Information

It is very important to throw light on the basic difference between these two elements: Data and information, as these terms are quite miss used among beginners. They sometime give us information on the place of data but both information and data are different from each other. The tabular representations of fact as shown below very well explained the differences.

Sr. No	**Data**	**Information**
1	Data is raw facts and unorganizedfigure that need to be processed.	Information is processed from of data which may be further processed to gain knowledge.
2	Data is useless until it is organized and does not convey any message.	Information is useful and conveys meaningful message.

3	Data is used as input for any data processing application.	Information is the result of any data processing application.
4	Decision making is not recommended on data as data may or may not be meaningful .That is why: data not help in decision making.	Information is always meaningful and plays vital role in any decision making process.
5	Data is available in unorganized and un specific format.	Information is always required in organized and in specific format.
6	Data is a collection of atomic levels of pieces. It collectively represents different fact and figure.	Information is organized collection of data and is always represent about specific entity.
7	Data itself has no significance in business as data is not in the form that can be interrelated.	Information is interrelated to data and has strong significance in business.
8	Data representation order is not significant as it may be in any order. It has no effect of meaning	Information must be in specific order otherwise It may have different meaning.
9	The data cannot be interpreted as it is very difficult to understand. It may have different meaning for different person in different situation.	Information is concrete in nature and easy to Understand. It has same meaning for everyone in any situation.
10	Example: Data may have figure like 20, 30, 50, 70 that it is a raw figure which has no significance.	Example: Data is proceed and associate with some meaningful facts like 20 year old,30 kg weight,50 gm Gold, 70 km/hr, etc May be processed from of data. Now

		these figures have some meaning.

Table 1.2: Different between Data and Information

Importance of information in organization: the organisation has maintained data of activities conducted during the session which include sale purchase data, human resource data, store data, etc as shown below in logical diagram.

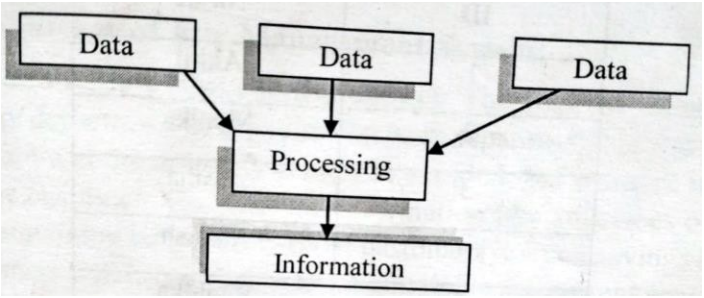


Fig. 1.2: Organizational Data Processing Conceptual Model

Every organization has data processing systems that applied on different data to fetch information for smooth working of their organization. The information may be significant to the organization for many purpose some of these are listed below which emphasis that information is very important for smooth working of any organization.

Based on the above points, we conclude that the information help in planning, the action in the process of running and protecting the system.

- To gain information about the organizational sale and purchase.
- To access information of employee in the organization.
- To know about the future predictions of the organization.
- To know about the surroundings and whatever is happening in the society and universe.
- To keep the system up to data.

1.2 TRADITIONAL FILE MANAGEMENT SYSTEM

The file management system is a traditional approach to store and mange data in files. It is early day approach when records are stored in different files with different format. Each department in organization have own file storage system where specific

applications are designed to process these applications. The departments have their own set rule to store and retrieve data from file system. The system to handle these file was called file management system. Such systems are file department and incompatible to other file system. It means file system of one department may not work for file processing of another department. But such system is preferably good as compare to manual file management.

Key points:

- File processing system is a simple computer file system.
- It is a group of files storing data of an organization.
- Files are in the form of text. Even records are also in text form.
- Each file is independent from one another.
- Each file is called a *falt file.*
- It uses hard disk or CD to store the data.
- Files are designed by using programs written in programs written in programminglanguages such as C, C++.
- Searching is very difficult. Searching will start and continue till find the result.
- If data is very large then searching will take long time.

ID	Name
1	Akhil
2	Monika
3	Aastha
4	Ankush
5	Radhika

Formula for searching = n+1/2

- Files are suitable when number of store items is small.
- It is not suitable when we have to perform data processing.
- As a system became more complex, file processing system presented many limitations and were difficult to maintain.

Example: To understand file management system in details we are taking practical example. Let us consider a business organization where different department are organized and performed different tasks. Suppose department are Sale and Purchase Department, Inventory, Control Department, HR Department and Production unit. Every department have own system to store information in files. The following diagram shows how file management system works and manages data is different files.

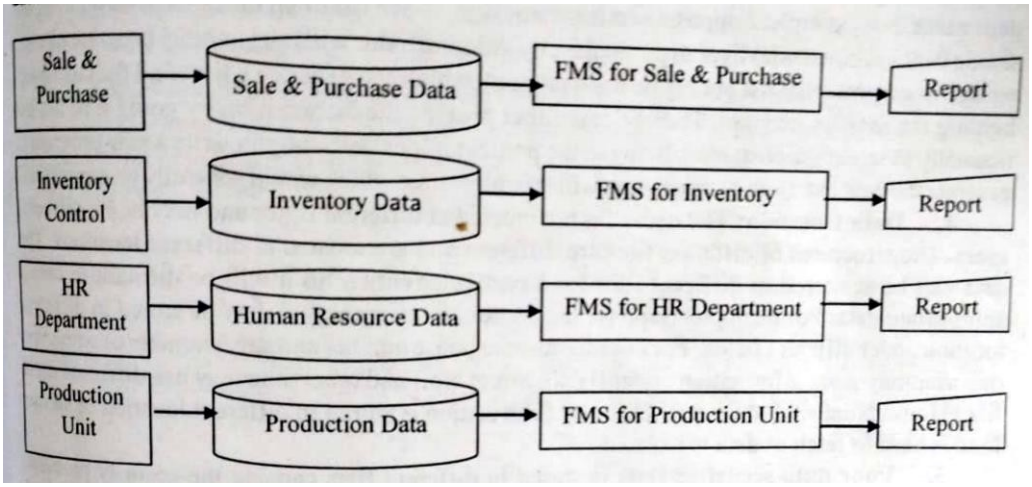


Fig. 1.3: File Management system in a business Organisation

It is very much clear from the above diagram that in file management system every department has separate storage of data and specific application for processing. In such as system inter-departmental access is not possible and there is duplication of organizational data. So the file management system has many limitations which are addressed in new concept, called, database management system.

Limitation/ Disadvantages of file Management System:

1. **Duplication of Data (Data Redundancy):** The files are created according to the application and every department in organization has separate file system. So in that case the repetition of information about an entity cannot be avoided. For instance in Bank, the files are maintained about the customer. The personal information like addresses of customer holdings savings account and also the address of the customers will be present in file maintaining the current account. Even in case if same customers

have a saving account and current account his address will be present at two places. There is duplication of data as files are not shear able among different applications.

2. **Data Inconsistency and Inflexibility:** Data isolation limited the flexibility of file processing system in providing users with ad-hoc information requests. Data inconsistency means data about same entity stored in different files are not up-to-date and is not identical at same time. It is due to duplication of data which leads to greater problem than just wasting the storage. Same data which has been repeated at several places may not match after it has been updated at some places. For example: Suppose the customer requests to change the address for his account in the Bank and the Program is executed to update the saving bank account file only but his current bank account file is not updated. Now the addresses of the same customer have two addresses stored in two different locations that are called data inconsistency.

3. **Difficulty in Accessing Data:** In file management system, the program is designed for generating ad hoc reports. It means that program is for not general purpose and is data dependent. For example: Suppose administrator want to see list of all the customers holding the saving banks account who lives in particular city. Administrator will not have any program already written to generate that list but say he has a program which can generate a list of all the customers holding the savings account. Then he can either provide the information by going thru the list manually to select the customers living in the particular locality or he can write a new program to generate the new list. Both of these ways will take large time which would generally be impractical.

4. **Data Isolation:** The data files are created at different times and may be by different users. The structures of different files are different and are located at different locations. The data will be scattered in different files for a particular entity. So it will be difficult to obtain appropriate data. For example: Suppose the address of an employee may be stored in different location under different fields. For instance to store house number and street number of employee, one user may store information under (HNo, Street No.) and other one may use different name like (House Number and Street). This

way information is stored in different location of similar kind is hard to fetch as data is isolated.

5. **Poor data security:** Data is stored in different files causing the security problem. The data should be protected from unauthorized users. Every user should not be allowed to access every data.

6. **Difficult to Show Data According to User:** In file processing system, it was difficult to determine relationships between isolated data in order to meet user requirements.

7. **Concurrency Problems:** When more than one user are allowed to process the database. If in that environment two or more users try to update a shared data element at about the same time then it may result into inconsistent data. In case of file management system such concurrent access to data is hard to implement.

8. **Data in separated files:** Data is in more than one file and it is difficult to take data from more than one files.

9. **Data Dependence:** Data dependence means it is impossible to change storage structure without affecting the application program. If the format of a certain record was changes, the code in each file containing that format must be updated.

10. **Incompatible File Formats:** Each programmer stores the data in the file in the format as per the choice as there is no standard file format for storing the file. It becomes very difficult to handle the different files in different format.

The database management systems are designed to overcome above listed problems along with other advanced database concepts. The following section is designed to address issues like Database, DBMS, Difference between Database and DBMS, Characteristics of DBMS, Functions of DBMS, Advantages and Disadvantages of DBMS.

1.3 DATABASE

Database: It is a computer based record keeping system whose over all purpose is to record and maintains data. It is designed to hold bundle of organizational data. It holds the records, fields, and cells of data.

The database stores the known facts that can be recorded and that have implicit meaning. Data is represented in database in different levels of abstraction in its architecture.

Typically, there are three levels: External, Conceptual, and internal. The following diagram show how data is represented in different levels:

- **External Level:** It defines how user views data. Single user may have multiple views.
- **Conceptual Level:** It is a communication medium between external and internal level. Its representation is unique regardless of external level and internal level.
- **Internal Level:** It defines how data is physically stored.

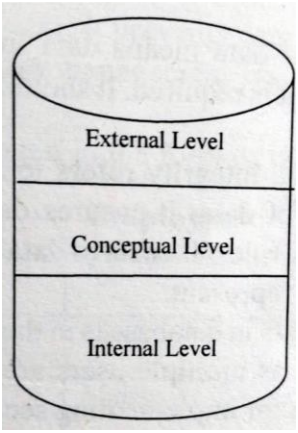


Fig. 1.4: Database Abstraction Levels

Key Points:

- The database is a shared collection of logically related data, designed to meet the information needs of an organization.
- It is a computer based record keeping system. Its overall purpose is to record and maintain the information.
- The database is a single large repository of data which can be used simultaneously by many departments and users.
- It holds not only the organization's operational data but also a description of thedata. It is also defined as a self-describing collection of integrated records.

- The description of the data is known as the Data Dictionary or Meta Data (the 'data about data").
- We can perform many operations on database such as:
    a) To add new operation.
    b) To modify/ edit the existing information.
    c) To remove/delete the unwanted information.
    d) To retrieve/ view the stored information.
    e) Arrange the information in a desired manner.
- Database is manages by an individual or group called Database Administrator (DBA), Who is responsible for designing, creating and maintaining the database to satisfy the needs of the users.
- All access to database is automated by special software called Database Management system (DBMS).
- The tern database is generally confused with DBMS. The database is a concept to representation data whereas DBMS an application program to provide access on database both together represent Database System.

Characteristics of Database:

Database has some Characteristics in order to meet the standards which are as follows:

1. **Data sharing:** Database should be capable to be shared among different users and applications.
2. **Persistence:** Persistence of data means data in a database exist permanently and available in time whenever it is required. It should live beyond the scope of the process that created it.
3. **Integrity/Correctness**: Data integrity refers to the property of data which enforce constrains to safe format of data. It ensures data should in a uniform format and implemented with integrity rules. It ensures data should be correct with respect to the real world entity that they represent.

4. **Security:** The security of data in database is in the top of priority. It should be protected from unauthorized access as multiple users are sharing database. Database should have their own mechanism for implementing security.
5. **Consistency:** The consistence of data is must whenever more than one data element in a database represents related real world values. The values should be consistent with respect to the relationship.
6. **Non-Redundancy**: The data in database should not be duplicated as no two data items in a database should represent the same real world entity. The non-redundancy helps to reduce size of the database and avoid inconsistency of data.
7. **Independence:** The database has three different levels (External, Conceptual,and Internal) to represent data. These levels should be independent of each other so that the changes in one level should not affect the other levels.

1.4 DATABASE MANAGEMENT SYSTEM (DBMS)

DBMS: A database management system is a collection of interrelated data and a set of programs to access those data. The interrelated data is called database which is a shared collection of logically related data, designed to meet the information needs of an organization.

The primary goal of a DBMS is to provide methods to store and retrieve database information that is both convenient and efficient. Database sustems are designed to manage large bodies of information. In addition, the database system must ensure the safety of the information stored. Despite system crashes or attempts at unauthorized access. If data is to be shared among several users, the system must avoid possible anomalous results.

Key Points:

- DBMS is a software system that allows user to create , maintain and delete a database. It provides controlled access to the data.
- It centralized the database.
- It is a computerized system which maintains the data.

- DBMS is an intermediate between programs and data. It is used to make information from data.
- DBMS is a collection of programs which are required to perform different task on database. It perform various operation on data like defining structure of data, accepting data, format data as per user requirement, hide data, allow concurrent access, backup and provide security to data.
- DBMS ensures the privacy of data. It prevent data from unauthorized users.
- Commercially available database management systems in the market are dbase, Foxpro, Oracle etc.
- In DBMS, data can be represented in the dorm of tables.

Employee

Ename	Empno	Job	Sal	Deptno
Nidhi	6258	Clerk	900	20
Aastha	6388	Manager	1500	30
Manmeet	6765	Clerk	1050	10
Navreet	6800	Analyst	1100	30

Entity : Employee

Attributes : Ename, Empno, Job, Sal, Deptno

Record : collection of related data i.e. Ename, Empno, Job, Sal, deptno

- In DBMS, each user can view data according to his/ her choice. Two users can use the same portion of data at the same time different forms.
- DBMS is used to create the reports and mathematical functions for the users.

1.4.1 Characteristics of Database Management system

A database management system is designed to define, manipulate, retrieve and manage data in a database. It generally manipulates the data itself, the data format, filed

names, record structure and file structure. It also defines rules to validate and manipulate this data. The modern DBMS has the following characteristics:

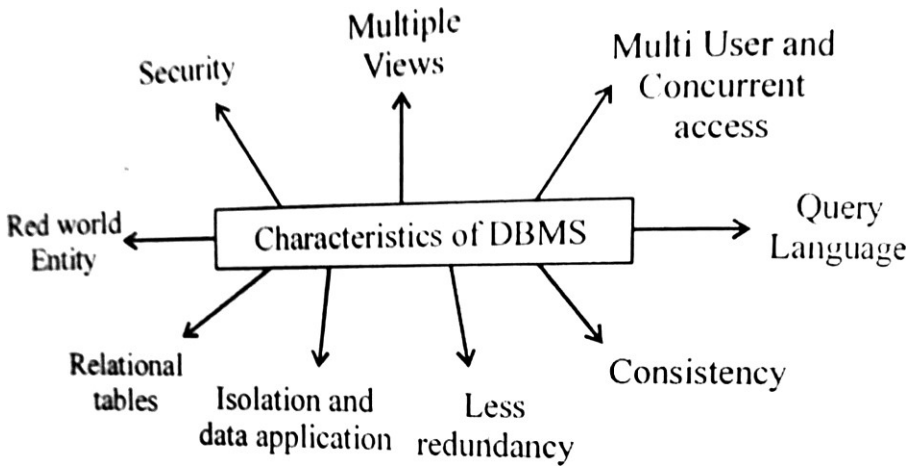
1. **Real world entity:** The DBMS is designed to represent real world entities consist of feature and behavior of real world object. The DBMS have constructs that can easily define real world entity.

2. **Relational tables:** The database contains tables which are mapped with entities. These entity tables are related with other tables to define relational. This eases the concept of data saving. A user can understand the architecture of database just by looking at table names.

3. **Isolation and data application:** The DBMS is designed to isolate data from other complication working of the system as data in preserve into database. The application programs are written to access data

4. **Less redundancy:** DBMS Follows rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Following normalization, which itself is a mathematically rich and scientific process, make the entire database to contain as less redundancy as possible.

5. **Consistency:** DBMS always enjoy the state of consistency where the previous from of data storing application like file processing does not guarantee this. Consistency is a state where every relation database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state.



6. **Query language:** DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many any different filtering options, as her on she wants. Traditionally it was not possible where file-processing system was used.

7. **Multiuser and concurrent access**: DBMS support multi-user environment and allows them to access and manipulate data in parallel. Though there are restriction on transactions when they attempt to handle same data item, but users are always unaware of them.

8. **Multiple views**: DBMS offers multiples view for different users. A user who is in sales department will have a different view of database than a person working in production department. The enables user to have a concentrate view of database according to their requirements.

9. **Security**: Features like multiple views offers security at some extent where users are unable to access data of other user and department. DBMS offers methods to impose constraints while entering data into database and retrieving data at later stage. DBMS offers many different levels of security features, which enables multiple users to have different view with different features, for example, a user in sales department cannot see data of purchase department is one thing, additionally how much data of sales department he can see, can also be managed. Because DBMS offer many different features, for examples, a user in sales department cannot see data of purchase department

is one thing, additionally how much data of sales department he can see, can also be managed. Because DBMS is not saved on disk as traditional file system it is very hard for a theif to break the code.

1.4.2 Operations/Functions of Database Management System

A DBMS is an intermediate between user and database. The DBMS provides multiple useful interfaces to interact with database. There are several function that a DBMS performs to ensure data integrity and consistency of data in the database. The following are some important function of data management system.

1. **Data Dictionary Management:** Data Dictionary is where the DBMS stores definitions of the data elements and their relationships i.e. metadata. It is often hidden from the user and is used by Database Administrators and Programmers. It also shows which program use which piece of database and record.

The DBMS uses different function to look up the required data along with relationships into data Dictionary. Whenever a request is made for a particular data in database then DBMS programs access data dictionary. The function removes structural and data dependency and provides the user with data abstraction.

2. **Data Storage Management** : The data storage management is one of core function of DBMS which is used for the storage of data and any related data entry forms, report definitions, data validation rules, procedural code, screen definition and structures. The DBMS manage data in such a way that users do not need to know how data is stored or manipulated.

3. **Data Transformation and Presentation:** The data transformation and presentation is one of integrated function of DBMS as it is helpful to store data in simple format and display information in uniform format. The function exits to transform any data entered into required data structures. By using the data transformation and presentation function; the DBMS can determine the difference between logical and physical data formats.

4. **Security Management:** The security management is implemented in DBMS at different levels. Security management sets rules that determine specific user that are allowed to access the database. Users are given a username and password or sometimes through biometric authentication. DBMS must monitor user request. It can reject the request which break the security rules.

5. **Data Definition:** DBMS must be able to accept data definition commands. These commands are: create Alter and Drop.

6. **Data Manipulation**: DBMS must be able to handle the request from user to retrieve update and delete data. These commands are: Select, insert, Update and Delete.

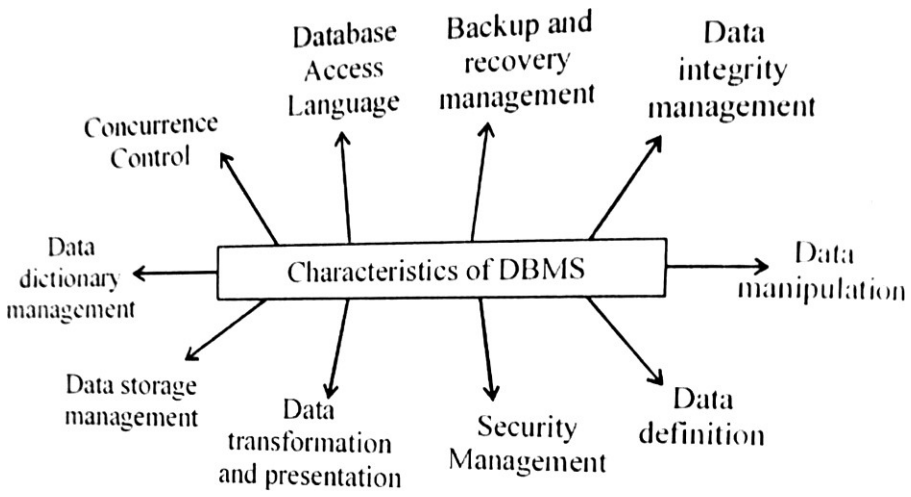


Fig. 1.6: Operations/Function of Database Management System

7. **Data Integrity Management:** Data integrity and data consistency are the core function of DBMS to provide security to data. The DBMS enforces these simultaneously without affecting the integrity of the database. The DBMS enforces these rules to reduce things such as data redundancy, which is when data is stored in more than one place unnecessarily, and maximizing data consistency, making sure database is returning correct/same answer each time for same question asked.

8. **Backup and Recovery Management:** Backup and recovery is done by DBMS to safeguard the old data so that unwanted damage to data can be recovered. Backup management refers to the data safety and integrity; for example backing up

document files. Similarly recovery of data can be implemented to go back to check the previous status of the data. DBMS software component ― transaction manager‖ is used to recover the data which was lost due to some misshapen.

9. **Database Access Languages and User Interfaces:** The DBMS provides multiple user interfaces to meet different requirements of the end user in different network environments DBMS may provides different terminals, web interfaces, etc. DBMS also provides user interface to interact with database. It is not feasible to provide everything in the form of drop down menu so DBMS supports a SQL (structured query language) language which is a non procedural language. The use of SQL language makes it easy for user to seek information according to the requirement. User can seek information by providing command to DBMS query processor which arranges data to the user.

10. **Concurrency Control:** since DBMS support sharing of data among multiple users they must provide a mechanism for managing concurrent access to the database. DBMS ensure that the database kept in accurate state.

1.4.3 Advantages of Database Management System

1. **Minimal Redundancy/Eliminate Duplication:** In non-database system each application program has its own private files. In this case, the duplicated copies of the same data are created in many places. In DBMS, all data of an organization is integrated into a single database file. The data is recorded in only one place in the database and it is not duplicated. Centralized control of unnecessary duplication of data. It also reduces the total amount of data storage. It also eliminates the extra processing required to trace the results.

2. **Data Integrity:** In database management system, data in database is stored in tables. A single database contains multiple tables and relationships can be created between tables (or associated data entities). This makes easy to retrieve and update data. Data store in database is accurate and consistent. Integrity of data means that data in

database is always accurate, such that incorrect information cannot be stored in database. If the system "crashes", we can retrieve the data easily

3. **Improved Data Consistency:** By controlling the data redundancy, the data consistency is obtained. If a data item appears only once, any update to its value has to be performed only once and the updated value is immediately available to all users. If the DBMS has controlled redundancy, the database system enforces consistency.

4. **Data in Shared Form:** In DBMS, data can be shared by authorized users of the organization. The database administrator manages the data and gives rights to users to access the data. Many users can be authorized to access the same piece of information simultaneously. The remote users can also share same data. Similarly, the data of same database can be shared between different application programs. The DBMS allows the sharing of data under its control by any number of application, programs or users. There is no need to insert the data separately by each department or user. Data enter by one user can be share by all the users.

5. **Enforcement of Standards:** DBMS is enforced laws and policies in the form of standards which helps in maintaining database. Data is stored according to the standards and in uniform pattern the common standards can be implemented to all databases.

6. **Data Security:** Data must not be accessed by unauthorized persons with the help of DBMS we can ensure that proper access procedures can be implemented. Different level of security could be implemented for various types of data and operations.

7. **Solving Enterprise Requirement than Individual Requirement:** The DBMS is designed for general purpose and it is developed that person with different technical skills can use it as per the requirements. Since many types of users with varying level of technical knowledge use a database, a DBMS should provide a variety of user interface. The overall requirements of the enterprise are more important than the individual user requirements. So the DBA (Database Administrator) O can structure the database system to provide an overall service,

8. **Providing Backup and Recovery:** The DBMS provide the backup and recovery system so that data can be stored for future. It is also stored to manage the unwanted lose Similarly recovery policies are used which is automatically create the backup of data and restore data if required A DBMS must provide facilities for recovering from hardware or software failures.

9. **Cost of Developing and Maintaining System is Lower:** The cost of using DBMS is low as it requires basic resources which are generally available in the organization. The cost involved in developing and maintaining the whole system is low whoever shitting manual data to electronic data may include labor cost which is addition to system cost.

10. **Concurrency Control:** The DBMS have control over the concurrent access to database It provides a common interface to perform access to database. The concurrent users may access data at same time being sharing by the DBMS. The DBMS designed polices such that one user access data other cannot perform updated task. At a time only one user is allowed to perform update operation other user need to wait for update the common data.

11. **Flexible System:** DBMS is a flexible system as it is designed for general purpose. The DBMS can handle small to large database and it can be redesigned to meet the requirements of the users. Moreover Database used by one system can be transformed into another system.

12. **Better Services to the User:** Because data are integrated into a single database, complex requests can be handled much more rapidly, then if the data were located in separate, non-integrated files. In many businesses, faster response means better customer service.

13. **Tools for Report Writing:** Most of the DBMS provide the report writer tools used to create reports. The users can create very easily and quickly. Once a report is created, it can be used may times and it can be modified very easily. The created reports are also saved along with database and behave like a software component.

14. **Controlled Concurrency:** In a computer file-based system, if two users are allowed to access data simultaneously, it is possible that they will interfere with each other. For example, if both users attempt to perform update operation on the same record, then one may overwrite the values recorded by the other. Most database management systems have sub-systems to control the concurrency so that transactions are always recorded with accuracy.

15. **Application Program/Data Independence:** The separation of data structure of database from the application program that uses the data is called data independence. In DBMS, we can easily change the structure of database without modifying the application program. Data is independent from one level to another level.

16. **Improved Decision Making Process:** Better-managed data and improved data access make it possible to generate better-quality information, on which better decisions are based. The quality of the information generated depends on the quality of the underlying data. Data quality is a comprehensive approach to promoting the accuracy, validity, and timeliness of the data. While the DBMS does not guarantee data quality, it provides a framework to facilitate data quality initiatives.

17. **Improvement in End-User Productivity:** The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.

18. **Application Development Ease:** The application programmer need not build the functions for handling issues like concurrent access, security, data integrity, etc. The programmer only needs to implement the application business rules. This brings in application development ease. Adding additional functional modules is also easier than in file-based systems.

19. **Data Atomicity:** A transaction in commercial databases is referred to as atomic unit of work. For example, when you purchase something from a point of sale (POS) terminal, a number of tasks are performed such as;

- Company stock is updated.

- Amount is added in company's account.
- Sales person's commission increases etc.

All these tasks collectively are called an atomic unit of work or transaction. These tasks must be completed in all, otherwise partially completed tasks are rolled back. Thus through DBMS, it is ensured that only consistent data exists within the database.

20. **No Data Isolation:** Data is stored in uniform format so there is no need to make different programs for each data.

21. **Advanced Capabilities:** DBMS also provides advance capabilities for online access and reporting of data through Internet. Today, most of the database systems are online. The database technology is used in conjunction with Internet technology to access data on the web servers

1.4.4 Disadvantages of Database Management System

In contrast to the lots of advantages, there are few disadvantages as well which are discussed below:

1. **Complexity:** The database designed is no of the major challenging task in DBMS as it is complex, difficult, and time-consuming. DBMS is an extremely complex piece of software database designer's developers DBA and end users must understand this functionality to take full advantage of it. Failure to understand the system can lead to bad design decisions.

2. **Large Size:** The complexity and breadth of functionality makes the DBMS an extremely large piece of software, occupying many megabytes of space and requiring huge amounts of memory to sun efficiently

3. **Performance:** File Based system is written for a specific application a result performance is generally very good. However, the DBMS is written to be more general to cater for many applications rather than just one. So performance is very poor.

4. **Higher Impact of Failure:** In most of the organizations, all data is integrated into a single database. If database is corrupted due to power failure or it is corrupted on the storage media, then our valuable data may be lost or whole system stops.

All users and applications rely on the availability of the DBMS the failure of any component can bring operations to a hall.

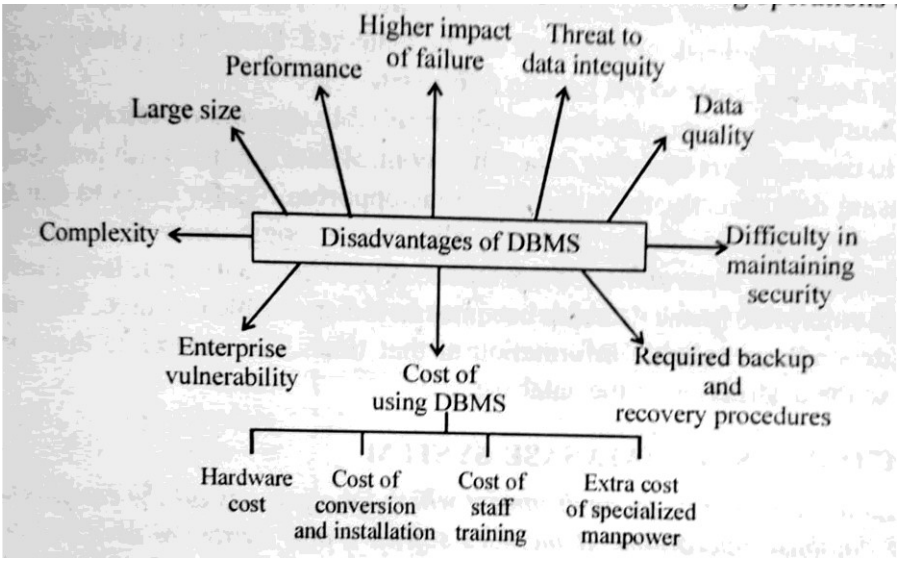


Fig. 1.7: Disadvantages of Database Management System

5. Cost of using DBMS

(a) **Hardware Cost:** If we want to implement DBMS then we need DBMS software which is very expensive. We need to upgrade the hardware the processing overheads to implement the security, integrity and sharing of data make the additional cost.

(b) **Cost of Conversion and Installation:** DBMS vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new features. Cost of DBMS and extra hardware may be insignificant compared with the cost of converting existing applications to run on the new DBMS and hardware. This cost includes cost of training staff to use these new systems and employment of specialist staff for help. That is way some organisations feel tied to their current systems and cannot switch to modern database technology.

(c) **Cost of Staff Training:** DBMS is a complex system which demands specialized users. The user need to get training which ultimately added to the total coast.

(d) **Extra Cost of Specialized Manpower:** The DBMS is managed by skilled people includes DBA, Programmer, and data entry staff.

6. **Threat to Data Integrity:** Database is shared among different users and concurrent access is permitted in DBMS. So there is always a threat to data integrity, especially when there is transition failure.

7. **Difficulty in Maintaining Security:** Data is reserved in common place and different users are accessing data with different security levels. The user management and security access is a challenging task which requires attention. Access policy design for secure access is difficult to maintain.

8. **Required Backup and Recovery Procedures:** The database need to be backed up in time so that unwanted risk of data lose can be managed. DBMS requires special extra hard disk space and special place to put backup of old data.

9. **Data Quality:** Since the database is accessible to users remotely, adequate controls are needed to control users updating data and to control data quality. With increased number of users accessing data directly, there are enormous opportunities for users to damage the data. Unless there are suitable controls, the data quality may be compromised.

10. **Enterprise Vulnerability:** When DBMS is used in an enterprise level then centralizing all data of an enterprise in one database becomes an indispensable resource. The survival of the enterprise depends on reliable information at that time. The enterprise therefore becomes vulnerable to the destruction of the database.

1.5 COMPONENTS OF DATABASE SYSTEM

The database system is an environment which incorporates all the components required to execute database operations. It includes software and hardware used in functioning of database.

The database system is designed to provide an environment that is both convenient and efficient to perform different task on data. There are five major components in the database system environment which are as follows:

1. ***Data (Data, Metadata)***
2. ***Software (DBMS and Application Programs)***
3. ***Hardware***
4. ***Users***
5. ***Procedure***

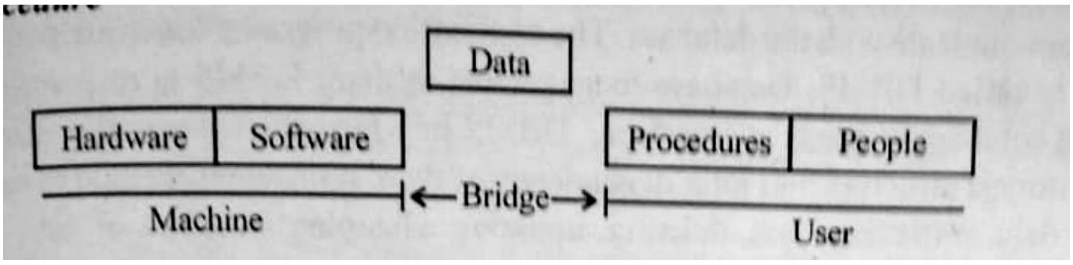


Fig. 1.8: Components of Database System

The following diagram shows these components and interaction with each other.

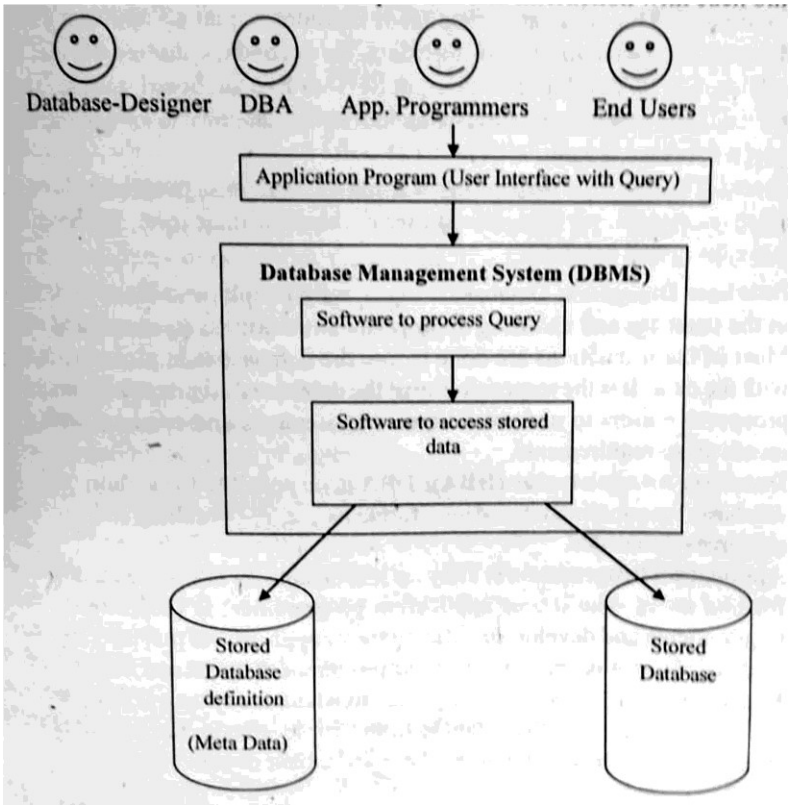


Fig. 1.9: Database System Environments

1. **Data (Data, Meta data):** The data is a fact and figure which is impotent to be stored in database. A database represents data and method used to preserve data in database. It designed to store data in such as way so that it can easily sharable and is being integrated. It also contains meta-data which is an information stored in catalo about the data. It is also called data about data or data dictionary which means it has information about data which stored in database. It defines the definition and representation information of data.

2. **Software (DBMS and Application Programs):** The software is actual DBMS. All requests from users for access to the database are handled by the DBMS. DBMS allows the users to communicate with the database. The application programs which are designed to handle database is called DBMS, Database management system. DBMS is responsible for smooth work with database and acts as interface. DBMS interface shields complex detail of data like physical storage structure and inter dependency of data. It provides method to handle data like insertion data, retrieving data, deleting, updating, changing structure of data, etc. It is also categorized into two components: Software to process Queries and Software to access stored data. The Software to process Queries deals with user interface and interaction to sort the query raised by the user whereas software to access data acts as an interface with physical database.

3. **Hardware:** The hardware component includes actual computer hardware used for keeping and accessing the database. To store data, the secondary storage devices are used such as Magnetic Disk, CD/DVD. Input and output devices like keyboard, mouse, scanner reader, monitors are used. Data processing hardware includes computer processor which plays significant role to support a database system.

4. **Users:** The users are person who is using the system as per their role or requirement. In typical database system, we categorized user on basis of their role. We have four types of users which are discussed below:

- **Database Designers:** Database designers are people who identify data to be stored in the database and choosing appropriate structures to represent and store the data Most of these functions are done before the database is implemented and populated with the data. It is the responsibility of the database designers to communicate with all prospective users to understand their requirements and come up with a design that meets these requirements.
- **Database Administrator (DBA):** DBA is responsible for authorizing access to the database, for coordinating and monitoring its use, and acquiring software and hardware resources as needed.
- **Application Programmers:** They are responsible for developing application program for end users. The job of application programmers is to determine the end use requirements and develop specifications for applications that meet the requirements. They implement the specifications as programs, then test and debug the programs.
- **End Users:** End users are those people to whom the system is designed. These use are actually accessing the database from their terminals. The end users are classified on the basis of knowledge of database and extent of use.

User	Role of User
Naive users	Naive users access the database through application programs that have been written by application programmers. These users don't have technical details of the database and its structures. The use system manual for accessing database. The example s of Naive users are: Operator in hotel for reservation, railway or airline clear, etc.
Casual User	Casual users are occasionally access database and use online query to fetch data from database. They have knowledge of query language and use it from their terminal to fetch data.
Sophisticated Users	These users interact with database without writing any program, use stand query to seek information in database. Examples of engineers, scientists, analysts who implement applications to meet their requirements.
Standard User	They interact with system with the help of menu driven interface. They don't have technical information of the database and use of query is minimum.
Specialized Users	Specialised user is system expert and hardcode professionals. They develop their own application program for system like expert system, knowledgebase system.

Table 1.3: Different End Users and their Rolls

The database system has different users as listed above, but role of DBA is significant in many aspects. We discuss the role of DBA in 1.6 in detail.

5. **Procedures:** Procedures refer to the instructions (rules) that govern the design and use of the database. The users of the system and the staff that manage the database require documented procedures to use or run the system. The followings are some instructions so that we can follow procedure systematically:

(i) Log on to the DBMS

(ii) Start and stop the DBMS

(iii) Make backup copies of the database

(iv) Handle hardware or software failures.

(v) Change the structure of a table.

1.6 DBA (DATABASE ADMINISTRATOR)

- The DBA is a person or group of persons who control the right to access to the data and over all maintain policies to ensure softy and smoothly working of the system.
- DBA controls the design and use of database. DBA is responsible for implementing the database system within an organization.
- DBA provides a necessary technical support for implementing policy for smooth working of the database.
- DBA is responsible for evaluation, selection and implementation of DBMS package.
- The DBA has to perform number of important task like authorizing access to database, coordinating and monitoring different types of users, handling software and hardware issues.
- Database Administrator's job requires a high degree of technical expertise.
- In practice, the DBA may consist of team of people rather than just one person.

Functions Responsibilities of Database Administrator (DBA)

The database administrator performs a critical role within an organization and has to perform different functions and responsibilities Depending on the organization and the department, the role DBA can either be highly specialized or incredibly diversified. The functions of DBA are as follows:

1. **Defining conceptual schema and database creation:** The DBA is responsible for designing conceptual schema of database. The DBA defined how data is to be represented in the database and how tables are related to each other.

2. **Storage structure and access method definition:** The DBA is responsible to define storage structure and provide access methods to database. The DBA defines access policies to an individual or a group. The DBA also decide how the data is to be represented in the database.

3. **Defining integrity constraints:** The DBA defines integrity rules to ensure the accuracy of the data. The integrity rules are defined according to the nature and requirement of the users. The DBA defines the checks and integrity policies so that users can access data with freedom as checks and integrity constrains do not allow illegal operations.

4. **Ensuring availability of data:** The DBA is ensured that whenever request is made for data, data should be available. The availability of data around the clock is possible with appropriate steps to take backup and switching load among different systems. The DBA defines policies such that data must be available in time.

5. **Deciding backup and recovery methods:** The backup and recovery methods are very crucial for the safety of the data as system accidental failure can happen any time. The DBA decides which data is to be backed up and when. DBA defines policy for backup so that data loss can be avoided.

6. **Granting authorisation to the users:** The DBA defines list of users with access level so that data can be access by the authorised users only. The authorisation of user is monitored and updated by DBA time to time as one user may changes his access level.

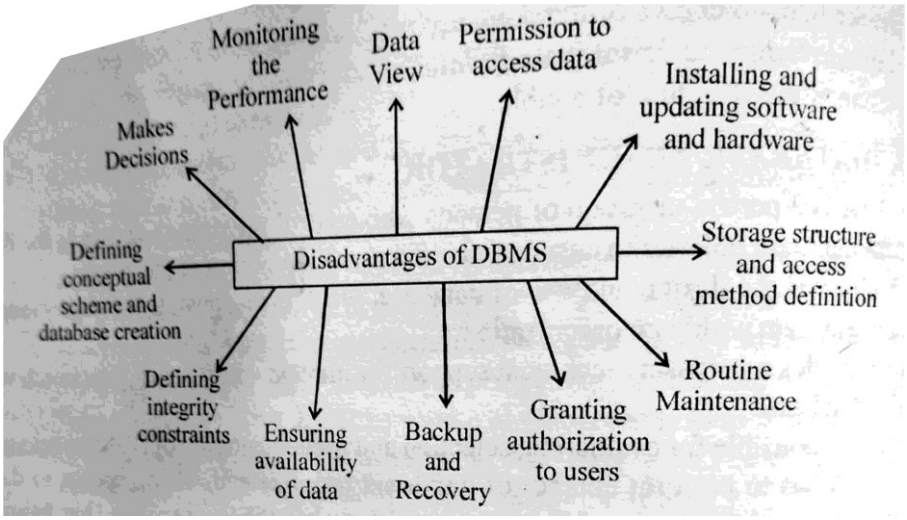


Fig. 1.10: Responsibilities of DBA

7. **Routine maintenance:** The routine maintenance includes up gradation of system, updating user profiles and other information regarding access policies. The Database Administrator understands the following routine maintenance activities:

- ***When transaction rollbacks occur*** DBA decides what to do when a transaction rollback. If such incidence occurs, then DBA checks the updating records and decide about re-do or undo of the transaction.
- ***When the database is out of system disk space:*** The database is stored into physical memory which in limited in size. When the data is about to reach maximum limit in disk space then DBA decide whether to erase some unwanted data or add new hard disk into system.
- ***When unique constraints have been violated:*** The access to data in database in done through some unique constrains whenever such rule is violated then DBA has to look into the matter The unique constraints policies are updated as their reports of violation such that violations can be avoided in future.
- When not to shut down the database while the application is running

8. **Installing and updating software and hardware:** The DBA is the person who is authorised to installed software into the database machine. The DBA decides the need of updating the hardware

9. **Permission to Access Data:** DBA gives permission to user to use database. Only authorised user can access data.

10. **Data view:** DBA can create different views of data that can be shown to different users.

11. **Monitoring the performance.** DBA is responsible for overall performance of the system. To improve the performance DBA regular monitor the system performance.

12. **Makes Decisions:** It is the DBA's job to decide exactly what information is to be held in the database.

1.7 COMPARISON OF FILE MANAGEMENT SYSTEM WITH DATABASE MANAGEMENT SYSTEM

Sr. No.	Concepts	File Management System	Database Management System
1.	Redundancy	Data redundancy (duplication) is possible. Data duplication is a commonly visible in file management system as multiple files are stored at different location.	Data redundancy (duplication) is not possible. In database management system, minimum redundancy occurred as duplication is avoidable.
2.	Consistency	Data is duplicated into number of files and consistency is a hard job in file management system. Consistency ensures that data at different location about single entity should be	Database management system avoids duplication as a result consistency can be avoided. Moreover data

		same with time.	
3.	Data Isolation	The data isolation is preserve in the file management system as data is stored in different file which are hardly associate with each other.	The data is stored into tables which are linked with each other so the isolation of data is not available.
4.	Standards enforcement	In an organization, every department has their own file system and format so uniform format and standards cannot be implemented. Moreover application programs are file dependent so new standard enforcement is hard to implement.	DBMS is enforced laws and policies in the form of standards which helps in maintaining database. Data is stored according to the standards and in uniform pattern so any programs are file dependent so new change in standard policies do not affect the working of the database.
5.	Data security	Data security is implemented at file level, whereas user level security itself is not feasible.	Data security is implemented at different levels and data security policies are upgradable to meet

			the requirements.
6.	Application dependency	The applications are developed according to the data file, it means we have to make certain changes whenever application applied on different format data file.	The application programs in DBMS are independent on database. Data independency is enforced in the system which helps the programmer to write general purpose programs
7.	Multiple Access	The multiple access to data file is not permitted in file management system.	In database management system, multiple access is permitted.
8.	Concurrency Problem	There is concurrency problem in file management system as multiple accesses is not desirable.	DBMS is designed to meet concurrent access which means more than one user can access same data without any problem.
9.	Technical platform	In file management system, every department developed their own applications to access data files. Generally applications are developed	DBMS provides integrated program kit which is developed using common language. Examples of DBMS are Oracle,

		in C, C++, COBOL etc.	Sequel & Foxpro etc.
10.	Real world modelling	Real world modelling is not possible as files management system stored data in files. The real world modelling requires object based data representation which is not possible in file based system.	Real world modelling is done through object representation in DBMS as data along with other attributes can be stored.
11.	No. of Files	There are less number of files as compared to DBMS	There are more number of files.
12.	Cost	It is cheaper as compared to DBMS.	It is costly
13.	Structure	It has simple structure	It is complex structure.
14.	Flexibility	It is less flexible as compared to DBMS.	It is more flexible.
15.	Efficiency	When the volume of data increases, its efficiency decreases	Volume of data not affect its working capability.

1.8 CATEGORIES OF DBMS

The classification of a database management system (DBMS) is greatly influenced by the underlying computing system on which it runs, in particular of computer architecture such as parallel, networked or distributed. However, the DBMS can be classified according to the number of users, the database site locations and the", expected

type and extent of use.

1.8.1 Centralized DBMS

1. In centralized .database systems, the database system, application programs, and user-interface all are executed on a single system and dummy terminals are connected to it.
2. It is physically confined to a single location.
3. The processing power of single system is utilized and dummy terminals are used only to display the information.
4. As the personal computers became faster, more powerful, and cheaper, the database system started to exploit the available processing power of the system at the user's side, which led to the development of client/server architecture,
5. In client/server architecture, the processing power of the computer system at the user's end is utilized by processing the user-interface on that system.
6. The centralised database system .consists of a single processor together with its associated data storage devices and other peripherals.
7. The system offers data processing capabilities to users who are located either at the same site, or, through remote terminals, at geographically dispersed sites.-
8. The management of the system and its data are controlled centrally form any one or central site.

The following diagram 2,6 show the centralized DBMS.

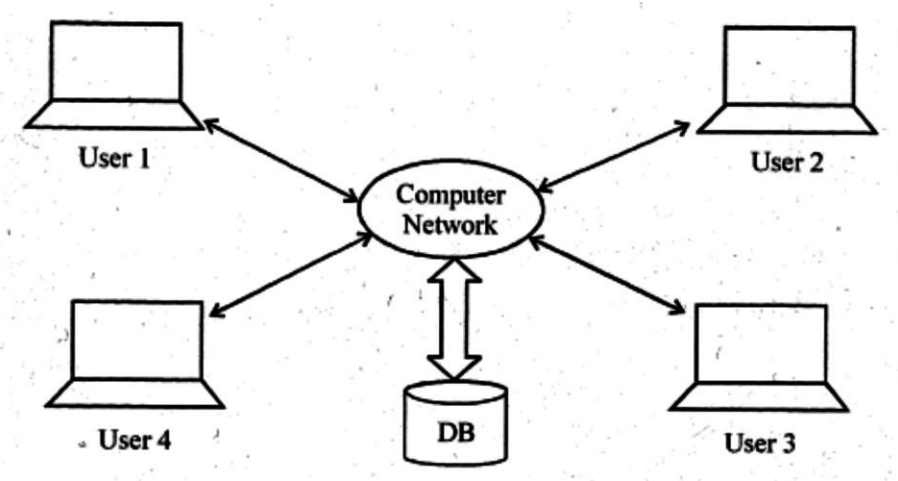


Fig. 1.11: Centralized DBMS

Advantage of a Centralized DBMS

1. **Centralized control:** The organization can exert centralized management and control over the data by Database Administrator (DBA). The database administrator is the focus of centralized control./
2. **Shared data:** A database allows the sharing of data under its control by any number of application programs or users.
3. **Reduction of redundancies: Centralized** control of data by DBA avoids unnecessary duplication of data and effectively reduces the total amount of data storage required. It also eliminates the extra processing necessary to trace the required data in a large mass of data.
4. **Integrity:** Centralized control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity. Data integrity means that the data contained in the database is both accurate and consistent. Therefore data values being entered for storage could be checked to ensure that they fall within a specified range and in the correct format.
5. **Security:** Data is a vital importance to an organization and may be confidential. Such confidential data must not be accessed by unauthorized person.
6. **Data Independence:** Data independence allows dynamics changes and growth potential.
7. **Operations:** Most of the functions such as update, backup, query, control access and so on, are easier to accomplish in a centralised database system.
8. **Size of the Database:** The size of the database and the computer on which it resides need not have any bearing on whether the database is centrally located.

Disadvantage of a Centralized DBMS

1. **Problems associated with centralization:** Several problems are associated

with centralization like networking the excessive load on the system at the central site would likely causes all accesses to be delayed etc.

2. **Cost of software and migration:** The cost of purchasing or developing the software, the hardware has to be upgraded to allow for the extensive programs and the work spaces required for their execution and storage. The processing overhead is also added by implement security integrityof data causes a degradation of the response and through put times. It is also added the cost of migration from .a traditionally separate application environment to an integrated one.
3. **Complexity of backup and recovery:** The centralization reduces duplication, the lack of duplication required that the database be adequately backed up so that in the case of failure the data can be recovered. Backup and recovery operations are fairly complex in a DBMS environment.
4. **Server Down:** When the central site computer or database system goes down, then every user is blocked from using the system until the system comes back.
5. **Communication costs:** The communication costs from the terminals to the central site can be expensive.

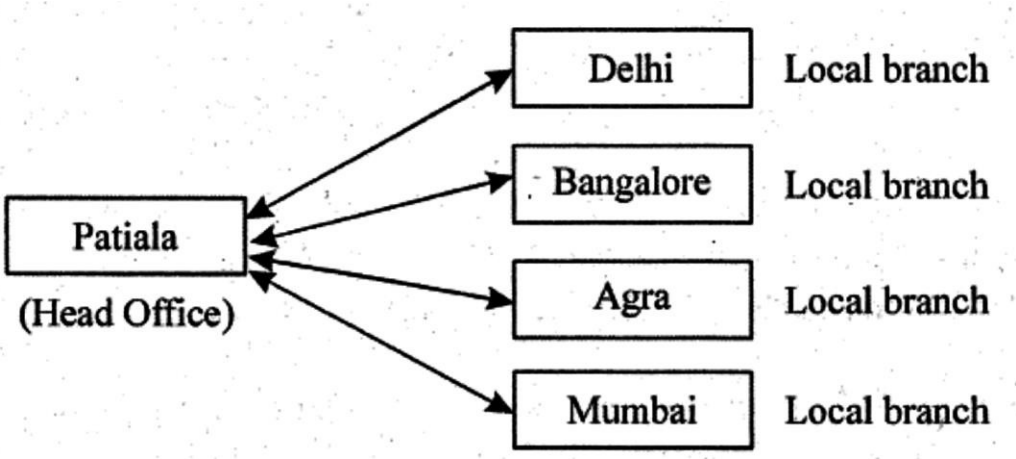
1.8.2 Parallel DBMS

1. Parallel database systems architecture consists of a multiple central processing units (CPUs) and data storage disks in parallel.
2. They improve processing and input/output (I/O) speeds.
3. Parallel DBMS are used in the applications that have to. query extremely large databases or that have to process an extremely large number of transactions per second.

The following diagram shows the parallel DBMS.

- Shared data storage disk
- Shared memory
- Hierarchical

- Independent resources



Advantages of-a Parallel DBMS.

1. Parallel database systems are very useful for the applications that have to query extremely large databases.
2. In a parallel database system, the throughput and the response time are very high. Throughput is number of tasks completed in -given time duration. Response time is among of time required by single task for completion.

Disadvantages of a Parallel DBMS

1. In a parallel DBMS, there is a .startup cost associated with initiating a single process and the startup-time may overshadow the processing time, affecting speedup adversely.
2. In parallel DBMS, the processes access the shared resources which slow down the result.

1.8.3 Distributed DBMS

1. Distributed DBMS consist soft of a single logical database that is spilt into number of fragments.
2. Distributed database systems are similar to client/server architecture in a number of ways.
3. Both typically involve the use of multiple computer systems and enable users to access data remote system.

4. Distribute database system broadens the extent to which data can be shared well beyond that which can be achieved with the client/server system.

Following diagram 2.8 shows the distributed DBMS architecture.

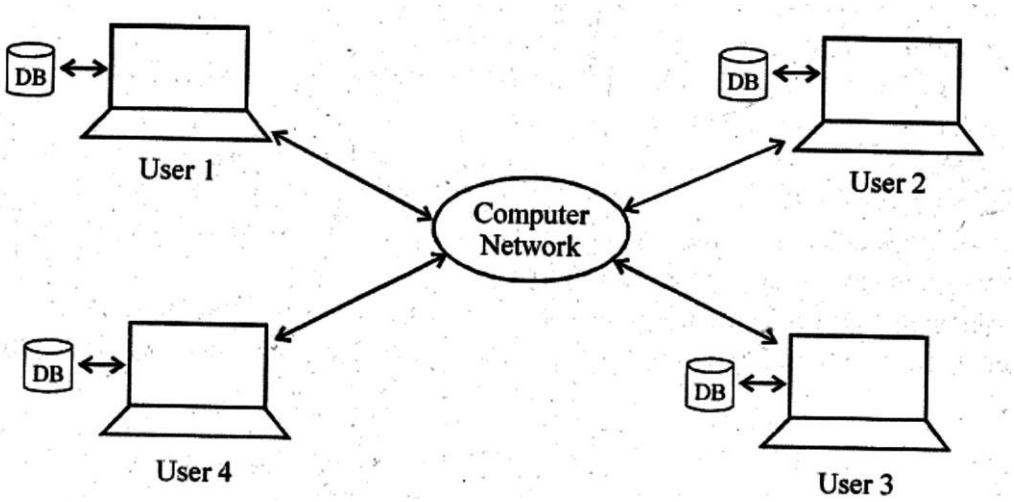


Fig. 1.12: Distributed DBMS

Advantages

1. **Efficiency and better Performance:** Distributed database architecture provides greater efficiency and better performance.
2. **Response time:** The response time and throughput is high as data is available at different places.
3. **Custom-built Machine:** The server database machine can be custom-built or tailored to the DBMS function and thus can provide better DBMS performance.
4. **Customized user interface:** The client application-database might be a personnel workstation tailored to the needs of the .end users and thus able to provide better interfaces, high availability, faster responses and overall improved ease of use to the user.
5. **Shearing of Database:** A single database on server can be shared across several distinct client application systems.
6. **Adding new location:** It causes less impact on ongoing operations when adding new locations. As data volumes and transaction rates increase, users

can grow the system incrementally.

7. Local autonomy: Distributed database system provides local autonomy.

Disadvantage of Distributed DBMS

The recovery from failure is more complex in distributed database systems than in centralized systems.

1.8.4 Client/Server Database System

1. Client/server architecture of database system has two logical components namely client, and server.
2. Clients are generally personal computers or workstations whereas server is large workstations, mini range computer system or a mainframe computers system.
3. The server computer is called backend and the client's computer is called front-end. These server and client computers are connected into a network.
4. The applications and tools of DBMS act as clients, making requests for its services.
5. DBMS software resides on the server.
6. The DBMS, in turn, processes these requests and returns the results to the client.
7. The client/server architecture is a part of the open systems architecture in which all computing hardware, operating systems, network protocols and other software are interconnected as a network and work in concert to achieve user goals.
8. It is well suited for online transaction processing and decision support applications, which tend to generate a number of relatively short transactions and require a high degree of concurrency.

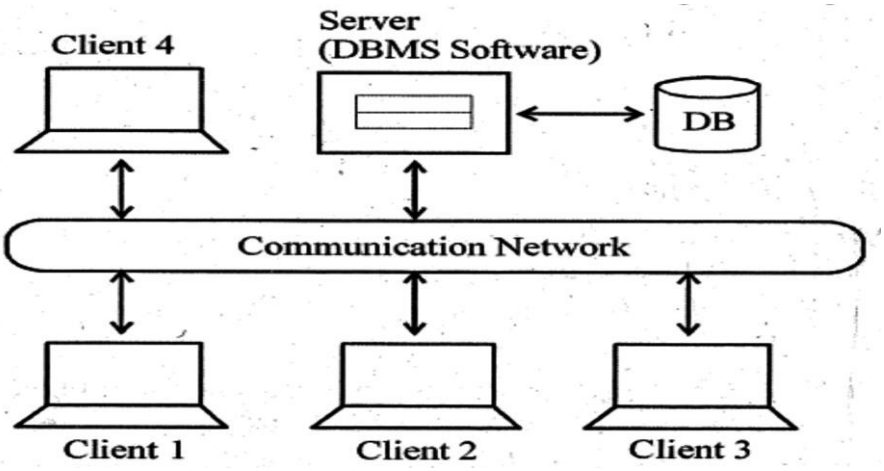


Fig. 1.13: Client Server DBMS

As shown in Fig. 1.13, the client/server database architecture consists of three components namely, client applications, a DBMS server and a communication network interface. The client applications may be tools, user-written applications or vendor-written applications. They issue SQL statements for data access. The DBMS server stores the related software; processes the SQL statements and returns results. The communication network interface enables client applications to connect to the server, send SQL statements and receive results or error messages or error return codes after the server has processed the SQL statements. In client/server database architecture, the majority of the. DBMS services are performed on the server.

Advantages of Client/Server DBMS

1. **Less expensive:** Client-server system has less expensive platforms to support^ applications that had previously, been running only on large and expensive mini or mainframe computers.

2. **Menu-drive interface:** Clients offer icon-based menu-driven interface, which is superior to the traditional command-line, dumb terminal interface typical of mini and mainframe computer systems.

3. **Flexible and productive environment:** Client-server database system is more flexible as compared to the centralized system. Client/server environment facilitates in more productive work by the users and making better use of existing data.

4. **Response time and throughput:** The client server model is based on

request and reply model when a machine make request to the server then server immediately reply so Response time and throughput is high.

5. **Custom-built Servers:** The database server machine can be custom-built or tailored to the DBMS function-and thus can provide a better DBMS performance.

6. **Custom build Client machine:** The client application database might be a personnel workstation, tailored to the needs of the end users and thus able to provide better interfaces, high availability, faster responses and overall improved ease of use to the user.

7. **Powerful Single Server:** A single database on server can be shared across several distinct client application systems.

Disadvantages of Client/Server DBMS

1. **High set cost:** The setup cost is high which include labour or programming, cost is high in client/server environments, particularly in initial phases.

2. **Lack of management tools:** There is a lack of management tools for diagnosis, performance monitoring and tuning and security control, for the DBMS, client and operating systems and networking environments.

Questions

1. What do you mean by data? How is it different from information, explain by example?
2. What is database system? What are four components of database system?
3. What are advantages of database system?
4. What is DBMS? What are the advantages and disadvantages offered by such system?
5. What are the main responsibilities of DBA? Explain.
6. What do you mean by file system? Explain it limitations.
7. Compare file management system with database management system.

UNIT 2: DBMS ARCHITECTURE

2. INTRODUCTION

2.1 THREE LEVEL ARCHITECTURE OF DBMS

- **2.1.1 OBJECTIVES OF ARCHITECTURE**
- **2.1.2 EXTERNAL LEVEL/EXTERNAL VIEW**
- **2.1.3 CONCEPTUAL LEVEL/COMMUNITY USER VIEW/LOGICAL LEVEL**
- **2.1.4 INTERNAL LEVEL/STORAGE VIEW/PHYSICAL LEVEL**
- **2.1.5 DATABASE SCHEMA AND DATABASE INSTANCE**
- **2.1.6 MAPPING BETWEEN DIFFERENT VIEWS**

2.2 EXAMPLE OF THREE LEVEL ARCHITECTURE

2.3 DATA INDEPENDENCE

2.4 DIFFERENCE BETWEEN LOGICAL DATA INDEPENDENCE AND PHYSICAL DATA INDEPENDENCE

2.5 COMPONENTS OF A DBMS

2.6 DATA DICTIONARY

2.7 DBMS LANGUAGES

2.INTRODUCTION

Architecture of Database Management System: *The architecture of DBMS is a framework for describing database concepts and specifies the structure of the database*

system. It describes Junctions of each component and describes how these components communicate with each other in a logical manner.

The database management system Is a sophisticated software application which is designed to provide interface to the user so that user can perform different operations on database with ease. The design of a database management system highly depends on its architecture. It can be centralized or decentralized or hierarchical depending upon the type of applications. Its architecture can be single tier or multi-tier. The multi-tier architecture divides the database management system into related but independent different modules which can be independently modified, altered, changed or replaced. In case of multi-tier architecture, the best suitable architecture is 3-tier architecture.

2.1 THREE LEVEL ARCHITECTURE OF DBMS

Database management system is described in three different levels which have separate functioning and working. These three different levels are named as **external level, conceptual level** and **internal level.** These levels are shown in a serial view of the architecture:

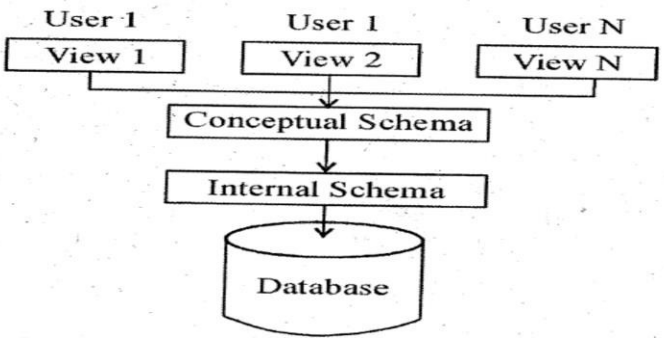


Fig. 2.1: Three Level Architecture (an Aerial view)

2.1.1 Objectives of Architecture

1. It should provide an interface to make changes into the structure of database without changing the application program at external schema.
2. Each user should be able to change the way he view the data and his change should not affect other users.
3. User should not directly deal with the physical database storage.

4. Users are independent of the storage complexities like indexing constraints etc. of the database
6. The conceptual structure of the database has no effect due to the change of the physical storage devices.
7. DBA should be able to change the storage structure and conceptual structure without affecting user's and his view level.

The core objective to design three levels is to provide data independence and physical independence. It is required to provide an easy to use interface to the end users.

2.1.2 External Level/External View

The external level is more concerned with the way in which the data is viewed by individual users. It is closer to the user and provides an interface to interact with the database. Each user has different requirement of the data so DBMS presents each user with a shared or single view or schema of the data. In external level, the different views may have different representations of the same data. For example, one user may view date in the form as (day-month-, year) while another may view as (year-month-day). Similar one view of data may show detail of employee with fields (Name, DOB, Address) and other view may show employee detail with salary (Name, DOB, Basic Pay, HRA, DA). The external view is user specific and provides an abstraction of data which helps to user to view important data and hide additional information.

Characteristics/Functions/Key Points of External Level

1. The external level is at the highest level of database abstraction where only those data is visible to the user which is concerned to the user at that time.
2. External view is user's view of database. It may provide limited and complete access to the database.
3. External schema consists of definition of logical records and their relationships in the external view.

4. External level is also known as view level and closest to the end users. It acts-as an interface to access data. User need not to know the details of data structure and physical storage.
5. External level provides the way in which individual users can view data according to his/her requirements i.e. one user may view data in the form (day, month, year) while another user may view data as (year, month, day).
6. Same database can have different views for different users.
7. Its core purpose is to provide user friendly interface to the end user.

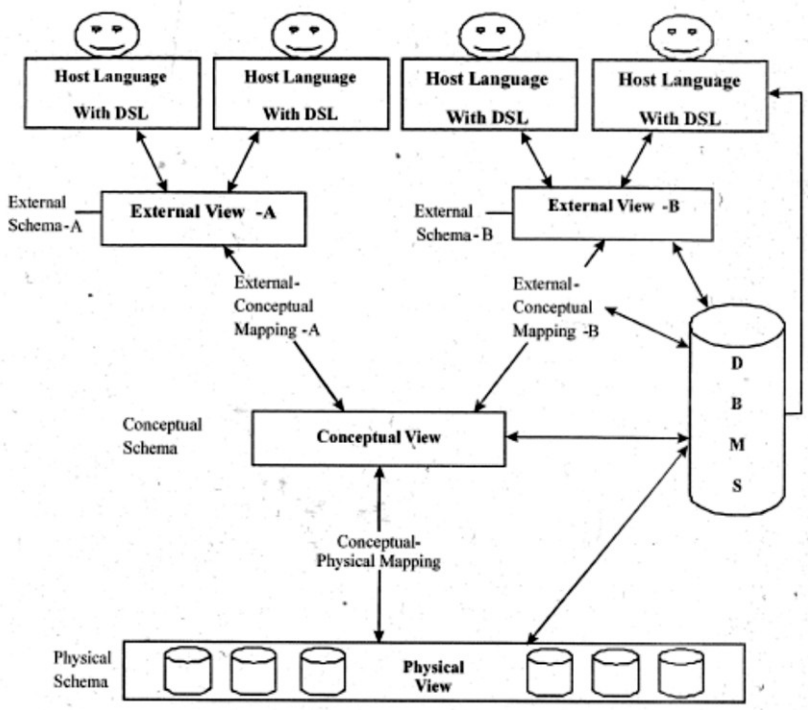


Fig. 2.2: Three Level Architecture with Different Schemas

2.1.3 Conceptual Level/Community User View/Logical Level

Conceptual Level represents the entire database. Conceptual schema describes the records and relationship included in the Conceptual view. The external level is concerned with individual user view whereas the conceptual level represents community user view. The conceptual schema hides the details of physical structure and concentrates on describing entities, data type, relationships, user operations and constraints. The view is

normally more stable than the other two views. The ultimate objective of the conceptual schema is to describe the complete enterprise-not just its data but also how that data is used, how it flows from point to point within the enterprise.

Characteristic/Functions/Key Points of Conceptual Level

1. Conceptual level is also known as middle level. It is created and maintained by DBA.
2. The conceptual schema hides the details of physical structure and concentrates on describing data type entities, their attributes and relationships, user operations.
3. It implements constraint on fie data.
4. At this level, different security and integrity rules can be imposed on data.
5. The semantic information about the data can be represented in conceptual view.
6. Different types of validation checks to retain data consistency and integrity are enforced at conceptual level.
7. It describes what data is stored in database and relationship among database.

2.1.4 Internal Level/Storage View/Physical Level

The internal level is closest to the physical storage which is concerned with the way in which the data is actually stored. The internal view is described by means of the internal schema, which not only defines the various stored record types but also specifies the indexes are in and so on.

Characteristic/Functions/Key Points of Internal Level

1. It is the physical representation of data.
2. It describes how the data is stored in database. It manages storage space allocation for data.
3. It concern with the physical implementation of the database to achieve optimal runtime performance and space utilization.
4. Record description for storage with stored sizes for data items.

5. Access path e.g. specification of primary and secondary keys, index and pointers.
6. Data compression and encryption techniques.
7. Optimization of the internal structures.
8. It builds the indexer, retrieve the data and so on.

2.1.5 Database Schema and Database Instance

While working with any data model, it is necessary to distinguish between the overall design or description of the database (database schema) and the database itself. The database schema is also known as intension of the database, and is specified while designing the database.

1. Schema

A schema is plan of the database that gives the names of the entities and attributes and the relationship among them. A schema includes the definition of the database name, the record type and the components that make up the records. Alternatively, it is defined as a framework into which the values of the data items are fitted. The values fitted into the framework changes regularly but the format of schema remains the same.

Key Points of Schema

- The plan or scheme of the database is known as Schema.
- It gives the names of the entities, attributes and relationship among them.
- It is the framework into which the values of data items are fitted.
- Overall description of database is known as database schema.

Types of Schema

Generally, a schema can be partitioned into/three categories which are as follows:

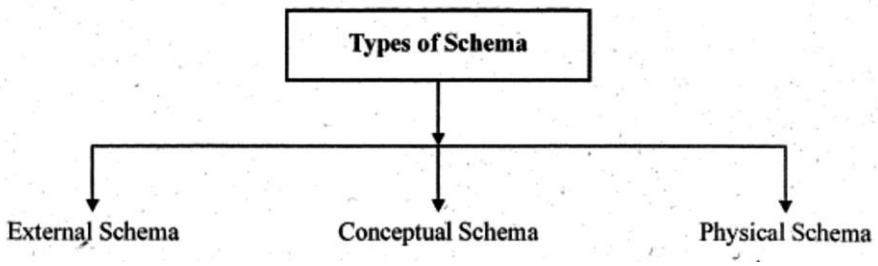


Fig. 2.3: Types of Schema

(a) ***External Schema:*** The external schema is concerned with the description of external view, correspond to different view according to the requirements of the users.

(b) ***Conceptual Schema:*** The conceptual schema is concerned with the description of all the entities, attributes and relationships along with the constraints. The logical (conceptual) schema is concerned with exploiting the data structures offered by the DBMS so that the schema becomes understandable to the computer.

(c) ***Physical Schema:*** The physical schema is concerned with the manner in which the conceptual database gets represented in the computer as a stored database. It is hidden behind the conceptual schema and can usually be modified without affecting the application programs.

2. Subschema

- A subschema is a subset of the schema having the same properties that a schema has.
- It identifies a subset of areas, sets, records, and data names defined in the database schema available to user sessions.
- It allows the user to view only that part of the database that is of interest to him.
- It defines the portion of the database as seen by the application programs and the application programs can have different view of data stored in the database.
- The different application programs can change their respective subschema without affecting other's subschema or view.

3. Instances

- The data in the database at a particular moment of time is called an instance or a database state.

- In a given instance, each schema construct has its own current set of instances. Many instances or database states can be constructed to correspond to a particular database schema.
- Every time we update (i.e., insert, delete or modify) the value of a data item in a record, one state of the database changes into another state.

The following figure shows an instance of the ITEM relation in a database schema.

ITEM		
ITEM-ID	**ITEM_DESC**	**ITEM_COST**
1111A	Disc	30
1112A	Mother Board	500
1113A	CD	100
1144B	Processor	5000

2.1.6 Mapping Between Different Views

Mapping: In three schema architecture, each user group refers only to its own external view. Whenever a user specifies a request to generate a new external view, the DBMS must transform the request specified at external level into a request at conceptual level, and then into a request at physical level. If the user requests for data retrieval, the data extracted from the database must be presented according to the need of the user. *This process of transforming the requests and results between various levels of DBMS architecture is known as mapping.*

The DBMS is responsible for mapping between the three types of schema. Two mapping are required in database systems which are as follows:

(a) **External/Conceptual Mapping:** Each external scheme is related to the conceptual schema by the external/conceptual mapping. The external/ conceptual mapping gives the correspondence among the records and the relationships of the external and conceptual views. A given external record could be derived from a number of conceptual records.

(b) Conceptual/Internal Mapping: Conceptual schema is related to the internal schema by the conceptual/internal mapping. The conceptual/ internal mapping specifies the method of deriving the conceptual record from the physical database.

Advantages of View Mapping

1. Each user is able to access the same data but have a different customized view of the data as per their own needs.
2. A user can change his/her view and this change does not affect other user views.
3. There user's interaction with the database is independent of physical data storage organization.
4. The database administrator is able to change the database storage structure without affecting the user's view.
5. The database administrator is able to change the conceptual structure of the database without affecting all users.
6. The database administrator can change existing storage devices with the new storage devices without affecting others user's.

2.2 EXAMPLE OF THREE LEVEL ARCHITECTURE

To understand the three-schema architecture, consider the three levels of the BOOK file in Online Book database as shown in Figure this figure, two views (view 1 and view 2) of the BOOK file have been defined at the external level. Different database users can see these views. The details of the data types are hidden from the users. At the conceptual level, the BOOK records are described by a type definition. The application programmers and the DBA generally work at this level of abstraction. At the internal level, the BOOK records are described as a block of consecutive storage locations such as words or bytes. The database users and the application programmers are not aware of these details; however, the DBA may be aware of certain details of the physical organization of the data.

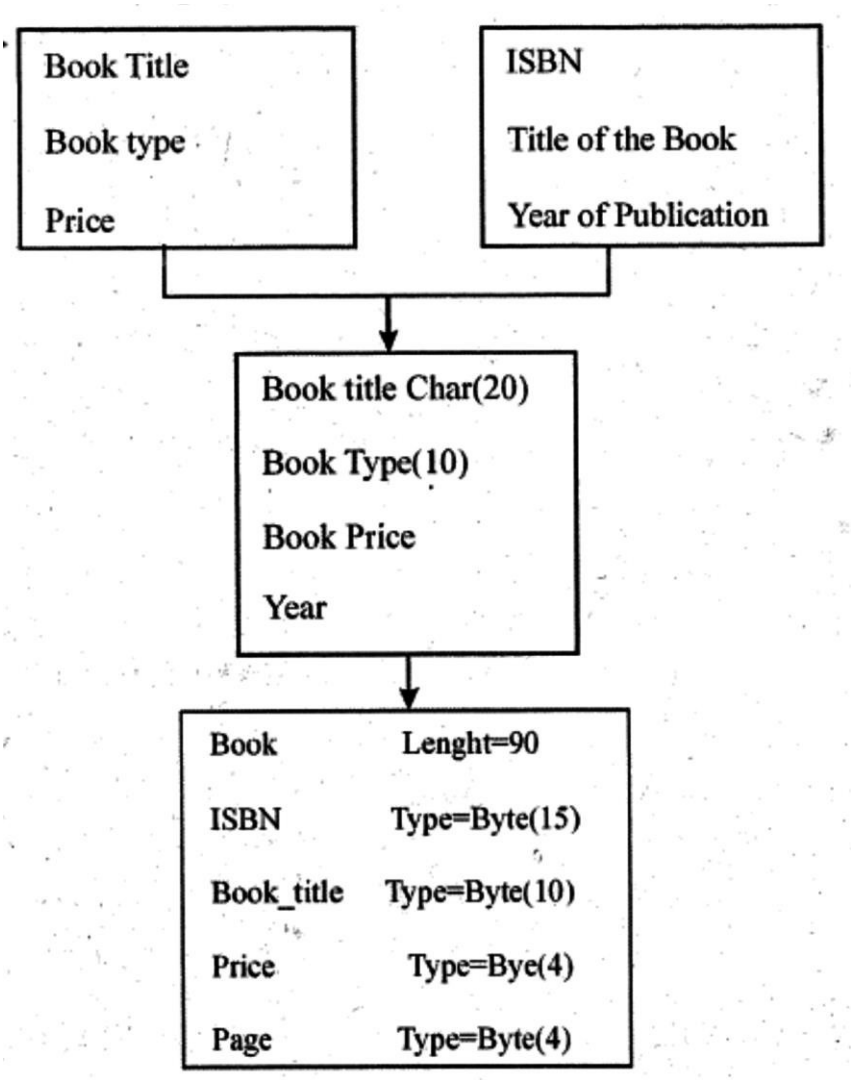


Fig. 2.4: Three Level Schema Architecture

2.3 DATA INDEPENDENCE

Data Independence: *The ability of a database management system to modify its Schema definition at one level without affecting a Schema definition at the next level is called data Independence. It provides flexibility to make changes at one Schema level without affecting the next level Schema.*

Key Points of Data Independence

- The main advantage of three-schema architecture is that it provides data independence.

- Data independence Is the ability to change the, schema at one level of the database system without having to change the schema at the other levels.
- The data independence deals with independence between the way the data is presented, structured and stored. It provides independence to make changes in one level without affecting other levels.
- Data independence means upper levels are unaffected by the changes in lower level.
- For example, DBMS may change the structure of the data without having to change application program. It is possible due to mapping between three levels which enable the user to make changes at one level without affecting the next level of architecture.
- Data Independence implies that the application programs should not need to know any of the following:
  - Ordering of data fields in a record
  - The size of the record
  - The size of the field
  - The format and type of each data item
  - The type of data structured used to store the data.
- The three level DBMS architecture provides two type of data independence. The first is called *logical data independence* and second is called physical data independence. The logical independence is enabled the user to change the conceptual view without affecting the external view. Whereas, Physical data independence is the idea to make changes into internal view without affecting the conceptual or external views. These two types of data independence are discussed in detail below:

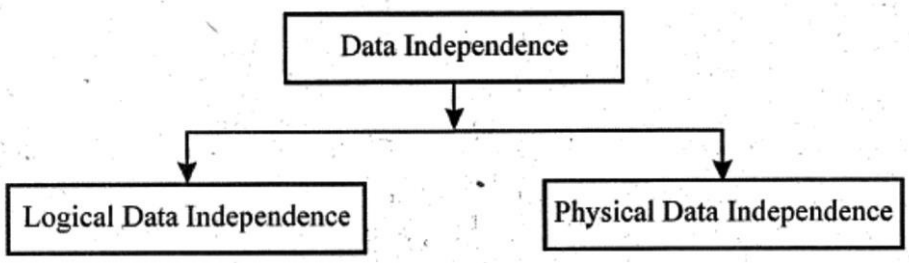


Fig. 2.5: Types of Data Independence Logical Data Independence*

- It is the ability to change the conceptual schema without affecting the external schemas or application programs.
- The conceptual schema may be changed due to change in constraints or addition of new data item or removal of existing data item, etc., from the database.
- The separation of the external level from the conceptual level enables the users to make changes at the conceptual level without affecting the external level or the application programs.
- For Example: The name field in conceptual view is stored as first hame, middle name and last name whereas in external view, it remains to be as a single name field.
- It indicates that the conceptual schema can be changes without affecting the existing external schema.
- It requires the flexibility-in the design of database.
- The programmer is required to make modification in the design as per the requirements.

(b) Physical data independence:

- It is the ability to change the internal schema without affecting the conceptual or external schema.
- An internal schema may be changed due to several reasons such as for creating additional access structure, changing the storage structure, etc.
- The separation of internal schema from the conceptual schema facilitates physical data independence.

- For Example: The location of the database, if changed from C drive to D drive will not affect the conceptual view or external view as the commands are independent of the location of the database.
- It indicates that the physical storage structure used for thedata could be change without affecting the conceptual schema.
- The storage structure and access methods used to retrieve of the data from physical storage medium are not concerned with conceptual schema.

Logical data independence is more difficult to achieve than the physical data independence because the application programs are always dependent on the logical structure of the database. Therefore, the change in the logical structure of the database may require change in the application programs.

2.4 DIFFERENCE BETWEEN LOGICAL DATA INDEPENDENCE AND PHYSICAL DATA INDEPENDENCE

Sr. No.	Logical Data Independence	Physical Data Independence
1.	Whenever, there is a change or modification at the conceptual level without t affecting the user level or external level, it is "known as logical data independence.	Whenever, the changes are made at the internal level without affecting the above layers, it is known as physical data independence.
2.	It is concerned with the structure of the data or changing the data definition	It is concerned with the storage of the data.
3.	It is concerned with the conceptual schema	It is concerned with the internal schema.
4.	Application program need not be change if new fields are added	Physical database is concerned with the change of the storage device
5.	It is very difficult to retrieve the data because data re heavily dependent on the	It is easy to retrieve the data.

	logical structure of data	

2.5 COMPONENTS OF A DBMS

The DBMS accepts the SQL commands generated from a variety of user interfaces, produces query evaluation these plans against the database, and returns the answers.

1. **Query processor:** The query processor transforms users queries into a series of low-level instructions directed to the run time database manager. It is used to interpret the online user's query and convert it into an efficient series of operations in a form capable of being sent to the run time data manager for execution. The query processor uses the data dictionary to find the structure of the relevant portion of the database and uses this information in modifying the query and preparing an optimal plan to access the database.

2. **Run time database manager:** Run time database manager is the central software component of the DBMS, which interfaces with user-submitted application programs and queries. It handles database access at run time. It converts operations in user's queries coming directly via the query processor or indirectly via an application program from the user's logical view to a physical file system. It-accepts queries and examines the external and conceptual schemas to determine what conceptual records are required to satisfy the users request. The run time data manager then places a call to the physical database to perform the request. It enforces constraints to maintain-the consistency and integrity of thewell as its security. It also performs backing and recovery operations. Run time database manager is sometimes referred to as the *database control system* and has the following components:

(i) **Authorization control:** The authorization control module checks that the user has necessary authorization to carry out the required operation.

(ii) **Command processor:** The command processor processes the queries passed by authorization control module.

(iii) **Integrity checker:** The integrity checker checks for necessary

integrity constraints for all the requested operations that changes the database.

(iv) **Query optimizer:** The query optimizer determines an optimal strategy for the query execution. It uses information on how the data is stored to produce an efficient execution plan for evaluating query.

(v) **Transaction manager:** The transaction manager performs the required processing of operations it receives from transactions. It ensures that (a) transactions request and release locks according to a suitable locking protocol and (b) schedules the execution of transactions.

(vi) **Scheduler:** The scheduler is responsible for ensuring that concurrent operations on the database proceed without conflicting with one another. It controls the relative order in which transaction operations are executed.

(vii) **Data manager:** The data manager is responsible for the actual handling of data in the database. This module has the following components:

(a) ***Recovery manager:*** The recovery manager ensures that the database remains in a consistent state in the presence of failures. It is responsible for (a) transaction commit and abort operations, (b) maintaining a log, and (c) restoring the system to a consistent state after a crash.

(b) ***Buffer manager:*** The buffer manager is responsible for the transfer of data between the main memory and secondary storage (such as disk or tape). It brings in pages from the disk to the main memory as needed in response to read user requests. Buffer manager is sometimes referred as the *cache manager.*

3. **DML processor:** Using a DML compiler, the DML processor converts the DML statements embedded in an application program into standard function calls in the host language. The DML compiler converts the DML statements written in a host programming language into object code for database access. The DML processor must interact with the query processor to generate the appropriate code.

4. **DDL processor:** Using a DDL compiler, the DDL processor converts the -

DDL statements into a set of tables containing metadata. These tables contain the metadata concerning the database and are in a form that can be used by other components of the DBMS. These tables are then stored in the system catalog while control information is stored in data file headers. The DDL compiler processes schema definitions, specified in the DDL and stores description of the schema (metadata) in the DBMS system catalog. The system catalog includes information such as the names of data files, data items, storage details of each data file, mapping information amongst schemas, and constraints.

2.6 DATA DICTIONARY

1. Data dictionary is also known as *Meta data.* A metadata is the data about the data. It is the self-describing nature of the database that provides program-data independence. It is also called as the *System Catalog.*

2. Data Dictionary is a repository-of information about a database that documents data elements of a database. It stores information about the database, attribute names and definitions for each table in the database

3. It holds the following information about each data element in the databases, it normally includes:

- Name
- Type
- Range of values
- Source
- Access authorization

4. Data dictionary is the integral part of the DBMS. Maintaining the data dictionary is the responsibility of DBA (Database Administrator).

5. The most general structure of data dictionary is shown in figure....

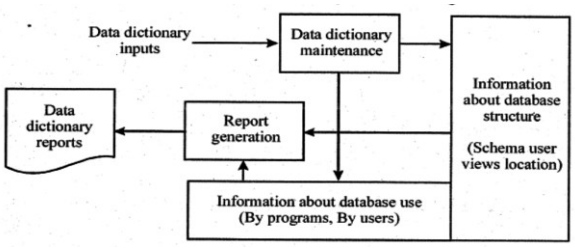


Fig. 2.10: Data Dictionary

6. Data dictionary is usually a part of the system catalog that is generated for each database. *A useful data dictionary system usually stores and manages the following types of information:*

- Descriptions of the schema of the database.
- Detailed information on physical database design, such as storage structures, access paths and file and record sizes.
- Description of the database users, their responsibilities and their access rights.
- High-level descriptions of the database transactions and applications and of the relationships of users to transactions.
- The relationship between database transactions and the data items referenced by them. This is useful in determining which transactions are affected when certain data definitions are changed.
- Usage statistics such as frequencies of queries and transactions and access counts to different portions of the database.
- Data dictionary provides the name of a data element, its description and data structure in which it may be found.
- Data dictionary provides great assistance in producing a report of where a data element is used in all programs that mention it.
- It is also possible to search for a data name, given keywords that describe the name. For example, one might want to determine the name of a variable that stands for net pay. Entering keywords would produce a list of possible identifiers and their definitions. Using keywords one can search the dictionary to locate the proper identifier to use in a program.

7. Data dictionary is used by developers to develop the programs, queries, controls-and procedures to manage and manipulate the data. It is available to database administrators (DBAs), designers and authorized user as on-line system documentation. This improves the control of database administrators (DBAs) over the information system

and the user's understanding and use of the system.

2.7 DBMS LANGUAGES

1. The main objective of a database management system is to allow its users to perform a number of operations on the database such as insert, delete, and retrieve data in abstract of data.
2. To provide the various facilities to different types of users, a DBMS normally provides one or more specialized programming languages called **Database** (or **DBMS) Languages.**
3. The DBMS mainly provides two database languages, namely, data definition language and data manipulation language to implement the databases.
4. Data definition language (DDL) is used for defining the database schema. The DBMS comprises DDL compiler that identifies and stores the schema description in the DBMS catalog.
5. Data manipulation language (DML) is used to manipulate the database.

The following are the DBMS languages:

1. **Data Definition Language:** DDL is used to specify the structure of table.

Sr.	Need And Usage	The SQL DDL Statement
1	Create schema objects	CREATE
2	Alter schema objects	ALTER
3	Delete schema objects	DROP
4	Rename schema objects	RENAME

2. **Data Manipulation Language:** The DBMS provides data manipulation language (DML) that enables users to retrieve and manipulate the data. The statement which is used to retrieve the information is called a query. The part of the DML used to retrieve the information is called a query language.

S No.	**Need And Usage**	**The SQL DDL Statement**

1	Remove rows from tables or views	DELETE
2	Add new rows of data into table or view	INSERT
3	Retrieve data from one or more tables	SELECT
4	Change columns values in existing rows of a table or view	UPDATE

3. **Data Control Language-(DCL):** DCL statements control access to data and the database using statements such as GRANT and REVOKE. A privilege can either be granted to a user with the help of GRANT statement. We can also revoke these statements by using REVOKES command.

S. NO.	**Need and Usage**	**The SQL DDL Statement**
1	Grant and take away privileges and roles	GRANT and REVOKE
2	Add a comment to the data dictionary	COMMENT

Questions

1. Discuss the concept of data independence and explain its importance in a database environment.
2. What is logical data independence and why is it important?
3. What is the difference between physical data independence and logical data independence?
4. Explain the difference between external, conceptual and internal schemas. How are these different schema layers related to the concepts of physical and logical data independence?
5. Describe the structure of a DBMS.
6. Describe the main components of a DBMS with a neat sketch, explain the structure of DBMS.
7. What do you mean by a data model? Describe the different types of data models

used.

8. Explain the following with their advantages and disadvantages:
   (a) Hierarchical database model
   (b) Network database model E-R data models
   (c) Relational database model
   (d) E-R data models
   (e) Object-oriented data model.
9. Define the following terms:
   (a) Data independence
   (b) Query processor
   (c) DDL processor
   (d) DML processor.
   (e) Run time database manager.
10. What is meant by the term client/server architecture and what are the advantages and disadvantages of this approach?
11. Compare and contrast the features of hierarchical, network and relational data models. What business needs led to the development of each of them?
12. Differentiate between schema, subschema and instances.
13. Explain the advantages and disadvantages of a centralised DBMS.
14. Explain the advantages and disadvantages of a parallel DBMS.
15. Explain the advantages and disadvantages of a distributed DBMS.
16. Explain data dictionary in detail.

3. INTRODUCTION

3.0 Evolution of Major Data Models

3.1 RDBMS

3.2 E.F. CODD'S RULES

3.3 COMPARISON BETWEEN DBMS AND RDBMS

3.4 DATA MODEL

3.5 CLASSIFICATION OF DATA MODEL

3.6 RECORD BASED MODELS

3.6.1 Hierarchical Model

3.6.2 Network Model

3.6.3 Relational Model

3.7 PHYSICAL MODEL

3.8 OBJECT BASED MODELS

3.8.1 E-R Model (Entity Relationship Model)

3.8.2 Object Oriented Model

3.8.3 Semantic Model

3.8.4 Functional Model

3.9 COMPARISON OF DATA MODELS

3.10 OTHER TERMS USED IN E-R MODEL

INTRODUCTION

A Data Model defines the logical design of the data. It describes the relationships between different parts of the data. Data model tells how the logical structure of a

database is modeled. Data Models are fundamental entities to introduce abstraction in DBMS. Data models define how data is connected to each other and how it will be processed and stored inside the system.

Evolution of Major Data Models

The historical literature reported drastic changes in the year 1970-1994. The Edgar F. Codd, in year 1970 disclose new concept of data representation. Mr. Codd suggested that all data in a database could be represented as a tabular structure (tables with columns and rows, which he called relations) and that these relations could be accessed using a high-level nonprocedural language. This research was result of several Relational DBMS like Oracle, Informix, Ingres and DB2. The following was high lights related to evolution database:

- **1980s:** The several vendors had developed OODBMSs like Object Design, Versant, O2 and Objectivity. The OODBMSs were no threat in the late 1980s to the now big commercial vendors developing and selling hierarchical, network or relational databases.
- **1990s:** In 1990s. The Object Database Management Group was founded, mainly &thanks to Rick Cattell of JavaSoft. The Green Team started the development of a new programming language which was loosely based on C++.

  The language was named Oak after the trees outside the office window of the language designer - James Gosling.
- **1993s to till date:** In 1993 several vendors of OODBMSs agreed upon an OODBMS standard called ODMG-93. The relational databases already had its standard-SQL-92, defined by its ANSI committee and ISO. The concept of internet, xml and other database management system was evolved in the time span. The detailed summery is represented in table below:

Generation	Time	Model	Examples	Comments
First	1960s-1970s	File . system	VMS/VSAM	Used mainly on IBM mainframe system Managed records, not relationships

Second	1970s	Hierarchical and network IDS --II	IMS ADABAS	Early database systems. Navigational access
Third	Mid-1970s to present	Relational	DB2 Oracle MS SQL-Server	Conceptual simplicity Entity relationship (ER) modeling support for relational data modeling
Fourth	Mid-1980s to present	Object oriented Extended Relational	Versant VFS/Fast Objects Objectivity/DB	Support complex data Extended relational products support objects and data warehousing Web databases become common
Next Generation	Present to future	XML	dbXML Tamino DB2 UDB Oracle 10gMS SQL Server	Organization and management of unstructured data Relational and object modelsaddsupport for XML documents

Table 3.1: A brief summary of how the major data models where developed

3.1 RDBMS

1. RDBMS stands for "Relational Database Management System[9]',
2. ***'"RDBMS is a DBMS in which data is stored in the formthe form of tables and the relationship among the data is also stored in the form of tables."***
3. RDBMS also provide relational operators to manipulate the data stored into the database tables.
4. It is based on the relational model and was introduced by E.F. Codd.
5. E.F. Codd, the famous mathematician has introduced 12 rules (known as Codd's rules) to assist a database product to qualify a RDBMS. - .
6. RDBMS product has to satisfy at least 6 of the 12 rules, of Codd to be accepted asa full-fledged RDBMS.
7. Examples of RDBMS are: Oracle, Sybase, SQL- Server.
8. In short, all the information in RDBMS should be presented in tabular form and it follows Codd's rules.

3.2 E.F. CODD'S RULES

1. E.R Codd the famous mathematician has introduced 12 rules for the relational model for databases commonly known as Codd's rules.
2. These rules define what is required for a DBMS to be considered RDBMS.

3. The Codd's rules are as follows:

(a) **Information Rule:** Every information in RDBMS is represented in the form of tables.

(b) **Guaranteed Access Rule:** Every information in RDBMS is accessed by using combination of table name and primary key. A primary key helps to identify a rowname and column name.

(c) **Systematic Treatment of Null Values:** RDBMS supports null values for representing missing or Inapplicable information.

(d) **The Description Rule:** The database description is represented at the logical level in the same way as ordinary data. The authorized users can apply the same relational language for its manipulation as they apply to the regular data.

(e) **The Comprehensive Data Sublanguage Rule:** RDBMS supports many languages which allow users to define tables, query and update the data and set integrity constraints.

(f) **The View Updating Rule:** All the views that are theoretically updatable must be updatable by the system.

(g) **High Level Insert, Update and Delete:** The system must support insert, update and delete operations.

(h) **Physical Data Independence:** It is the ability to change the internal schema (Physical Level) without affecting the conceptual/logical or external schema.

(i) **Logical Data Independence:** Logical data independence is more difficult to achieve than the physical data independence. It is the ability to change the conceptual/logical schema without affecting the external schema.

(j) **Integrity Independence:** Integrity constraints should be specified separately from application programs and stored in the catalog. Integrity constraints canbe changed without affecting the application programs.

(k) **Distribution Independence:** User should not have to be aware of whether a database is distributed at different sites or not.

(l) **The Non-Subversion Rule:** If the RDBMS has a language that accesses the information of a record at a time, this language should not be used to bypass the integrity constraints. .

3.3 COMPARISON BETWEEN DBMS AND RDBMS

DBMS	RDBMS
1. It stands for "Database Management System."	It stands for "Relational Database Management System."
2. It can store data in any format (graph, table, tree etc.)	It can store data only in tabular form.
3. It does not support client/server architecture.	It supports client/server architecture.
4. It does not satisfy Codd's rules.	It satisfyCodd's rules.
5. It requires low software and hardware requirements.	It requires high software and hardware requirements.
6. It can maintain only single user at a time. It supports single user.	It can maintain many users at a time. It supports multi-user.
7. It is designed for small organizations with small amount of data, where security of data is not a major issue.	It is designed for large organization with large amount of data where security of data is a major issue.
8. It does not support referential constraints.	It supports referential integrity constraints.
9. Examples of DBMS: Dbase, Foxpro	Examples of RDBMS: Oracle, Sybase, SQL-Server.

3.4 DATA MODEL

1. A model is a representation of reality, 'real world' objects and events,and their association.
2. A data model represents the organization itself.
3. Data model can be defined as an integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.
4. The purpose of a data model is to represent data and to make the data understandable.

Objectives of Data Model

- The main objective of database system is to highlight only the essential features and to hide the storage and data organization details from the user.
- A database model provides the necessary means to achieve data abstraction.
- A data, model is an abstract model that describes how the data is represented and used.
- A data model consists of a set of data structures and conceptual tools that is used to describe the structure (data types, relationships, and constraints) of a database.
- A data model not only describes the structure of the data, it also defines a set of operations that can be performed on the data.
- A data model generally consists of data model theory, which is a formal description of how data may be structured and used.
- The process of applying a data model theory to create a data model instance is known as data modeling.

3.5 CLASSIFICATION OF DATA MODEL

Data Model is a collection of concepts to provide abstraction into DBMS so that superfluous details can be hide while highlighting important detail of data entities. It defines the logical design of data and establishes relationship between them. Data representation provides mechanisms to structure data for entities being modeled and

allow a set of operations to be performed on them. A number of Models has been developed which are further categorized as below

1. Object based Logical Model
2. Record based Logical Model
3. Physical Data Model

Depending on the concept they use to model the structure of the database, the data models are categorized. The following logical tree display detailed classification:

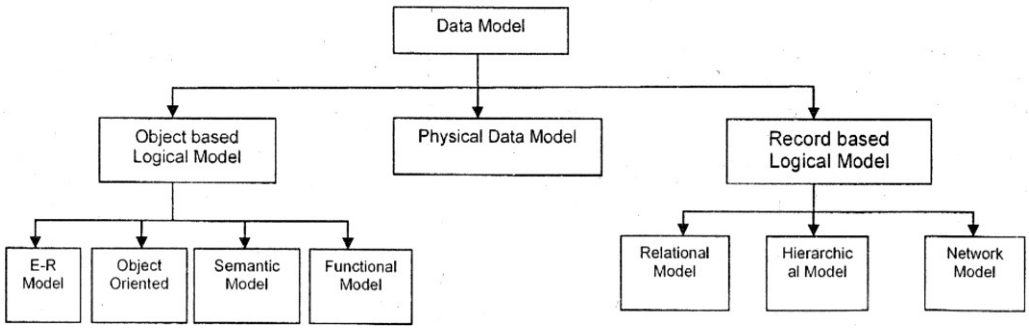


Fig. 3.1: Classification of Data Model

3.6 RECORD BASED MODELS

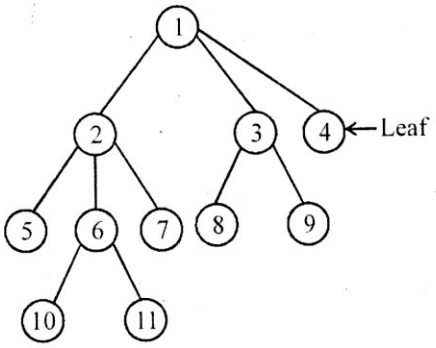
A record-based data models are used to specify the overall logical structures of the database. This model is used describing data at logical and view level. In the record based models, the database consists of a number of fixed-format records possibly of different types. Each record type defines a fixed number of fields, each typically of a fixed length. Data integrity constraints cannot be explicitly specified using record-based data models. There are three principle types of record-based data models:

3.6.1 Hierarchical Model

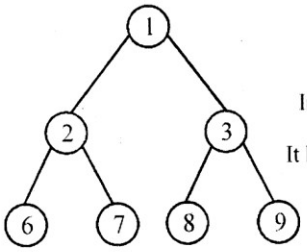
1. It was developed jointly by North American Rokwell Company and IBM.
2. It is the oldest model.
3. It follows tree as its basic structure.
4. Node at highest level is called Root.
5. A node may have any no. of children but each child node has only one parent.
6. Children of same parents are called siblings.

7. Anode that has no child is leaf node.

8. Representation of hierarchical model with suitable diagram

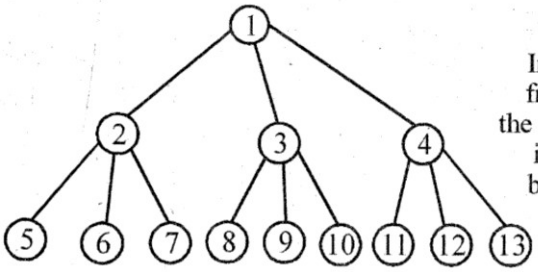


Leaf node is that node which has further no child. Here 5,10,11,7,8,9 & 4 are leaf nodes.



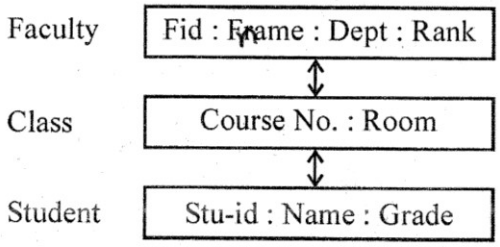
In binary tree, each node has not more than two child. It has exactly 2 child or 0 child.

Binary Tree

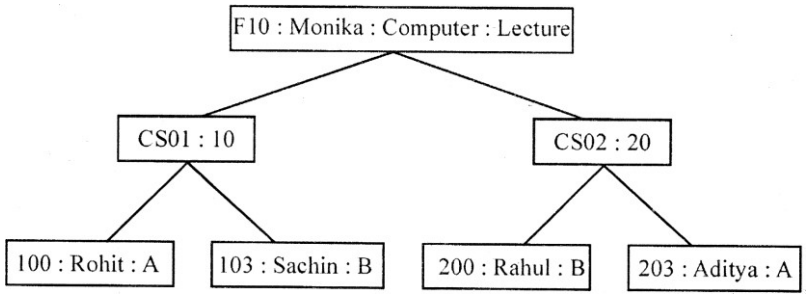


In balanced tree, every part from root node to a leaf has the same length. A balanced tree is always a binary tree but binary tree may or may not be a balanced tree.

Balanced Tree



- Faculty is on root node containing four attributes (Fid, Fname, Dept, Rank)
- Class is child of faculty node contains (Course no., room) attributes.
- There is one to many relationship between each faculty record and its class record.
- There is one to many relationship between each class record and its student record.



Operations on Hierarchical Model

(a) Insertion: A new class says CSO3 cannot be inserted unless some faculty is available at root level because without parent we can't insert any child node. This operation is used to insert a new record into the database. There are two possibilities:

(i) If the inserted, record is a root record then it creates new tree with the new record as the root

(ii) If the inserted record is a child record, then we need to determine its parent first because no child record can exist without a parent record. So, insertion problem exists for the children who have no parents.

(b) Deletion: If we want to remove the class 100 then student Rohit will also have to be removed. This operation is used to delete a record from the database. To delete a record, we must first make it the current record of the database and then delete it. Here also, there are two possibilities:

(i) If the deleted record has no child node. It can be deleted easily.

(ii) If the deleted record has one or more child nodes, then the deletion process will deleteall the child nodes also. This may lead to loss of Information also.

(c) Update: If we want to updated the room 10, then we have to find all the records related to room 10 and have to modify. This operation is used to update a record. There are two possibilities:

(i) If the record to be updated is a parent record, then updating it requires only one updation operation to be performed because there is only one occurrence of a parent record.

(ii) If the record to be updated is a child record, multiple updations may be required. If it not happens, this may lead to inconsistency in the database.

(d) Record Retrieval: Record retrieval methods for hierarchical model are complex and asymmetric. Retrieval means first searching the required record and then fetching it. Retrieval involves pointers from the parent node to the-child node in the tree and hence is complex and time consuming.

Advantages of Hierarchical Model

1. **Simplicity:** The relationship between the various layers is logically simple. Thus the designof a hierarchical database is simple.
2. **Data Security:** Hierarchical model was the first database model that offered the data security that is provided and enforced by the DBMS.
3. **Data Integrity:** Hierarchical model is based on the parent/child there is always a link between the parent segment and the child segments under it. The child segments are always automatically referred by its parents, so this model promotes data integrity.
4. **Efficiency:** The hierarchical database model is a very efficient one when the users require large number of transactions, using data whose relationships are fixed.

Disadvantages of Hierarchical Model

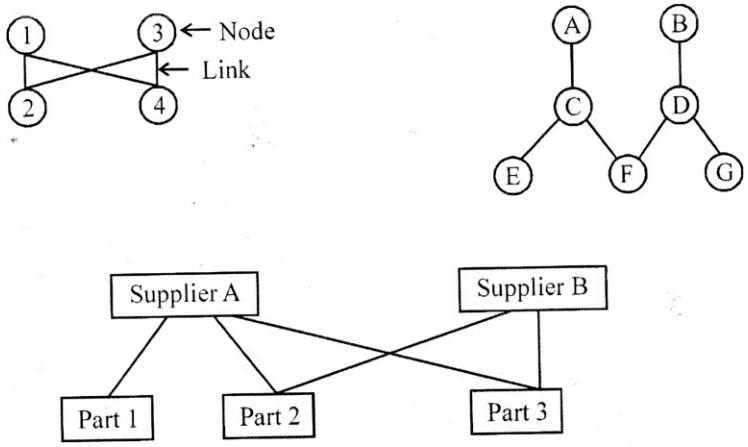
1. **Implementation Complexity:** Although the hierarchical database model is conceptually simple and easy to design, it is quite complex to implement, the database designers should have very good knowledge of the physical data storage characteristics.
2. **Lack of Structural Independence:** Structural independence exists when the changes to the database structure does not affect the DBMS's ability to access data. Thus in a hierarchical database the benefits of data independence is limited by structural dependence.
3. **Programs Complexity:** Due to the structural dependence and the navigational structural, the application programs and the end users must know precisely how the data is distributed physically in the database in order to access data. This

requires knowledge of complex painter systems, which is often beyond the grasp of ordinary users.

4. **Operational Limitations:** Hierarchical model suffers from the Insert anomalies, anomalies and deletion anomalies, also the retrieval operation is complex and asymmetric, thus hierarchical model is not suitable for all the cases.
5. **Implementation Limitations:** Many of the common relationships do not confirm to the 1'N format required by the hierarchical model. The many-to-many (N;N) relationships, which are more common in real life are very difficult to implement in a hierarchical model.

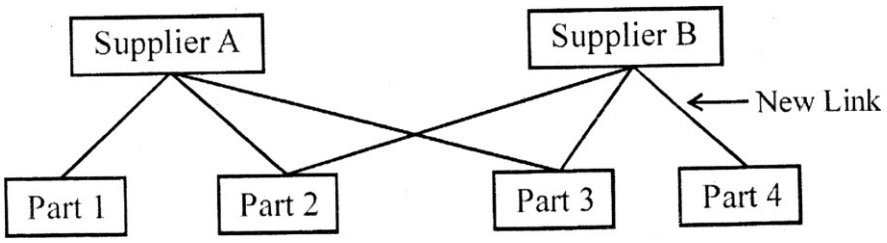
3.6.2 Network Model

1. Data in this model is represented by Links.
2. It looks like a tree structure containing nodes.
3. Every node may have one or more than one parent node.
4. Dependent node is called child node.

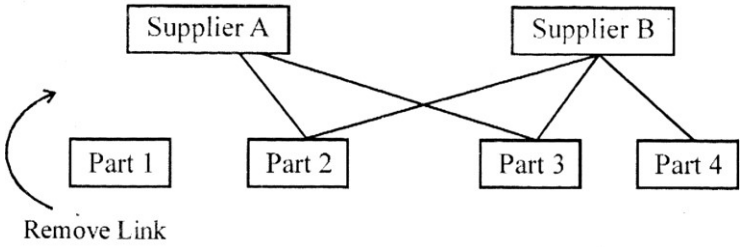


Operations on Network Model

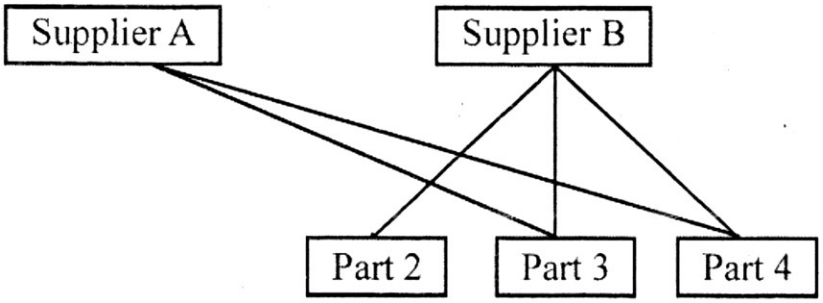
(a) **Insert Operation:** Insertion is easy i.e. supplier B supplies new part then we have to create a new link only. No other updation is required.



(b) **Delete Operation**: If we want to delete the information of any part, say supplier A doesn't want to supply part 1 now, so we have to remove only the link.



(c) **Updation Operation:** Updation is also easy. Suppose supplier A doesn't supply part 2 it supplies part 4 now.



(d) **Retrieval Operation:** Record retrieval method for network model are symmetric but complex.

Advantages of Network Model

1. **Conceptual Simplicity:** Like hierarchical model; the network model is also conceptually simple and easy to design.
2. **Capability to handle mass relationship types:** The network model can handle the one-to-many (1;N) and many to many (N:N) relationships, which is a real help in modeling the real life situations.

3. **Ease of data access:** The data access is easier than and flexible than the hierarchical model.
4. **Data Integrity:** The network model does not allow a member to exist without an owner. Thus a user must first define the owner record and then the member record. This ensures the data integrity.
5. **Data Independence:** The network model is better than the hierarchical model in isolating the programs from the complex physical storage details.
6. **Database Standard:** One of the major drawbacks of the hierarchical model was the non availability of universal standards for database design and modeling.

All the network database management systems conformed to these standards. These standards included a Data Definition Language [DDL] and the Data Manipulation Language [DML]$_9$ thus quality enhancing database administration and portability.

Disadvantages of Network Model

The network database model was significantly better than the hierarchical database model, it also had many drawbacks. These are

1. **System Complexity:** All the records are maintained using pointers and hence the whole database structure becomes very complex.
2. **Operational Anomalies:** Network model's insertion, deletion and updating operations of any record require large number of pointer adjustments, which makes its implementation very complex and complicated.
3. **Absence of Structural Independence:** If changes are mode to the database structure then all the application programs need to be modified before they can access data. Thus, every though the network database model succeeds in achieving data independence, it still fails to achieve structural independence.

Note: *We can conclude that network model does not suffer from the Insert anomalies, Update anomalies and Deletion anomalies. The retrieval operation is symmetric, as compared to hierarchical model, but the main disadvantage is complexity of the model.*

Hierarchical	Network

Each child node have only one parent.	Each child node may have more than one parent.
Hierarchical model records are organized as collection of trees.	Network model they are represented as arbitrary graphs.

3.6.3 Relational Model

1. It is primary data model for commercial data processing. The relational model wasproposed by E.F.Codd of the IBM in 1972.
2. Relational model is a collection of tables. Tables are also known as relations.Therefore it is known as relational model.
3. Relational model represents the database as a collection of relations. Each relation (table)is a collection of row and columns.
4. Each table has a unique, name in database.
5. Columns are called attributes and rows are called tuples.
6. For each attribute there is a set of permitted values called domain.
7. Attribute name will be unique in a table.
8. Domain value can be NULL which shows that the value is unknown or does not exist.
9. The order of attribute has no significance. We can arrange attributes in any order.
10. We can insert record in any order.
11. Representation of data in Relational Model: A relational database consists of any number of relations. We can represent relation schemes by giving the name of the relation, followed by the attribute names in parenthesis.

Note: We will study Relational Model in detail in Unit-4

3.7 PHYSICAL MODEL

Physical model describes the in terms of a collection of files, indices, and other storage structures such as record formats, record ordering, and access paths. This model specifies how the database will be executed in a particular DBMS software such as Oracle, Sybase, etc., by taking into account the facilities and constraints of a given database management system. It also describes how the is stored on disk. Physical models are used for a higher-level description of storage structure and access mechanism.

They describe how data is stored in thecomputer, representinginformation such as record structures, record orderings and access paths. It is possible to implement the database at system level using physical data models. There are not as many physical data models so far. The most common physical data models are as follows:

- Unifying model
- Frame memory model.

3.8 OBJECT BASED MODELS

The object based models use the concepts of entities or objects and relationships among them. An entity is a distinct object (a person, place, concept, and event) in the organization that is to be represented In the database. An attribute is a property that describes some aspect of the object that we wish to recordand a relationship is an association between 'entitles. It provides flexible structuring capabilities and allows data constraints to be specified explicitly. The object based logical model are classified as follows:

1. E-RModel
2. Object Oriented Model
3. Semantic Model
4. Functional Model

3.8.1 E-R Model (Entity Relationship Model)

E-R model is an effective and standard method of communication amongst different designers, programmers and end-users who tend to view data and its use in different ways. It is a non- technical method, which is free from ambiguities and provides a standard and a logical way of visualizing the data. It gives precise understanding of the nature of the data and how it is used by the enterprise. It provides useful concepts that allow the database designers to move from an informal description of what users want from their database, to a more detailed and precise description that can be implemented in a database management system. Thus, E-R modeling is an important technique for any database designer to master. It has found wide acceptance in database design. A basic

concept of E-R model has been introduced and few examples of E-R diagram of an enterprise database have been illustrated. The ER model is based on a concept of a real world entities and relationships among these entities. It can be used developed database design by allowing specification schema, which represents the overall logical structure of a database. It is very useful in mapping the meanings and interactions of real-world entities onto a conceptual schema.

1. E-R model is based on real world. It is a collection of basic objects, called entities and of relationships among these objects (Entity).
2. E-R model employs three basic features:
   - Entity
   - Attributes
   - Relationship
3. The overall logical structure of a database can be expressed graphically by an E-R diagram, which is built up by the following components:
   - Rectangle, which represent entity sets.
   - Ellipses, which represent attributes.
   - Diamonds, which represent relationships among entity sets.
   - Lines, which link attributes to entity sets and entity sets to relationships.

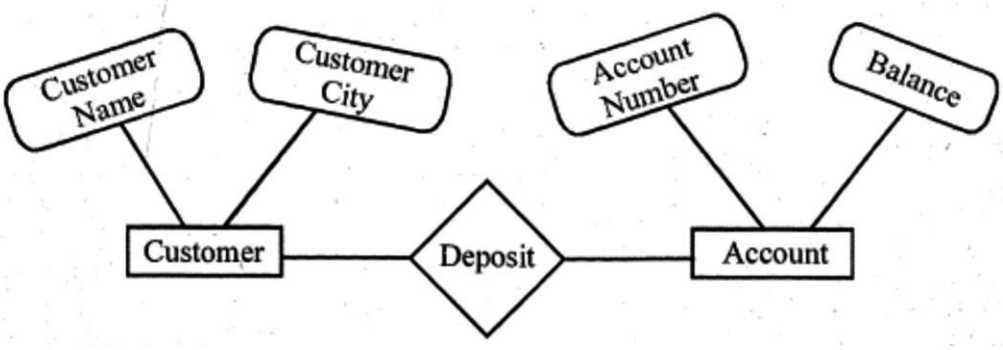


Fig. 3.2: E-R Model

Entity: Customer and Account are entity.

Attribute: Customer name, customer city are attributes of customer entity. Account number, balance are attributes of Account Entity.

Relation: Deposit is relationship among customer and Account.

I. ENTITY

- Entity is a thing which can be identified.
- Entity is a person, place thing event or concept which can be identified. We can say about which we want to store information i.e. employee, student, customer.
- Rectangle sign is used to represent the entity in E-R diagram.
- Entity can be of two types which are as follows:

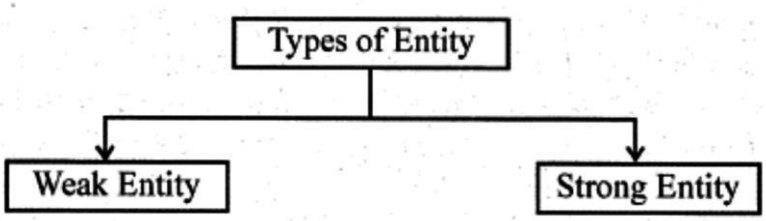
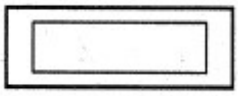
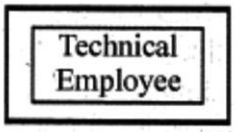


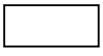
Fig. 3.3: Types of Entity

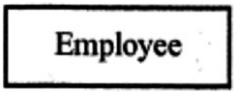
(a) Weak Entity:

- Weak entity depends on some other entity.
- It can't exist if other entity on which is depends does not exist.
- It is represented by 
- For example:



(b) Regular Entity/Strong Entity:

- Regular entity does not dependent on other entity.
- Its existence doesn't depend upon any other entity.
- It is represented by 
- For example:



(c) Entity Set: It is a set of entities of the same type that share the same properties (attributes) i.e. the set of all persons who are customer at same bank.

(d) Entity Subtype and Super type: Entity can be sub or super i.e. we have an Entity employee and employee can be programmer or operation.

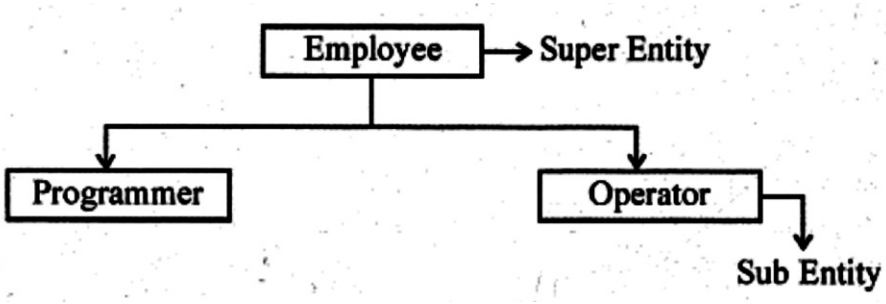
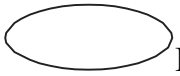


Fig. 3.4. Super Entity and Sub Entity

II. ATTRIBUTE

- Attributes are properties of entity. They are also called columns.
- Entity is about which we want to store information and attribute is what information we want to store.
- For example: If we want to store Name, City, and Salary information about employee. Then employee is our entity and name, city, salary are attributes.
-  Ellipse sign is used to represent attributes.
- Attribute can be of four types which are as follows:

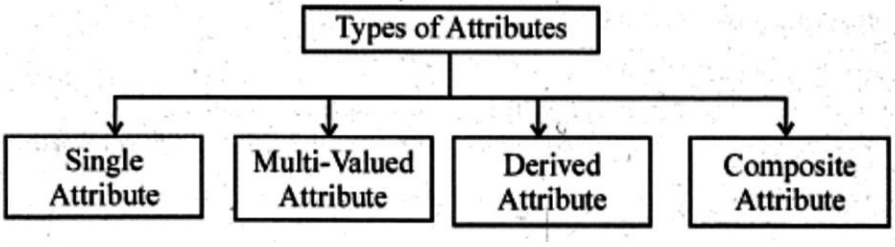
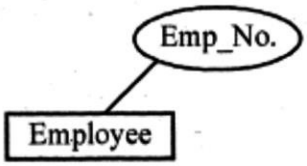
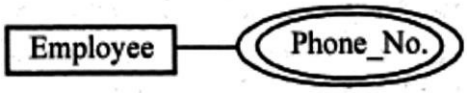


Fig. 3.5 :Types of Attribues

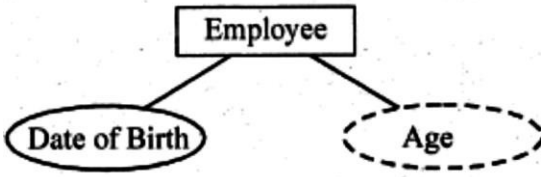
(a) **Single Attribute:** Single attributes are those attributes which can't be divided into sub parts i.e. Employee number is a simple attribute.



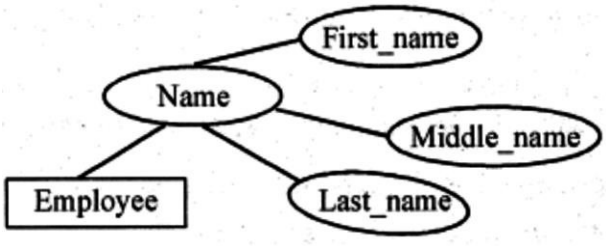
(b) **Multi Valued Attribute:** Multi valued attributes are those attribute which have more than one value i.e. Phone number is a multi-valued attribute. One employee may have more than one phone number.



(c) **Derived Attribute:** Derived attributes are those attributes whose value is derived from another attribute. For example: value of age attribute can be drive (calculate) from date of birth attribute and current date.



(d) **Composite Attribute:** Composite attributes are those attributes which can be divided into parts. For example: Name attribute can be divided into First Name, Middle Name and Last Name. Address attribute can be divided into street, city, state, zip-code.



III. RELATIONSHIP

- It is used to connect the entities.

- The entities involved in given relationship are called participants.
- The no. of participants in a given relationship is called degree of *relationship*.
-  sign is used to represent relationship among entities.
- Deposit is a relationship among entity customer and entity account.
- Relationship can be of four types which are as follows:

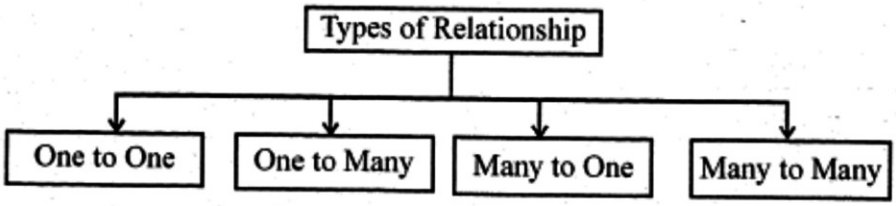
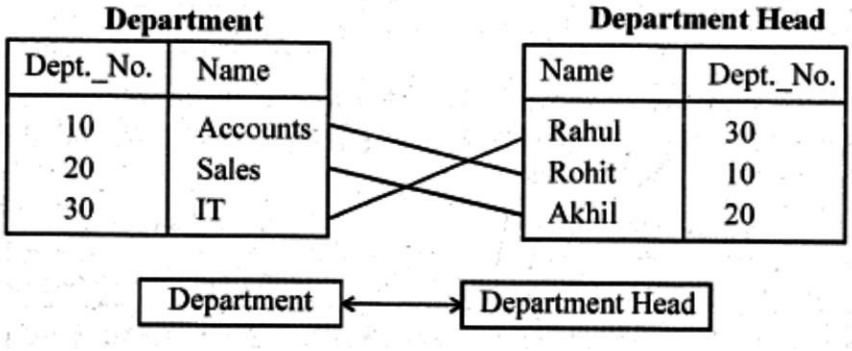
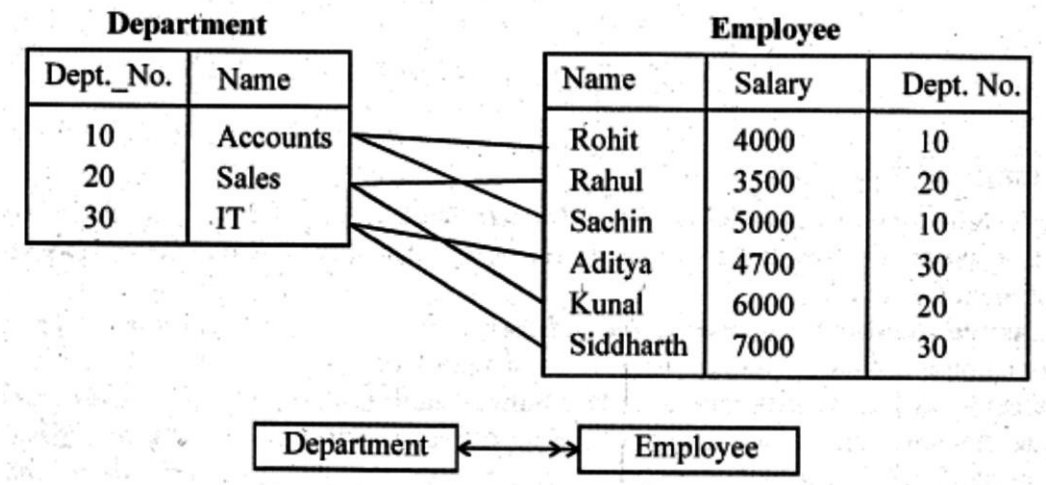


Fig. 4.5: Types of Relationship

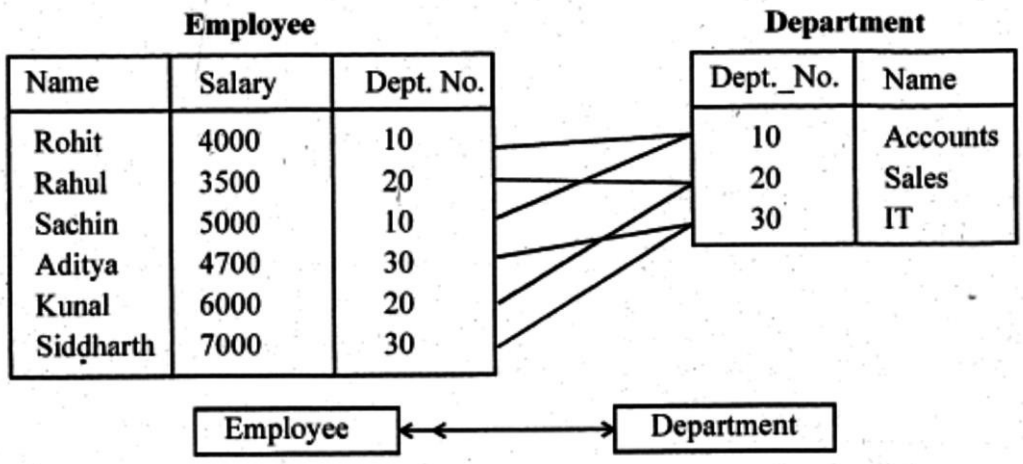
(a) One to One Relationship: In one to one relationship for one record in entity A, there is exactly one record in entity B. For example: we have two entities department and department head. There is one to one relationship because one department will be under one head and one head will be appointed for one department.



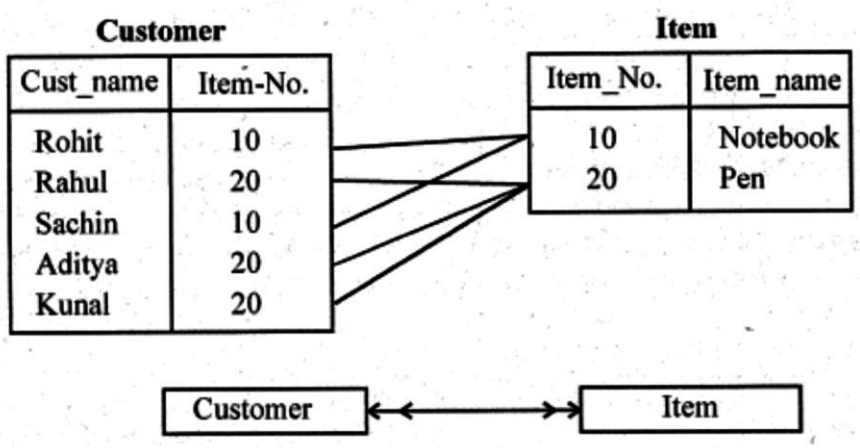
(b) One to Many Relationships: In one to many relationships for one record in entity A, there is more than one record in entity B. For example: We have two entities department and employee. There is one to many relationships because there will be one department in a company and more than one employee will work in that particular department.



(c) Many to One Relationship: In many to one relationship, for many records in entity A, there is only one record in entity B. For example: We have two entities employee and department. There is many to one relationship because there will be many employees in a single



(d) Many to Many Relationships: In many to many relationships, for many record is an entity A, there will be many record in entity B. There is many to many relationship because there will be many customers for many items.



Advantages of E-R Model

1. **Straight forward relation representation:** Having designed an E-R diagram for a database application, the relational representation of database model becomes relatively straight forward.
2. **Easy conversion for E-R over Data Model:** Conversion from E-R diagram to network or hierarchical data model can easily be accomplished.
3. **Graphical Representation for better understanding:** An E-R model gives graphical and diagrammatical representation of various entities, its attributes and relationship between entities. This helps in understanding the data structure in easy way, minimize the redundancy and other problem.

Disadvantages of E-R Model

1. Popular for high-level design: It is especially popular for high level design.
2. No Industry standard of Notation.

Difference between Strong Entity and Weak Entity

Sr. No.	Strong Entity	Weak Entity
1	It has an attribute having the capability that can act as primary key.	It does not have attribute which may act as a primary key.
2		

3.8.2 Object Oriented Model

The object oriented data model is an. adaptation of the object oriented programming language paradigm to database systems. The model is based on the concept of encapsulating data and code that operates on that data in an object. On the other hand, the object-relational data model is an., extension of relational data model. It combines the features of both the relational data model and object-oriented data model.

Object oriented data models for databases "extend the above mentioned data modeling features of the object oriented paradigm The extensions include data integrity constraints, persistence of data which allows transient data to be distinguished from persistent data and support for collections.

Advantages of Object-Oriented Data Model

1. **Capable of handling a large variety of data types:** hierarchical, network or relational), the object-oriented database are capable of storing different types of data, for example, pictures, voices, video, including text, numbers and soon.
2. **Combining object-oriented programming with database technology:** Object-oriented data model is capable of combining object-oriented programming with database technology and thus, providing an Integrated application development system.
3. **Improved productivity:** Object-oriented data models provide powerful features such as inheritance, polymorphism and dynamic binding that allow the users to compose objects and provide solutions without writing object-specific code. These features increase-the productivity of the database application developers significantly.
4. **Improved data access:** Object-oriented data model represents relationships explicitly, supporting both navigational and associative access to information. It further improves the data access performance over relational-value-based relationships.

Disadvantages of Object-Oriented Data Model

1. **No precise definition:** It is difficult to provide a precise definition of what constitutes an object-oriented DBMS because the name has been applied to a variety of products and prototypes, some of which differ considerably from one another.
2. **Difficult to maintain:** The definition of objects is required to be changed periodically and migration of existing databases to confirm to the new object definition with change in organisational information needs. It possess real challenge when changing object definitions and migrating databases.
3. **Not suited for all applications:** Object-oriented data models are used where there is a need to manage complex relationships among data objects. They are especially suited for specific applications such as engineering, e-commerce, medicines and so on, and not for all applications. Its performance degrades and requires high processing requirements when used for ordinary applications.

3.8.3 Semantic Model

This model is used to express greater interdependencies among entities of interest. These independencies enable the models to represent the semantics of the data in the database. The Semantic Data Model (SDM), like other data models, is a way of structuring data to represent it in a logical way. SDM differs from other data models in that it focuses on providing more meaning of the data itself, rather than only on the relationships and attributes of the data.

SDM provides a high-level understanding of the data by abstracting it further away from the physical aspects of data storage.

In SDM, an entity represents some aspect or item in the real world, such as a student An entity is similar to a record in a relational system or an object in an object-oriented system. These entities in SDM focus on types, which are more general, instead of sets of data. In SDM, an entity is a very basic notion of a real-world or conceptual object that is defined by a single attribute.

For instance, an SDM entity type might be *person* which would be a list of names of people that are to be represented by the data. The objects in this domain would then point to specific instances of a person that are represented by each person entity.

3.8.4 Functional Model

The functional data model describes those aspects of a system concerned with transformations of values-functions, mappings, constraints and functional dependencies. The functional data model describes the computations within a system.

- It shows how output value is derived from input values without regard for the order which the values are computed. It also includes constraints among values.
- It consists of multiple data flow diagrams.
- Data flow diagrams show the dependencies between values and computation of output values from input values and functions, without regard for when the functions are executed.
- Traditional computation concepts such as expression trees are examples of functional models.

3.9 COMPARISON OF DATA MODELS

Sr. No.	Hierarchical Model	Network Data Model	Relational Data Model
1.	Hierarchical data model represents data in a tree format where Parent and Child relationship is represented to show association.	Network model represents data in graphs where data is a record which is linked by pointers.	Relational data model logically represents data in Tabular form where data is placed in row and column.
2.	Many to many relationship cannot be expressed in hierarchical model	Many to many relationship can be expressed in hierarchical model.	Many to many relationship can be expressed in hierarchical model.
3.	It is good for expressing data in < parent child relationship	It is good for modelling of many to many relationship.	It is good for modelling real world entities.

4	Relationship are represented by pointer and relationship among records are physical in nature	Network model also represents relationship through pointers and nature of the relationship is physical.	Relational model is stored data in form of rows and column. There is no physical connection is established between different tables whereas connection is logical in nature and established through keys.
5	Searching of a particular record is a time consuming task as to reach a particular child we have to process through its parent record.	Searching of a particular record is easy since there are multiple access path available to reach a node in graph.	In case of relations tables we use concept of keys to identify the records and search a key through indexing is quite simple task
6.	Insertion is done in the form of parent node and child node relationship. We cannot insert child node in tree without parent node.	Network model insertion can be performed by inserting new node in the graph with ease and has no insertion anomaly	in Relation model , new record can be added any time and has no insertion anomaly
7.	Updation operation may results in inconsistency as there are multiple child records in a tree	Updation operation is free from any anomaly as there is only single occurrence of each record in a graph which may be connected with multiple records.	Updation operation is safe in a relational model as duplication of record is avoidable by applying normalisation and Primary keys relationships
8.	Hierarchical model is based on parent child relationship and deleting of child is easy as compare to parent, if we delete parent then child node will automatically deleted from the tree.	There is no deletion anomaly as deleting of one node does not affect other nodes due to many to many relationships.	The deleting of record from a relation is again a simple process and there is no anomaly related to deleting of records. Deletion of reference records is not allowed as it may linked to other records

3.10. OTHER TERMS USED IN E-R MODEL

CONSTRAINTS

Relationship types usually have certain constraints that limit the possible combinations of entities that may participate in the corresponding relationship set. The constraints should reflect the restrictions on the relationships as perceived in the 'real

world'. For example, there could be a requirement that each department in the entity DEPT must have a person and each person in the PERSON entity must have a skill. The main types of constraints on relationships are multiplicity, cardinality, participation and so on.

1. **Multiplicity Constraints:** Multiplicity is the number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship. It constrains the way that entities are related. It is a representation of the policies and business rules established by the enterprise or the user. It is important that all appropriate enterprise constraints are identified and represented while modeling an enterprise.

2. **Cardinality Constraints:** A cardinality constraint specifies the number of instances of one entity that can (or must) be associated with each instance of entity. There are two types of cardinality constraints namely minimum and maximum cardinality constraints. The minimum cardinality constraint of a relationship is the minimum number of instances of an entity that may be associated with each instance of another entity. The maximum cardinality constraint of a relationship is the maximum number of instances of one entity that may be associated with a single occurrence of another entity.

3. **Participation Constraints:** The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type. There are two types of participation constraints namely total and partial participation constraints. Total participation constraints means that every entity in 'the total set' of an entity must be related to another entity via a relationship. Total participation is also called existence dependency. A partial participation constraint means that some or the 'part of the set of an entity are related to another entity via a relationship, but not necessarily all. The cardinality ratio and participation constraints are together known as the structural constraints of a relationship type.

4. **Exclusion and Uniqueness Constraints:** E-R modeling has also constraints such exclusion constraint and uniqueness constraint that results into poor semantic base and tries to make entity-attribute decisions early in the conceptual

modeling process. In exclusion constraint the normal or default treatment of multiple relationships is inclusive OR, which allows any or all of the entities to participate. In some situations, however, multiple relationships may be affected by the exclusive (disjoint or exclusive OR) constraint, which allows at most one entity instance among several entity types to participate in the relationship with a single root entity.

GENERALIZATION

1. A generalization hierarchy is a form of abstraction that specify the two or more entities that share the common attributes can be generalized into a higher-level entity type called a super type or generic entity.
2. The lower level of entities becomes the subtype. Subtypes are dependent entities.

SPECIALIZATION

1. Specialization is a process of taking subsets of higher level entity set to form lower level entity sets.
2. It is a process of defining a set of subclasses of an entity type which is called as super class of the specialization.

AGGREGATION

1. One limitation of the E-R model is that it cannot express relationship among relationships.
2. Aggregation is the process of compiling information on an object, thereby abstracting a higher level object.
3. Aggregation allows us to indicate that a relationship set participate in another relationship set.

Questions

1. What do you mean[1] by data models? Explain the answer.
2. How can we classify data models?
3. What do mean by relationships in a data model
4. What is an attribute in data modeling?

5. Explain the Relational Model? Write advantages and disadvantages.
6. Explain the Hierarchical Model? Write advantages and disadvantages.
7. Explain different operations that can be performed on Hierarchical
8. Compare different data models.
9 Define the following terms:
   (a) Entity Set
   (b) Attribute
   (c) Relationship Set
   (d) Simple attributes
   (e) Composite attributes
   (f) Multivalve attributes.
10 What are the different types of attributes? Explain using examples.
11 What are mapping constraints? What are its types?
12 What are weak entity sets? Why are they used?
13 What is generalization? Explain with a suitable example.
14 What is aggregation? Explain using a suitable example.
15 The E-R Diagram for an Employee Payroll System.
16 Discuss the advantages and disadvantages of ER model.
17 The E-R Diagram for Book Purchasing System..
18 Explain with diagrammatical illustrations about the different types of relationships

SECTION-B

UNIT 4: RELATIONAL DATA MODEL

4.1 RELATIONAL MODEL

4.2 COMPARISON OF DATA MODELS

4.3 RELATIONAL ALGEBRA AND RELATIONAL CALCULUS

4.4 RELATIONAL ALGEBRA

4.5 RELATIONAL CALCULUS

4.6 DIFFERENCE BETWEEN RELATIONAL ALGEBRA AND RELATIONAL CALCULUS

4.1 RELATIONAL MODEL

1. It is primary data model for commercial data processing. The relational model wasproposed by E.F.Codd of the IBM in 1972.
2. Relational model is a collection of tables. Tables are also known as relations. Therefore it is known as relational model.
3. Relational model represents the database as a collection of relations. Each relation (table) is a collection of row and columns.
4. Each table has a unique, name in database.
5. Columns are called attributes and rows are called tuples.
6. For each attribute there is a set of permitted values called domain.
7. Attribute name will be unique in a table.
8. Domain value can be NULL which shows that the value is unknown or does not exist.
9. The order of attribute has no significance. We can arrange attributes in any order.
10. We can insert record in any order.
11. Representation of data in Relational Model: A relational database consists of any number of relations. We can represent relation schemes by giving the name of the relation, followed by the attribute names in parenthesis.

12. Components of Relational Model

(1) Data Structure

(2) Data Integrity

(3) Integrity Constraints

1. Data Structure

(a) Relation

- All data is represented in table.
- Table contains rows and columns. Table isalso known as Relation.
- Columns are called attributed and rows are called tuples. Each cell of relation contains only single value.
- Relation contains information on one subject only.

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10^{th}	Chennai	1010
2.	Akhil	12^{th}	Mumbai	1229
3.	Aditya	8^{th}	Bangalore	801
4.	Sachin	9^{th}	Gurgaon	906
5.	Rahul	11^{th}	Patiala	1112
6.	Sid	12^{th}	Pune	1225
7.	Kunal	8^{th}	Hyderabad	806

Student is the title of the relation. There are 5 column (S.No, Name, Class, City, Roll Number) and 7 rows.

(b) Attribute

- Columns-are called attributes.
- Attributes appear vertically in a relation.
- Attributes can appear In any order and provide specific information.
- Attributes.in a relation "Student" are S.No., Name, Class, City, Roll Number.

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10^{th}	Chennai	1010
2.	Akhil	12^{th}	Mumbai	1229
3.	Aditya	8^{th}	Bangalore	801
4.	Sachin	9^{th}	Gurgaon	906
5.	Rahul	11^{th}	Patiala	1112
6.	Sid	12^{th}	Pune	1225
7.	Kunal	8^{th}	Hyderabad	806

(c) Tuple

- Rows are called tuples.
- Tuples appear horizontally in a relation.
- Tuples can appear in any order and provide complete information (full record).
- There are 7 tuples in a relation "Student".

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10^{th}	Chennai	1010
2.	Akhil	12^{th}	Mumbai	1229
3.	Aditya	8^{th}	Bangalore	801
4.	Sachin	9^{th}	Gurgaon	906
5.	Rahul	11^{th}	Patiala	1112
6.	Sid	12^{th}	Pune	1225
7.	Kunal	8^{th}	Hyderabad	806

(d) Domain

- Domain is a set of all allowed or possible values for an attribute in a relation.

- Domain specifies the type of data used in an attribute.
- For example, in a relation student, students are from different cities. In this case, city is an attribute of relation student and Mumbai, Patiala, Gurgaon etc. are domain values of city attribute.

(e) Degree

- Degree is the number of attributes in a relation.
- If a relation have only one attribute, then its degree is one and known as called unary relation
- If a relation have two attributes, then its degree is two and known as called binary relation
- If a relation have three attributes, then its degree is three and known as called ternary and so on.
- Degree of relation "Student" is 5.

(f) Cardinality

- Cardinality is the number of tuples in a relation.
- Cardinality changes on the basis of insertions or deletions of records in a relation.
- Cardinality of relation "Student" is 7

2. **Data Integrity:** Data integrity ensures the accuracy of data. For this purpose, we should know about the keys.

"A key is a single attribute or a combination of two or more attributes of a relation. It is used to identify one or more instance of the set."

Types of Keys: There are six types of keys which are as follows:

(a) Candidate key:

- Candidate keys are those attributes which have unique values. But Null value is not allowed in a candidate key.
- If there is no attribute in a relation containing unique value then combination of two attributes of that relation can make candidate key.

- There can be any number of candidate keys in a relation (table).

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10^{th}	Chennai	1010
2.	Akhil	12^{th}	Mumbai	1229
3.	Aditya	8^{th}	Bangalore	801
4.	Sachin	9^{th}	Gurgaon	906
5.	Rahul	11^{th}	Patiala	1112
6.	Sid	12^{th}	Mumbai	1225
7.	Kunal	8^{th}	Hyderabad	806
8.	Rohit	10^{th}	Chennai	1011

In the above relation "Student", S. No. and Roll Number both attributes have unique value, therefore they are candidate keys.

(b) Primary Key

- The attribute which have unique value is known as primary key. But Null value is notallowed in a primary key.
- Primary key is used for query purposes.
- There will be only one primary keys in a relation (table).

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10^{th}	Chennai	1010
2.	Akhil	12^{th}	Mumbai	1229
3.	Aditya	8^{th}	Bangalore	801
4.	Sachin	9^{th}	Gurgaon	906
5.	Rahul	11^{th}	Patiala	1112
6.	Sid	12^{th}	Pune	1225
7.	Kunal	8^{th}	Hyderabad	806
8.	Rohit	10^{th}	Chennai	1011

In the above relation "Student", only one attribute Roll Number have unique value, therefore attribute Roll Number is a primary key.

(c) Alternate Key

- Alternate key also contains unique value.
- After identifying candidate keys, one key is known as primary key and another key (which is not selected as primary key) is known as alternate key.

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10^{th}	Chennai	1010
2.	Akhil	12^{th}	Mumbai	1229
3.	Aditya	8^{th}	Bangalore	801
4.	Sachin	9^{th}	Gurgaon	906
5.	Rahul	11^{th}	Patiala	1112
6.	Sid	12^{th}	Pune	1225
7.	Kunal	8^{th}	Hyderabad	806
8.	Rohit	10^{th}	Chennai	1011

In the above relation "Student", S. No. and Roll number both have unique values and called as candidate keys. S.No. is called alternate key and Roll number is known as primary key.

If Primary key: Roll Number

Then Alternate key: S. No.

(d) Composite Key

- Sometimes in a relation, there is no primary key. In that situation, more than one attributes are used to identify a unique entity.

- The combination of those attributes are known as composite key.

Name	Class	City	Age
Rohit	10^{th}	Chennai	16
Akhil	12^{th}	Mumbai	18
Aditya	8^{th}	Bangalore	14
Sachin	9^{th}	Gurgaon	15
Rahul	11^{th}	Patiala	17
Sid	12^{th}	Pune	18
Kunal	8^{th}	Hyderabad	14
Rohit	10^{th}	Chennai	17

In the above relation "Student‖, Name and Age are used to identify an entity, therefore both are called composite key.

(e) Artificial Key: Sometimes in a relations, there is no primary key and there is no possibility to make primary key.In that situation, we can insert a key in a relation which has no meaning is known as an artificial key.

Student

Name	Class	City	Age
Rohit	10^{th}	Chennai	16
Akhil	12^{th}	Mumbai	18
Aditya	8^{th}	Bangalore	14
Sachin	9^{th}	Gurgaon	15
Rahul	11^{th}	Patiala	17
Sid	12^{th}	Pune	18
Kunal	8^{th}	Hyderabad	14
Rohit	10^{th}	Chennai	17

In the above relation "Student", there is no unique key. We can insert a new attribute S.No. into relation as a artificial key. This attribute S. No. has no meaning.

Student

S. No.	Name	Class	City	Age
1	Rohit	10^{th}	Chennai	16
2	Akhil	12^{th}	Mumbai	18
3	Aditya	8^{th}	Bangalore	14
4	Sachin	9^{th}	Gurgaon	15
5	Rahul	11^{th}	Patiala	17
6	Sid	12^{th}	Pune	18
7	Kunal	8^{th}	Hyderabad	14
8	Rohit	10^{th}	Chennai	17

(f) Foreign Key

- Foreign key is the attribute of a relation which acts as a primary key of another table.
- Foreign key .allows only those values which appears in primary key or may be null.
- Foreign key is used to make a relationship between two tables and to maintain thereferential integrity.

Class

Class	Class Incharge
8^{th}	Mrs. Nidhi
9^{th}	Ms. Aastha
10^{th}	Mrs. Prathiba
11^{th}	Mrs. Manmeet
12^{th}	Mrs. Navreet

Student

S. No.	Name	Class	City	Roll Number
1	Rohit	10^{th}	Chennai	1010
2	Akhil	12^{th}	Mumbai	1229
3	Aditya	8^{th}	Bangalore	801
4	Sachin	9^{th}	Gurgaon	906
5	Rahul	11^{th}	Patiala	1112
6	Sid	12^{th}	Pune	1225
7	Kunal	8^{th}	Hyderabad	806
				1011

In the above relation, class acts as an foreign key.

3. Relational Model Constraints/Integrity Constraints

- Integrity constraints ensure that changes made to the database by authorized users and any change do not lose the data.
- Integrity constraints also ensure the restrictions on the data and provide the security against the accidental damage to the database.

Type of Constraints

(a) Domain Constraint:

- It ensures that each attribute have a correct value.
- The data type associated with domains includes integer, character, string, data, and time.
- For example: A is not allowed in the attribute Roll Number because Roll Number is an integer attribute.

S. No.	Name	Class	City	Roll Number
1	Rohit	10^{th}	Chennai	1010
2	Akhil	12^{th}	Mumbai	1229
3	Aditya	8^{th}	Bangalore	801
4	Sachin	9^{th}	Gurgaon	906
5	Rahul	11^{th}	Patiala	1112

6	Sid	12^{th}	Pune	1225
7	Kunal	8^{th}	Hyderabad	806
8	Rohit	10^{th}	Chennai	1011

(b) Tuple Uniqueness Constraint

- Relation is a set of tuples (rows).
- All tuples in a relation must be different from each other. It means there must be unique value or attribute by which we can identify a tuple.

(c) Key Constraint

- Primary key must have unique value in the relation (table).
- If S.No. is considered as a primary key then there must be unique value in this attribute. We cannot insert duplicate value in the primary key.

S. No.	Name	Class	City	Roll Number
1	Rohit	10^{th}	Chennai	1010
2	Akhil	12^{th}	Mumbai	1229
3	Aditya	8^{th}	Bangalore	801
4	Sachin	9^{th}	Gurgaon	906
5	Rahul	11^{th}	Patiala	1112
5/3	Sid	12^{th}	Pune	1225
7	Kunal	8^{th}	Hyderabad	806
7/4	Rohit	10^{th}	Chennai	1011

(d) Entity Integrity: Entity integrity ensures that primary key cannot have NULL value.

S. No.	Name	Class	City	Roll Number
1	Rohit	10^{th}	Chennai	1010
2	Akhil	12^{th}	Mumbai	1229
3	Aditya	8^{th}	Bangalore	801
4	Sachin	9^{th}	Gurgaon	906
5	Rahul	11^{th}	Patiala	1112

	Sid	12th	Pune	1225
7	Kunal	8th	Hyderabad	806
	Rohit	10th	Chennai	1011

(e) Referential Integrity: Referential integrity ensures that if a foreign key of a table I refers to the primary key of table II, then every value of the foreign key in table I must be null or be available in table II.

I	
Class	**Class Incharge**
8th	Mrs. Nidhi
9th	Ms. Aastha
10th	Mrs. Prathiba
11th	Mrs. Manmeet
12th	Mrs. Navreet

II				
S. No.	**Name**	**Class**	**City**	**Roll Number**
1	Rohit	10th	Chennai	1010
2	Akhil	12th	Mumbai	1229
3	Aditya	8th	Bangalore	801
4	Sachin	9th	Gurgaon	906
5	Rahul	11th	Patiala	1112
6	Sid	12th	Pune	1225
7	Kunal	8th	Hyderabad	806
8	Rohit	10th	Chennai	1011

Operations of Relational Model

1. **Insert Operation:** Relational model does not suffer from any insert anomaly.
2. **Update Operation:** Relational model does not suffer from any update anomaly.

3. **Delete Operation:** Relational model does not suffer from any delete anomaly.
4. **Retrieve Operation:** Retrieve operation for relational data model is simple and symmetric.

Advantages of Relational Data Model

1. **Simplicity:** A relational data model is even simpler than hierarchical and network models. It frees the designers from the actual physical data storage details, thereby allowing them to concentrate on the logical view of the database.
2. **Structural Independence:** Unlike hierarchical and network models, the relational data model does not depend on the navigational data access system. Changes in the database structure do not affect the data access. Ease of design, implementation, maintenance and uses: The relational model provides bothstructural independence and data independence. Therefore, itmakes the database design, implementation, maintenance and usage much easier.
3. **Flexible and Powerful Query Capability:** Its structured query capability makesad hoc queries a reality.The relational database model provides very powerful, flexible, and easy-to-use query facilities. Information in a table can be easily modified.
4. **Easy to Use:** To collect the information in table consisting columns and rows is very easy.
5. **Security:** In relational model, security control and authorization can be implemented.

Disadvantages of Relational Data Model

1. **Hardware overheads:** The relational data models need more powerful computing hardware and data storage devices to perform complex tasks. Consequently, they tend to be slower than the other database systems. However, with rapid advancement in computing technology and development of much more efficient operating systems, the disadvantage of being slow is getting faded.
2. **Easy-to-design capability leading to bad design:** Easy-to-use feature of relational database results into untrained people generating queries and reports

without much understanding and giving much thought to the need of proper database design. With the growth of database, the poor design results into slower system, degraded performance and data corruption.

4.2 COMPARISON OF DATA MODELS

Sr. No.	Hierarchical Model	Network Data Model	Relational Data Model
1.	Hierarchical data model represents data in a tree format where Parent and Child relationship is represented to show association.	Network model represents data in graphs where data is a record which is linked by pointers.	Relational data model logically represents data in Tabular form where data is placed in row and column.
2.	Many to many relationship cannot be expressed in hierarchical model	Many to many relationship can be expressed in hierarchical model.	Many to many relationship can be expressed in hierarchical model.
3.	It is good for expressing data in < parent child relationship	It is good for modeling of many to many relationships.	It is good for modeling real world entities.
4	Relationship are represented by pointer and relationship among records are physical in nature	Network model also represents relationship through pointers and nature of the relationship is physical.	Relational model is stored data in form of rows and column. There is no physical connection is established between different tables whereas connection is logical in nature and established through keys.
5	Searching of a particular record is a time consuming task as to reach a particular child we have to process through its parent record.	Searching of a particular record is easy since there are multiple access path available to reach a node in graph.	In case of relations tables we use concept of keys to identify the records and search a key through indexing is quite simple task
6.	Insertion is done in the form of parent node and child node relationship. We cannot insert child node in tree without parent node.	Network model insertion can be performed by inserting new node in the graph with ease and has no insertion anomaly	in Relation model , new record can be added any time and has no insertion anomaly

7.	Updation operation may results in inconsistency as there are multiple child records in a tree	Updation operation is free from any anomaly as there is only single occurrence of each record in a graph which may be connected with multiple records.	Updation operation is safe in a relational model as duplication of record is avoidable by applying normalisation and Primary keys relationships
8.	Hierarchical model is based on parent child relationship and deleting of child is easy as compare to parent, if we delete parent then child node will automatically deleted from the tree.	There is no deletion anomaly as deleting of one node does not affect other nodes due to many to many relationships.	The deleting of record from a relation is again a simple process and there is no anomaly related to deleting of records. Deletion of reference records is not allowed as it may linked to other records

4.3 RELATIONAL ALGEBRA AND RELATIONAL CALCULUS

The relational model uses the concept of a mathematical relation in the form of table of values which acts as building block. The table is a logical representation of data in the form of rows and columns. The relational algebra is a formal query language applied on relational model. It is a procedural language which specifies the operations to be performed on relations. The operations are performed in form of sequence of algebra operations which results in a new relation/table. The relational algebra operations can be classified into two types.

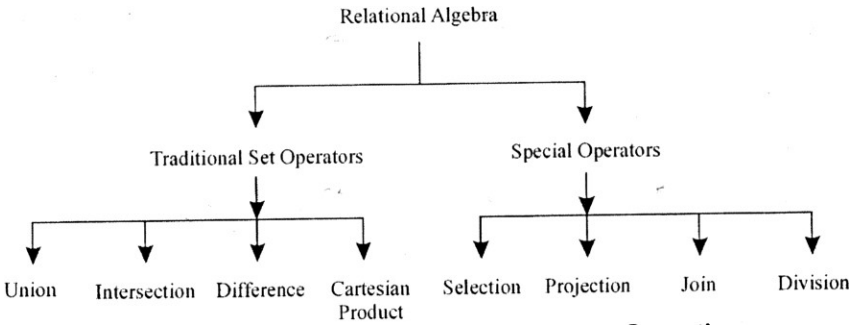


Figure 4.1: Classification of Relational Algebra Operations

Relational calculus is a non-procedural query language. Here, no procedures are provided to generate result based on 'query'. In relational calculus, query is expressed as

variables and formulas on these variables. There are two types of relational calculus: tuple Relational Calculus and Domain Relational Calculus:

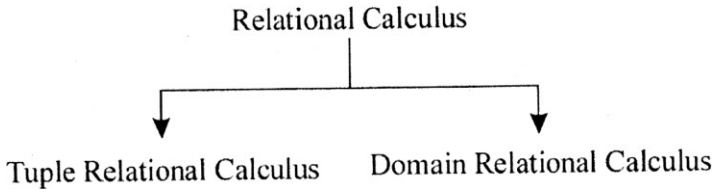


Figure 4.2: Classification of Relational Calculus

4.4 RELATIONAL ALGEBRA

1. Relational algebra is a procedural query language.
2. It consists of set of operators that take one or two relations as input and produce a new relation as output.
3. It uses relational operators.
4. It is of mainly two types which are as follows:

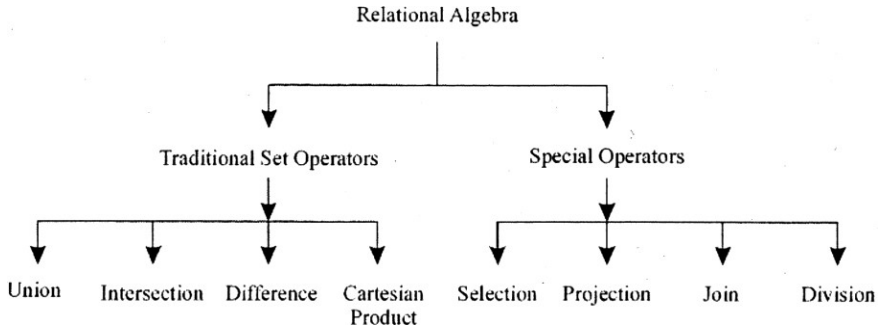


Figure 4.3: Classification of Relational Algebra

I. Traditional Set Operators

(a)Union Operator

(b)Intersection Operator

(c)Difference Operator

(d)Cartesian Product Operator

(a) Union Operator:

- Union of two relations is the set of all elements belonging to both relations.
- Result must not contain duplicate elements.
- It is denoted by U.

- For example: We want to list all the names and roll numbers which are present in both tables:'A' and 'B'.

AB

Name	Roll Number
Akhil	211
Monika	129

Name	Roll Number
Aastha	112
Akhil	211

Formula:πName, Roll Number (A) U πName, Roll Number (B).

AUB

Name	**Roll Number**
Akhil	211
Monika	129
Aastha	112

(b) Intersection Operator:

- Intersection of two relations produces a relation which contains all elements that are common to both relations.
- It is denoted by ∩.
- For example: We want to list only those names and roll numbers which are common inboth tables 'A' and 'B'.

A

Name	Roll Number
Akhil	211
Monika	129

B

Name	Roll Number
Aastha	112
Akhil	211

Formula:πName, Roll Number (A) ∩πName, Roll Number (B)

A∩B

Name	**Roll Number**
Akhil	211

(c) Difference Operator

- Difference operator is used to find those tuples which are present in one relation but not in another relation.
- It is denoted by (-) sign.
- For example: We want to list those names and roll numbers which are present in table 'A[9] only, not in table'B'.

AB

Name	Roll Number
Akhil	211
Monika	129

Name	Roll Number
Aastha	112
Akhil	211

Formula:πName, Roll Number (A) –πName, Roll Number (B)

A-BB-A

Name	Roll Number
Monika	129

Name	Roll Number
Aastha	112

(d) Cartesian Product

- Cartesian product operator is used to combine information from any two relations.
- It is denoted by (X) symbol.
- For example: We want to list the names of employees with all departments of tables 'A'and'B'.

AB

Name	Emp_No	Dept_Id
Akhil	101	11
Monika	102	12
Aastha	101	11

Dept_Name	Dept_Id
Production	11
Accounts	12

Formula: πName (A) X π Dept_Name (B)

AXB

Name	Dept_Name
Akhil	Production
Akhil	Accounts
Monika	Accounts
Monika	Production
Aastha	Production
Aastha	Accounts

II. Special Operators

(a) Selection Operator

(b) Projection Operator

(c) Join Operator

(d) Division Operator

(a) Selection Operator

- Selection operator selects tuples (rows) that satisfy a given condition.
- It is denoted by lower Greek letter sigma (a).
- We can also use folio wing symbols: = >,<>>=,<= #
- For example: We want to list the tuples (employees) who live in city 'chd'.

Formula: σcity = "chd" (employee)

(b) Projection Operator

- Projection operator returns a new relation as output with certain attributes.
- It is denoted by Greek letter pie (π).
- For example: We want to list all the emp_no and name of employee.

Formula: πemp_no, name (employee)

(c) Join Operator

- Join operator is also known as natural join operator.
- It is denoted by the symbol ([><]).

- Cartesian product operator is used to combine two tables, but the output of Cartesianproduct is not correct
- Join operator is used to combine the two tables instead of Cartesian product operator.
- For example: We want to combine the two tables 'A'and 'B'.

A

Name	Emp_No	Dept_Id
Akhil	101	11
Monika	102	12
Aastha	101	11

B

Dept_Name	Dept_Id
Production	11
Accounts	12

Formula : π Name, Dept_Id (A) ($\bowtie$) π Dept_Name, Dept_Id (B)

A $\bowtie$ B

Name	Dept_Name
Akhil	Production
Monika	Accounts
Aastha	Production

(d) Division Operator

- Division operator will work on two relations (tables).
- It make another relation consisting of values of an attribute of one relation that match all the values in the another relation.
- It is denoted by the (÷) symbol.

A

Branch_Name	Branch_Id
Chd	11
Delhi	12
Mumbai	13

B

Branch_Name	Branch_Id
Akhil	Delhi
Monika	Chd
Aastha	Mumbai
Ankush	Delhi
Radhika	Chd

Formula: πName (A ÷B)

(A ÷B)

Name
Akhil

4.5 RELATIONAL CALCULUS

1. It was first proposed by E.F.Codd.
2. It is a formal language used to symbolize logical arguments in mathematics.

3. In relational calculus, query is expressed as formula containing number of variables and expression.
4. User will only tell the requirement without knowing the methods of retrieval.
5. User is not concerned with the procedure to obtain the results.
6. It is the responsibility of DBMS to transform these queries and give the result to the user.
7. Relational calculus is of mainly two types which are as follows:

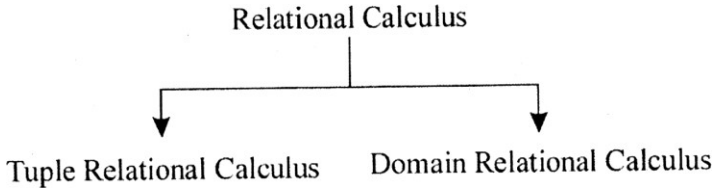


Figure 4.4: Classification of Relation Calculus

I. Tuple Oriented Relational Calculus

- It is based on specifying a number of tuples variables.
- The query of tuple relational calculus is

  {t/COND(t)}

  t-> is tuple variable

  COND (t)->is conditional expression.
- The result of such query is a relation that contains all the types (rows) that satisfy condition COND (t).

  Query of relational calculus is:

  {t. title, t. author/Book(t) and t. PRICE > 100}

  It will give us title, author of all the books whose price is greater than 100.

  Expression of tuple relational calculus is:

  {t1. A1, t2.A2, t3.t3,…tn.An/COND (t1, t2, t3, …tn)}

  t1, t2 …. are tuple variables.

  A1, A2 … are the attributes of relations.

  COND is condition.

II. Domain oriented relational calculus

- Domain calculus is different from tuple calculus in the type of variables used in formula.
- In domain oriented relational calculus, variable range will be single value rather than multiple values.
- Expression of domain oriented relational calculus is:

  {X1, X2, …Xn | COND (X1, X2, … Xn)}

  X1, X2, …Xn are domain variables.

  COND is condition or formula of domain relation calculus.

  i.e. Get employee no. of for job clerk

  EX where EMP (emp no: EX, job = 'clerk')

  Get employee name that belongs to dept no. 10 and having salary > 2000.

  Ex where EMP (ename: EX, deptno = 10, sal> 2000)

4.6 DIFFERENCE BETWEEN RELATIONAL ALGEBRA AND RELATIONAL CALCULUS

Sr. No.	Relational Algebra	Relational Calculus
1	It is a procedural method of solving the queries.	It is a non-procedural method of solving the queries.
2	It is used as a vehicle for implementation of relational calculus.	The queries of relational calculus are transformed into equivalent relational algebra format and then implemented with the help of relational algebra operators.
3	The solution to the database access problem using a relational algebra is obtained by stating what is required? And what are the steps to obtain that / information?	The solution to the database access problem using a relational calculus is obtained by stating what is required? And system will find the answer?

Questions

1. What do you mean[1] by data models? Explain the answer.
2. How can we classify data models?
3. What do mean by relationships in a data model
4. What is an attribute in data modelling?
5. Explain the Relational Model? Write advantages and disadvantages.
6. Explain the Hierarchical Model? Write advantages and disadvantages.
7. Explain different operations that can be performed on Hierarchical
8. Compare different data models.
9. List the various relational operators available in a relational model.
10. What is the difference between select and project operators?
11. Explain the various set operators available in relational algebra.
12. What is the Cartesian product operation? Why is it rarely used without a select operation?
13. What is the significance of the join operator? Explain the different types of join.
14. Explain relational calculus in detail.

5.1 NORMALIZATION

5.2 FUNCTIONAL DEPENDENCY

5.3 FULLY FUNCTIONAL DEPENDENCY

5.4 PARTIAL FUNCTIONAL DEPENDENCY

5.5 TRANSITIVE FUNCTIONAL DEPENDENCY

5.6 MULTI VALUED DEPENDENCY

5.7 FIRST NORMAL FORM (1NF)

5.8 SECOND NORMAL FORM (2NF)

5.9 THIRD NORMAL FORM (3NF)

5.10 BOYCE CODD NORMAL FORM (BCNF)

5.11 FOURTH NORMAL FORM (4NF)

5.12 FIFTH NORNAL FORM (5NF)

5.1 NORMALIZATION

"Normalization is the process of efficiently organizing data to minimize redundancy in a database and makes database more flexible."

1. E.F. Codd introduced the concept of normalization.
2. Normalization technique is used in designing relational model.
3. It improves database design and removes anomalies for database activities.
4. Its objective is to reduce the redundancy (duplicity) and eliminates the insertion, updation, and deletion anomalies from the database.
5. To achieve its objective, it breaks the database into smaller tables and establishes the relationships between those tables.
6. It makes data consistent throughout the database.
7. Normalization follows some rules. Each rule is known as normal form.
8. E.F. Codd introduced the first normal form (INF) in 1970.
9. He introduced the second normal form (2NF), third normal form (3NF) in 1971 and boyce codd normal form (BCNF) in 1974.
10. For many applications, third normal form (3NF) is necessary.
11. Fourth normal form (4NF) was introduced by Ronald Fagin in 1977.
12. Normal forms are numbered from lowest (INF) to highest (5NF).
13. ***The following are the disadvantages of normalizations:***

- It is a difficult and time consuming process.
- Sometimes, the performance of database degrades from lowest (INF) to highest (5NF).

14. Un-normalized Form (UNF) is one in which a table contains non atomic values at each row. Non atomic values need further decomposition for simplification. For the simplification, un-normalized form goes into first normal form.
15. ***The levels/steps of normalization are as follows:***

Un-normalized Form (UNF)

↓

First Normal Form (INF)

↓

Second Normal Form (2NF)

↓

Third Normal Form (3NF)

↓

Boyce Cod'd Normal Form (BCNF)

↓

Fourth Normal Form (4NF)

↓

Fifth Normal Form (5NF)

Fig. 5.1: Steps of Normalization

5.2 FUNCTIONAL DEPENDENCY

1. Functional dependency is an association between two attributes (columns) of the same relation (table).
2. It is basically a constraint between two sets attributes from the same relation in a database.
3. One attribute is called determinant and other is called determined.
4. For each value of determinant, there is only one value of determined.
5. For example: A→B
   - "B is functionally dependent on A" because for each value of attribute 'A', there is exactly one value of attribute 'B'.
   - If A is determinant and B is determined then we can say that "A functionally determines B" OR "B is functionally dependent on A".

Supplier

Sr. No,	Name	Status	City
S1	Akhil	10	Delhi
S2	Monika	2.0	Patiala
S3	Aastha	30	Delhi

In above table "Supplier", attribute 'Name' is functionally dependent (FD) on attribute 'Sr. No.' because 'Name' has only one value for given 'Sr. No.'.

We can say Sr. No. $\rightarrow$ Name

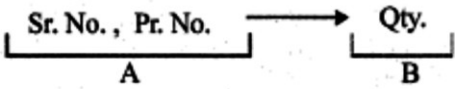
'Sr. No' is determinant and 'Name' is determined.

But attribute 'City' is not functionally dependent (FD) on attribute ' Sr. No.' because 'City' has more than one value for given 'Sr. No.'.

5.3 FULLY FUNCTIONAL DEPENDENCY

1. Fully functional dependency is a functional dependency in which all the non-key attributes are dependent on the key attributes.
2. For example: A $\rightarrow$ B
   - "B is fully functionally dependent on A" because 'B' is functionally dependent on 'A' but not on any proper subset of 'A'.
   - "B is fully functionally dependent on A" means we cannot identify the value of 'B' only from 'A', we can identify the value of 'B' from 'A' and another attribute from the same relation. Another attribute will help the 'A' to find the value of 'B'.
   - If we delete any attribute from the relation, then it will violate the concept of functional dependency.
3. In the below table, Qty. is F.F.D. on 'Sr. No.' and 'Pr. No.' because we can get the value of Qty. only by the combination of both 'Sf. No.' and 'Pr. No.'.

Sr. No.	Pr. No.	Qty.
S1	P1	270
S1	P2	300
S1	P3	700
S2	P1	270
S2	P2	700
S3	P2	300



5.4 PARTIAL FUNCTIONAL DEPENDENCY

1. Partial functional dependency occurs, when some non-key attribute depends on primary key attribute.
2. For example: A $\rightarrow$ B

The attribute 'B' is partial functional dependent on attribute 'A', if there is some attribute that can be removed from 'A' and yet the dependency holds.

5.5 TRANSITIVE FUNCTIONAL DEPENDENCY

1. Transitive functional dependency occurs, when some non-key attribute depends upon other non-key attributes.
2. For example: There are three attributes 'AVB' and 'C'.
   - A$\rightarrow$B
   - B$\rightarrow$C
   - $\Rightarrow$A$\rightarrow$C

   It means 'C' is transitively dependent on 'A'.

5.6 MULTI VALUED DEPENDENCY

1. Multivalued dependency is a full constraint between two sets of attributes in a relation.
2. It plays a role in fourth normal form (4NF) of normalization.
3. For example: If there are three attributes 'A', 'B' and 'C' in a relation.
   - 'B' and 'C' are independent from each other.
   - 'B' and 'C' are multi valued fact about A.

     Then A $\rightarrow\rightarrow$B

     A$\rightarrow\rightarrow$ C
   - Then we can say that "A multi determines B" OR "B is multi dependent on A".

Course_Student_Book

Course	Student	Book
Chemistry	Akhil	B1
Chemistry	Akhil	C1
Physics	Moaika	A1
Physics	Monika	D1
Chemistry	Aastha	Bl
Chemistry	Aastha	Cl
English	Rohit	A1
English	Rohit	D1

Course $\rightarrow\rightarrow$ Student

Course $\rightarrow\rightarrow$ Book

5.7 FIRST NORMAL FORM (1NF)

1. E.F. Cold introduced the first normal form (1NF) in 1970.
2. First normal form (1NF) eliminates the repeating columns from an un-normalized table.
3. In 1NF, there is no repeating column (group).
4. We convert un-normalized table into normalized for.
5. Primary key is required in each table to identify a record.
6. The purpose of primary key is to uniquely identify a record.
7. First normal from depends on the functional dependency.
8. Formula : f(x)=y

   For every value of x, there is only one value for y.
9. For example: The following table "Student" having columns (Name, Course, Roll Number) is an un-normalized table. We have to convert this un-normalized table into normalized table.

Student

Name	Course	Roll Number
Akhil	Science	211, 128
Monika	Computer	129
Aastha	Management	112

The above table "Student" is un-normalized because it contains more than one value for the column 'Roll Number'. 'Akhil' has two values (211, 128) for the column 'roll number' which is not possible. For normalization, there should be only one value in one column.

The following are two methods to convert un-normalized table into normalized table:

- **Method 1:** To convert the un-normalized table "Student" into normalized form, we decompose (divide) this un-normalized table into two tables.

Student 1

Name	Course
Akhil	Science
Monika	Computer
A'astha	Management

Student 2

Name	Roll Number
Akhil	211
Akhil	128
Monika	129
Aastha	112

- **Method 2:** To convert the un-normalized table "Student" into normalized form, we convert this this un-normalized table into flat table.

Student

Name	Course	Roll Number
Akhil	Science	211
Akhil	Science	128
Monika	Computer	129
Aastha	Management	112

Anomalies in First Normal Form (INF)

1. **Insert Anomaly: We** cannot insert any information of new student in table "Student" until he join any course. Similarly, we cannot insert any information about the course until there is any student. This phenomenon is known as insert anomaly.

Student

Name	Roll Number	Course
Akhil	211	Science
Monika	129	Computer
Aastha	112	Management

Rohit	**111**

The details of new student 'Rohit' cannot insert into the table "Student" until he join any course. It is called insert anomaly.

2. **Update Anomaly :In** the update anomaly, if we want to change (update) the course of any student, then we have to change (update) the multiple records. If we change the course of the student but forget to change the details of that student from all the locations where it occures, then data become inconsistent/This phenomenon is known as update anomaly.
3. **Delete Anomaly**: If we delete any course from table "Student", then all the related information to that course automatically deletes.

For Example: if we delete the course 'management' from the table "Student", then it automatically ceases the name 'Aastha' and roll number '112'.

Student

Name	Roll Number	Course
Akhil	211	Science
Monika	129	Computer
Aastha	112	Management

After deletion, table "Student will be look like:

Student

Name	Roll Number	Course
Akhil	211	Science
Monika	129	Computer

5.8 SECOND NORMAL FORM (2NF)

1. E.F. Codd introduced the second normal form (2NF) in 1971.
2. A relation is in 2NF if it fulfills the following conditions
   - Relation should be in INF and
   - Every non-key attribute (non-prime attribute) is fully functionally dependent on Primary key.
3. For example-.The following table "Products" having columns (Item, Price, Quantity, Order Number, and Order Date) is in INF.

Products

Item	Price	Quantity	Order Number	Order Date
Mobile	2000	20	11	1-7-2015
Sunglasses	1000	15	12	2-7-2015

Watch	800	18	13	3-7-2015
Wallet	600	12	14	4-7-2015

- The table "Products" has two primary key columns (Item and Order Number).
- Price (non-primary key column) is fully functionally dependent on Item (prime key column).
- Order Date (non-primary key column) is fully functionally dependent on Order Number (prime key column).
- The table "Products" can be converted into second normal form (2NF) by decomposing it into sub tables such as:

Item	Price
Mobile	2000
Sunglasses	1000
Watch	800
Wallet	600

Order Number	Order Date
11	1-7-2015
12	2-7-2015
13	3-7-2015
14	4-7-2015

Item	Quantity	Number
Mobile	20	11
Sunglasses	15	12
Watch	18	13
Wallet	12	14

Anomalies in Second Normal Form (2NF):

1. **Insert Anomaly:** Second form (2NF) also Suffers from the inset anomaly same like the first normal form (1NF). We cannot insert any information of 'Price' in table "Products" until is associates with any 'item'. Similarly, we cannot insert any information about the 'item' in the table "Products" until its price is fixed. This phenomenon is known as insert anomaly.

Products

Item	Price	Quantity	Order Number	Order Date
Mobile	2000	20	11	1-7-2015
Sunglasses	1000	15	12	2-7-2015
Watch	800	18	13	3-7-2015
Wallet	600	12	14	4-7-2015
	1200	16	15	5-5-2015

The details of new price '1200' cannot insert into the table "Products" until it associates with any 'item'. We cannot left blank the value of any column. It is called insert anomaly.

2. **Update Anomaly:** In the update anomaly, if we want to change (update) the 'price' of any 'item', them we has to change (update) the multiple records. It we change the 'price' of any 'item' but forget to change the details of that 'item' from all the locations where is occurs, then data become inconsistent. This phenomenon is known as update anomaly.
3. **Delete Anomaly:** Like 1NF, 2NF also suffers with delete anomaly. If we delete any 'item' from table "Products", then all the related information to that 'item' automatically deletes. For example : if we delete the item 'watch' from the table "products", then it automatically deletes it's all related information (price, quantity, order number, order date).

Products

Item	Price	Quantity	Order Number	Order Date
Mobile	2000	20	11	1-7-2015
Sunglasses	1000	15	12	2-7-2015

Watch	800	18	13	3-7-2015
Wallet	600	12	14	4-7-2015

After deletion, table "products" will be look:

Product

Item	Price	Quantity	Order Number	Order Date
Mobile	2000	20	11	1-7-2015
Sunglasses	1000	15	12	2-7-2015
Wallet	600	12	14	4-7-2015

5.9 THIRD NORMAL FORM (3NF)

1. E.F. Codd introduced the third normal form (3NF) in 1971.
2. It means a relation (table) is in 3NF if it is in 2NF and there is no transitive dependency.
3. The objective to 3NF is to remove all transitive dependencies.
4. A relation is in 3NF if it fulfills the following conditions:
   - Relation should be in 2NF and
   - Every non-key attribute (non-prime attribute) is transitively dependent on Primary key only.
5. It removes the anomalies of 2NF.
6. For many applications, third-normal form (3NF) is necessary.
7. For example: The following table "Record" having columns (Name, Roll Number, System, Number, Hours_Rate) is in 2NF.

Record

Name	Roll Number	System Number	Hours Rate
Aastha	112	S1	20

Akhil	211	S2	18
Monika	129	S3	17
Rohit	219	S2	15
Aditya	285	S3	16
Kunal	712	S4	12
Sachin	125	S1	23
Rahul	231	S4	25
Siddharth	123	S5	13

- 'Name' is a primary key and the entire non-key attributes (Roll Number, System Number, Homrs_Rate) are dependent on it.
- To convert the table "Record" into 3NF, we decompose it into two tables (Student 'Record, Charge Record).

Student Record

Name	Roll Number	System Number
Aastha	112	SI
Akhil	211	S2
Monika	129	S3
Rohit	219	S2
Aditya	285	S3
Kunal	712	S4
Sachin	125	S1
Rahul	231	S4
Siddharth	123	S5

Charge Record

System Number	Hours Rate

SI	43
S2	33
S3	35
S4	37
S5	13

- Table "Student Record" provides the detail of student like Name, Roll Number and System Number used by him/her.
- Table "Charge Record" provides the details of system like System Number, Charges for using System.

Anomalies in Third Normal Form (3NF)

1. **Insert Anomaly:** Third normal form (3NF) is also suffers from insert anomaly but upto some extent. It is possible to insert in advance, the rate to be charged from student for a system.
2. **Update Anomaly:** If Hours_Rate for a system in table "System Record" changed (updated), then we need only to change a single record in table "Charge Record".
3. **Delete Anomaly:** It we delete the record of a student who is only student working on a particular system, then we will not lose the information of the system and hours_rate of that system.

5.10 BOYCE CODD NORMAL FORM (BCNF)

1. E.F. Codd introduced the Boyce Codd Normal Form (BCNF) in 1 974.
2. A relation is in BCNF, if it is in 3NF and every determinant (attribute) is a candidate key.
3. It means BCNF have multiple candidate keys (more than one primary key).

5.11 FOURTH NORMAL FORM (4NF)

1. Fourth normal form (4NF) was introduced by Ronald Fagin in 1977.

2. 2NF, 3NF and BCNF are concerned with functional dependencies whereas 4NF concerned with multivalued dependencies.
3. A relation is in 4NF if it is in 3NF or BCNF and contains no multi valued dependencies.
4. For example: The following table "Course_Student_Book" is in 3NF.

Course Student Book

Course	Student	Book
Chemistry	Akhil	Organic Chemistry
Chemistry	Akhil	Physical Chemistry
Physics	Monika	Optics
Physics	Monika	Mechanics
Chemistry	Aastha	Organic Chemistry
Chemistry	Aastha	Physical Chemistry
English	Rohit	English Literature
English	Rohit	English Grammar

- Attributes 'Student' and 'Book' are multivalued facts about the attribute 'Course'. There are many students for one course and many books for one course.
- The condition of 4NF is that there should be no multi valued attribute in a table.
- To convert the table "Course_Student_Book" into 4NF, we decompose it into two tables (Course Student, Course Book).
- Table "Course-Student" tells us which student is studying which course.
- Table "Course_Book" tells us which book is available for which course.

Course Student

Course	Student
Chemistry	Akhil
Physics	Monika
Chemistry	Aastha

Course Book

Course	Student
Chemistry	Organic Chemistry
Chemistry	Physical Chemistry
Physics	Optics

English	Rohit

Physics	Mechanics
English	English Literature
English	English Grammar

Note: If anew student 'Rahul' wants to join a course 'English' and use books of 'English' and 'Chemistry', then we have to insert new information of student 'Rahul'. We will insert the name 'Rahul' twice. First entry for 'English' and second entry for 'Chemistry'.

5.12 FIFTH NORNAL FORM (5NF)

1. A relation is in 5NF if it is in 4NF and based on join dependency.
2. Join dependency means when a table is decompose/divide into three or more tables, and then the resulting tables (divided tables) can be rejoined to form the original table.
3. The following are three sub tables (Course_Student, Course_Book and Student_Book) of original table "Course_Student_Book". *The table "Course_Student_Book" is used in the fourth normal form (4NF).*

Course_Student

Course	Student
Chemistiy	Akliil
Physics	Monika
Chemistry	Aastha
English	Rohit

Course	Book
Chemistry	Organic Chemistry
Chemistry	Physical Chemistry
Physics	Optics
Physics	Mechanics
English	English Literature
English	English Grammar

Student_Book

Student	Book
Akhil	Organic Chemistry
Akhil	Physical Chemistry
Monika	Optics
Monika	Mechanics
Aastha	Organic Chemistry
Aastha	Physical Chemistry '
Rohit	English Literature
Rohit	English Grammar

4. When we will join these three tables (Course_Student, Course_Book and Student_Book), then we will get the original table "Course_Student_Book".

Questions

1. What is Normalization? State and explain its types.
2. What is the need of Normalization of data? What are the various techniques for normalization in relational database model?
3. What is Functional dependency? Explain in detail Give an example also.
4. What do you mean by redundancy? Explain the ways to remove it from the database?
5. What do you mean by Normal forms? Explain the various types of it along with the suitable example.
6. What is the difference between First and second Normal Forms?
7. What is INF? Give example to demonstrate how INF improves a table.
8. Discuss 2NF. Discuss the problems that can be encountered in a table, which is in INF, How 2NF solve them?
9. Define 3NF? Discuss its need.
10. Explain Boyce Codd Normal Form.
11. Explain multivalued dependency. Give an example.

12. Explain Join dependency. Give an example.
13. Explain 4NF along with example.
14. What do you mean by FDs? Explain the Closure of a Set of FDs.
15. Explain 5NF along with example.
16. What is fully functional dependency? Give an example.

UNIT 6: MS ACCESS

6.1 MS-ACCESS

6.2 COMPONENTS OF MS ACCESS

6.3 CREATING A DATABASE

6.4 CREATING A TABLE

6.5 FORMS

6.6 QUERY

6.7 REPORTS

6.8 SECURITY

6.1 MS-ACCESS

- Microsoft Access is a computer package used to create and manage data-in computer. This software comes with MS Office.
- It is relational database system in which data is stored in tables.
- It can be used for personal information management as well as for commercial business database management.
- It provides various features (creating and editing) to the user. With the help of these features, user can organize the data into the forms and reports.
- It also allows the import and export of data to and from various applications.
- MS Access has an .mdb extension by default but in particular case of Access 2007, where the extension is now an .accdb extension.
- MS ACCESS has following advantages:

(a) **Import Data of Different Format:** One of the important advantages of Data Accessis capability to access data from different sources. Data of different formats can beimported and exported with user friendly interface.

(b) **Jet Batabase:**MS access is, equipped with Jet Database Format. The access databasecan usethis Jet Database Format to distribute the files to others who can use thiselsewhere.

(c) **Data concurrently:**Data in MS access can be used concurrently as locking facility isavailable to avoid concurrent access problem.

(d) **Compatibility with SQL:**It has strong features to edit anduse SQL command asAccess database acts as database server.

6.2 COMPONENTS OF MS ACCESS

MS Access is software package developed to provide database server facilities. It has following major components

1. Tables
2. Queries.
3. Forms
4. Reports
5. Macros
6. Modules

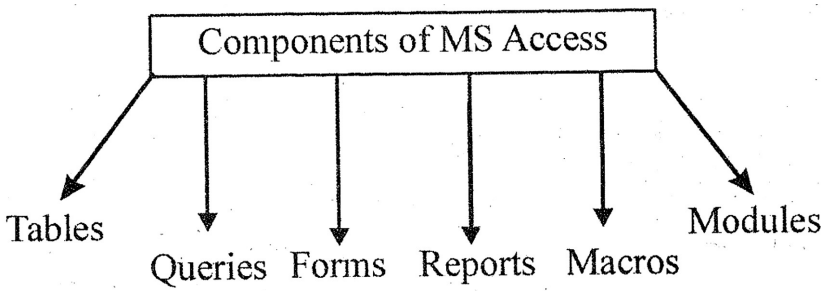


Fig. 6.1: Components of MS Access

6.2.1 Tables

- The tables are storage container of the data entered into the database
- A table is a collection of data about a specific topic.
- Table is also known as relation.
- It is a set of columns and rows. Each column is called a field.
- Within a table, each field must be given a name and no two fields can have the same name.
- Each value in a field represents a single category of data.
- For example, a table might have three fields: First Name, Middle Name andLast Nam. Thetable consists of three columns: one for first name, one for middle name, and one for lastname. In every row of the table, the Last Name field contains the last name, the First Namefield contains the first name, and the Middle name field contains the Middle name.
- Each row in a table is called a record.

- A primary key is a field or combination of fields that uniquely identify each record in a table.
- No two records in a table should have the same values in every field.
- Ifkeys are not set up correctly, with the correct relationships, then the database may be slow, give you the wrong results or not react the way you expect.
- MS ACCESS is a relational database system in which relations (tables) are used to store the information.
- When the information is stored into different tables then there is a need to establish relationship between tables.
- Relationships are the bonds we build between the tables. They join tables that have associatedelements. To do this there is a field in each table, which is linked to each other, and have thesame values. The field is called primary key in first table and foreign key in secondary table.

6.2.2 **Queries**

- A query is required to access the database.
- It is used to view, change and analyze the data in different ways.
- It is considered as 21 source to create forms and reports.
- It has created a GUI (Graphical User Interface) called a 'Query Designer' that easilyallows users to create queries without prior knowledge of SQL.
- The 'Query Designer' allows the users to select the data sources of the query (which canbe tables or queries) and select the fields they want returned by clicking and dragging them into the grid.
- Access allows users to view and manipulate the SQL code if desired. Any Access table, including linked tables from different data sources, can be used in a query.

6.2.3 **Forms**

- Forms are the primary interface through which the users of the database enter data.
- Forms act as input containers and provide a quick and efficient way to modify and insert records into the databases.
- The person who enters the data will interact with forms. The programmer can set the forms to show only the data required.
- Different types of filters can be used to take correct information from users. It also allows users a method of navigating through the system.

6.2.4 **Reports**

- Reports are used to show the output of processing or to show results of manipulation of the data you have entered into the database.
- Unlike forms, reports cannot be edited.
- Reports are intended to be used to output data to another device or application, i.e. printer, fax, Microsoft Word or Microsoft Excel.
- Reports contain detailed or summary information of the data that is contained in one or more tables in the databases.
- Reports are easily generated by using the Report Wizard.

6.2.5 **Macros**

- Macros are automatic methods that may be carrying out a series of actions for the database.
- Access gives a selection of actions that are carried out in the order we enter.
- It can perform a number of the common tasks that we can also use Visual Basic code to perform and can dramatically increase the productivity when working with the database.
- It can open forms, run queries, change values of a field, run other macros, etc. the list is almost endless.

6.2.6 **Modules**

- A module is a collection of declarations, statements, user-defined functions, subroutines and procedures that are stored together as a unit.
- Modules are very similar to macros since they are objects that add functionality to the database.
- ***Modules are basically of two types:***

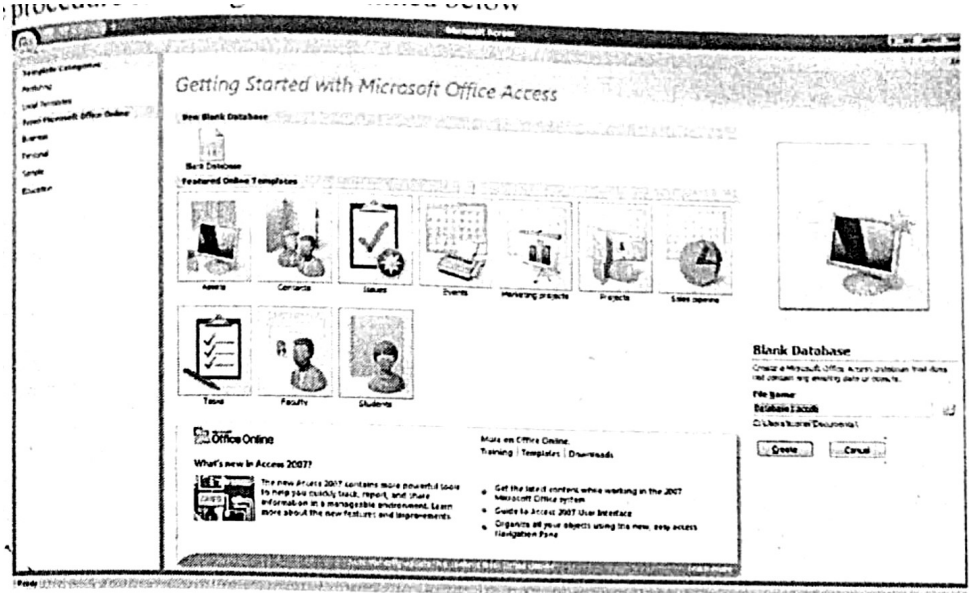
(a) Class Modules: Class modules contain procedures that are associated with a specific form or report it is attached to.

(b) Standard Modules: Standard modules contain general procedures that aren't associated with specific objects. Standard modules are listed under Modules in the Navigation Pane, whereas class modules are not.

6.3 **CREATING A DATABASE**

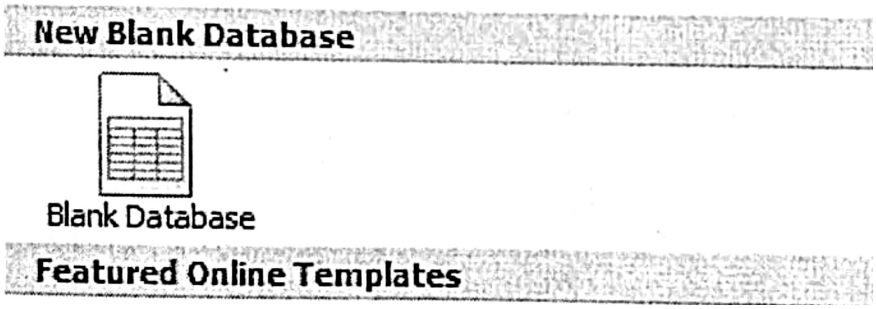
- The first step in creating an Access database is to create a blank database file.
- This is done from the Getting Started Screen when we launch Access.
- The file is saved onto one of your PCs folders (which we specify).

- The procedure for doing this is outlined below



Select Blank Database Template

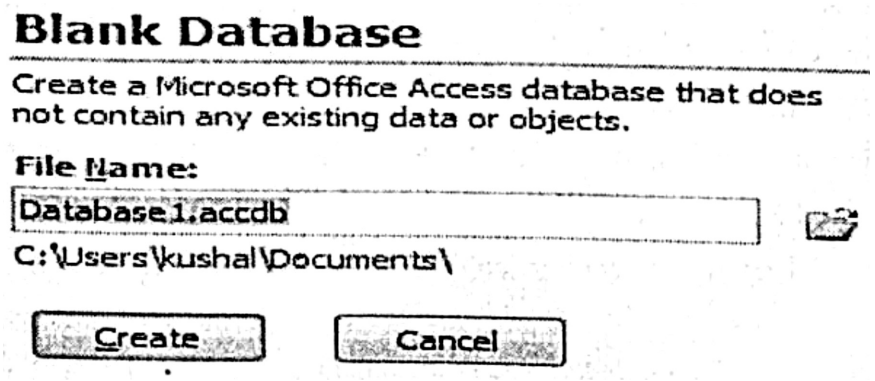
Towards the top left of the screen we will see a "Blank Database" icon.



Click this icon to bring up the Blank Database side bar on the right hand side of the screen. This is where we will enter details about the database file that we are about to create.

1. Enter filename for your Access 2007 database

Begin by entering the name that we want to call thedatabase in the filename textbox.



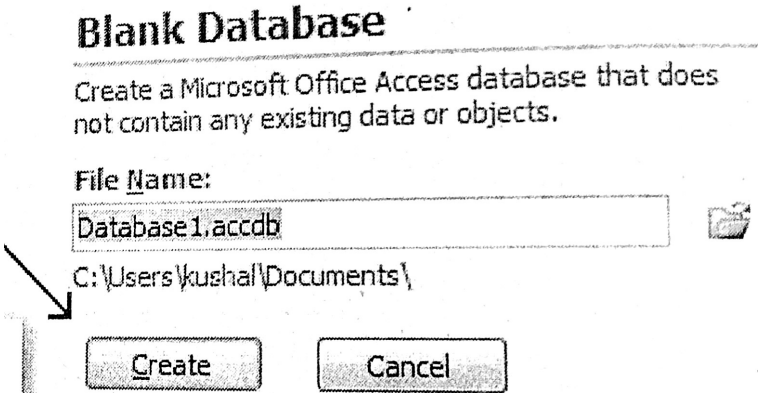
2. Browse and select folder

Next click the folder icon and browse for a folder to put our database. Once selected, we should see the file path below the textbox.

3. Click Create

All we need to do now is click the "Create" command button below, and our database file saves to the location, that we specified, and opens to work on.

We are now ready to work on our newly created database file. The next step is to create an Access table



6.4 CREATING A TABLE

- Like all other databases. Access 2007 stores data in tables.
- They look a lot like the cells of a spreadsheet with columns and rows.
- Each horizontal column represents a table record, and each vertical column represents a table field.

The following is the example of creation of table:

ID Number	FirstName	Surname	Age
1	Amit	Sharma	20
2	Rohit	Pathak	25
3	Amu	Verma	22

In the above example, there are four fields containing information about an individual's ID number, first name, surname and age.

Below the field headings, there are 3 records containing information or data for each

individual. As such, a database table is a list with each column containing the same specific sort of information. Each row of information is an individual record' that might relate to a particular person, a business, or a product etc.

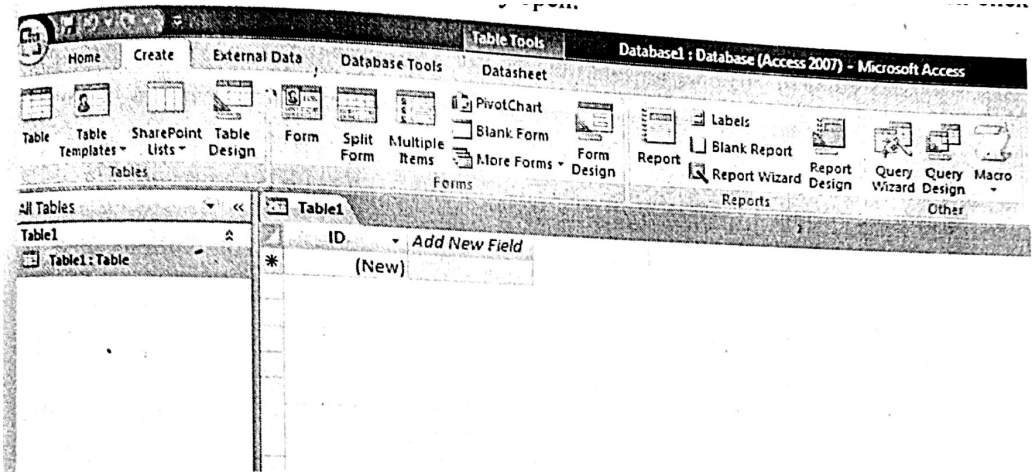
- When planning a database table, most database designers will decide which columnheadings or fields theyare going to use. This is the basis of the table structure. The actual is added later and is not a part of the design process.

Creating a Table by Design View

1. Open database file

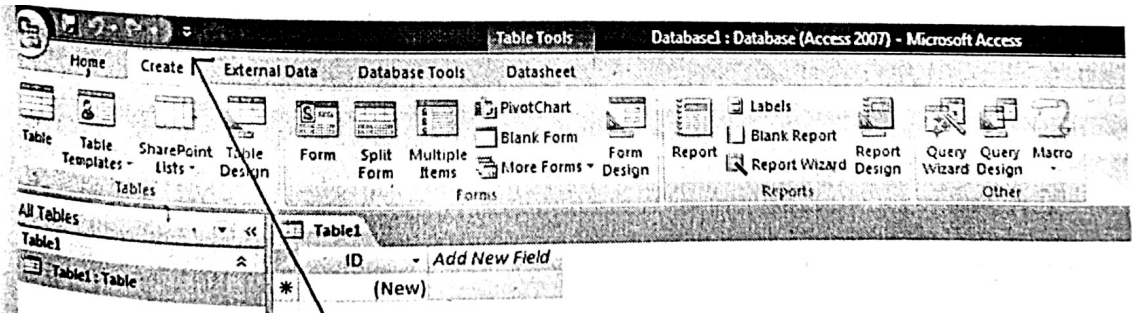
Begin by opening our existing database file if it is not already open. Do this by clicking on the Access desktop icon to bring up the getting started screen again. We should see the file name that we just created towards the top of the right hand side bar (If wecan'tsee it, click the folder icon to browse for the file). Click on the file name to bring up our blank database.

If, at this stage, we get a security warning underneath the ribbon, click where it says options, select the "enable this content" radio button (in the pop-up window), and then click ok.The blank database file should now be fully open.



2. Create Access Table

Select the 'CREATE tab' on the Access ribbon. Next select the 'TABLE DESIGN icon' from the TABLES group. This creates a new table.



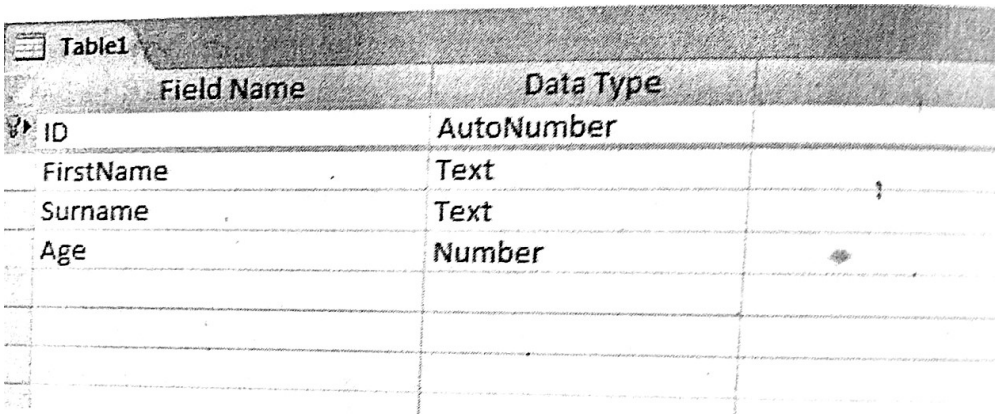
3. Create fields in DESIGN VIEW

This brings up the 'TABLE DESIGN GRID' where we enter each field name and its data type. The first field we are going to create is the ID field which is going to contain a unique reference number for each record. Enter the name "ID" into the first column of the first row in the grid. Because we want Access to automatically generate a unique reference number, select

AUTONUMBER from the drop down list in the data type column. We can also enter a description for each field, but this is not essential.

On the next row, the field is going to be called FIRSTNAME and the data type is going to be TEXT. On the third row, the field name is SURNAME with the data type again being TEXT. And finally, the last field name is AGE and the data type here is going to be NUMBER.

Before we save the table we will need to choose the Primary Key, which in this case is ID. To do this, select the ID row by clicking on it, and then simply click the primary key symbol on the Ribbon.



We can now save the table by clicking the save icon on the top left of the screenabove the Access Ribbon. To view the table, select 'DATASHEET VIEW' from the VIEWSgroup under the 'DESIGN TAB'. This brings up the datasheet view of the table that we have just created. We should see the field headings running across the top of the table.

4. Data entry in DATASHEET VIEW

Although entering data onto the datasheet table is not a part of the design process, the table now exists as a database object, and we can test it by entering some information into the cells.

Select the first cell in the FIRSTNAME column and enter the name Amit. For SURNAME enter Sharma, and for AGE enter 20.

NB just ignore the column underneath the heading ADD NEW FIELD. We created all the fields we needed in design view.

We have now entered the first record in the table - record 1 for AmitSharma aged 20. We can now press the return key and the record will save automatically. We are now ready to enter the second record on the next row -RohitPathak 25. Press return and then fill in the data for the last record we are going to do for now -ArunVerma 22.

Our datasheet table should now look like this:

Table1

ID	FirstName	Surname	Age	Add New Field
1	Amit	Sharma	20	
2	.Rohit	Pathak	25	
3	Arun	Verma	22	
(New)				

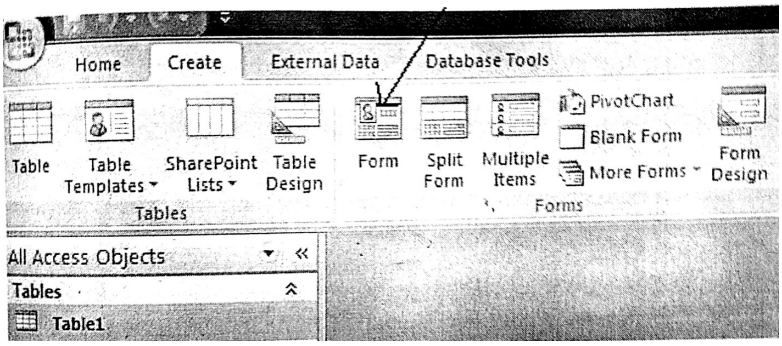
Once we have an Access Table populated with data, we have the option to work with it right away. Indeed, some people may not go on to create forms, queries, and reports once they have created their first table. For example, store, edit, filter, search, and calculate a set of database records. Now that we have created our first table and stored a number of records we, will take a look how you go about working with tables. We shall use your newly created table with some additional records to illustrate how to do this.

6.5 FORMS

- It is easier and quicker to enter data onto a form than it is into a table, not least because wehave more control over the layout and labeling of our form.
- The following are the steps to learn how to create a FORM that uses newly created table as the DATA SOURCE.

Create Access form

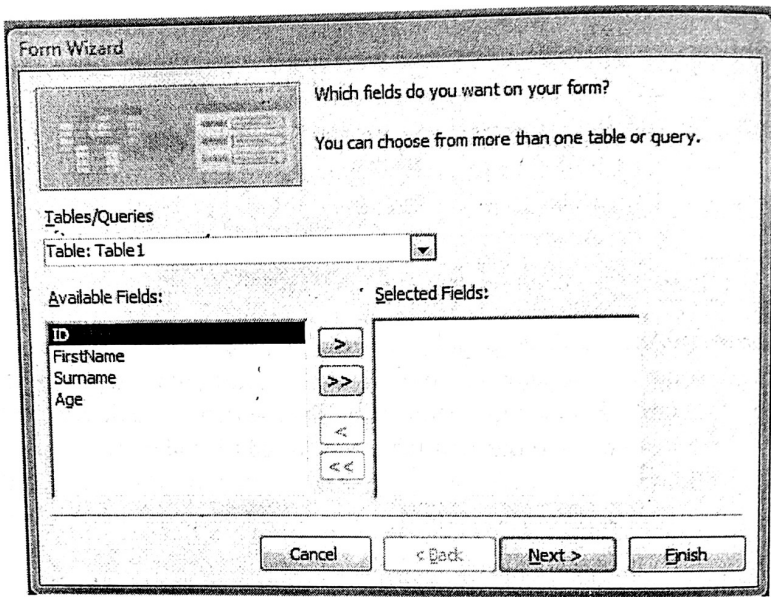
There are a number of ways to create an Access form. We are going to use the ACCESS FORM WIZARD.



A drop down list of form types appears. We just need to select FORM WIZARD from thislist.

1. Select table and fields

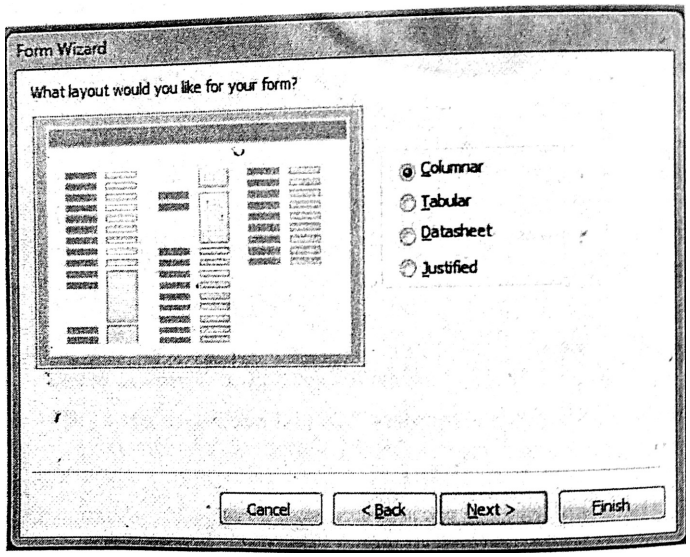
The Form wizard then opens as the pop up window below.



On the first page of the form wizard, we select which table we are going to use as the DATA SOURCE. Since there is only one table in this database so far, it should already be selected for us in the TABLES/QUERIES combo box.

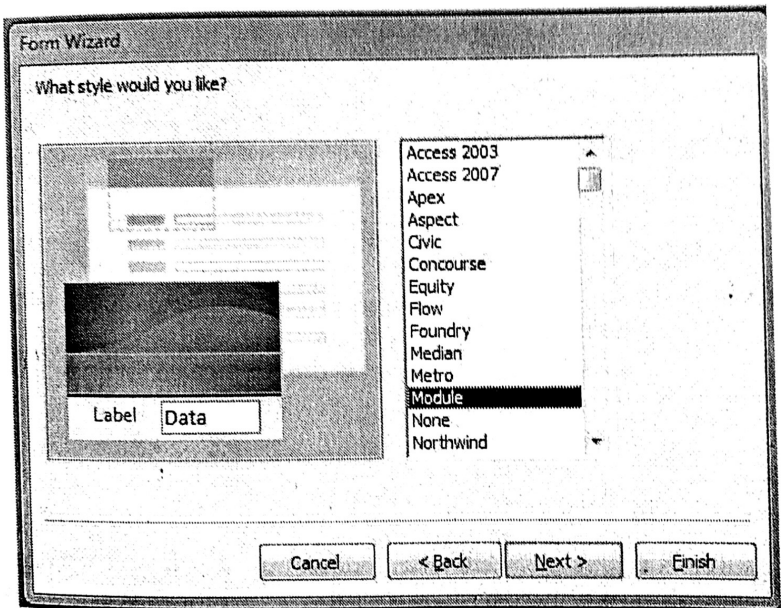
There is a list of the four fields in the table. We could select each field individually by highlighting the field name and clicking the single arrow>. We can select all four just by clicking the double arrow >>. Once they have been transferred into the right hand box they are selected.We can now click NEXT to .go to page 2 of the form wizard.

2. Form Layout



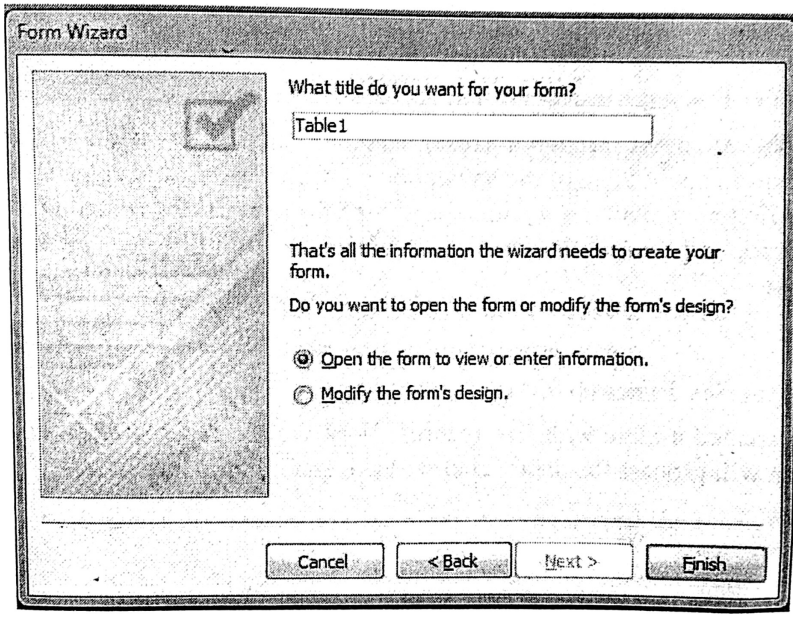
This is where we choose the FORM LAYOUT. We are going to leave this on the default setting ofCOLUMNAR. Click NEXT again to move onto page three of the form wizard.

3. Form Style



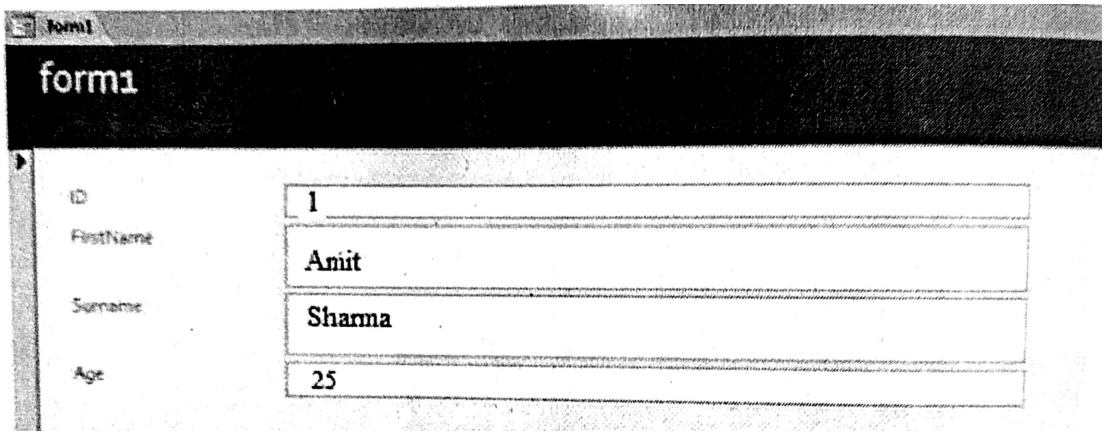
Here we select a style that from the list box. Preview a style by clicking on its name. This step just determines how the form looks aesthetically. Once we have selected a style to your taste,click NEXT again.

4. Form Title



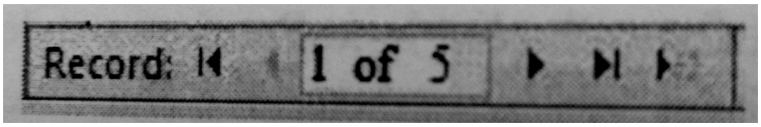
This is the last stage of the wizard. Here we are going to choose the name for our form: we can say FORM1. We are going to leave the radio button selected for OPEN THE FORM TO ENTER OR VIEW INFORMATION

All we need to do now is click FINISH and our form should like something like this:



5. Navigate Records via Access Form

The form opens with the first record in our original table displayed. We can navigate through each record using the NAVIGATION BUTTON at the bottom left of the form window.



To move to the next record, click the right arrow button on the navigation control. To move to the last record, click the >| button on the navigation control. To move backwards, through the recordset use the left arrow buttons. We could edit any record from the form, which will update the data in our table.

6. Create New Record via Access Form

We know how to move through the recordset, let's finish off by adding two more records. Click the star button on the right of the navigation control. This should bring up a blank record. Add Rahul for firstname, Kumar for surname, and 12 for age. Click the star again to bring up a new blank record, and enter Rahul Kumar 12. When we have finished, click x,on the form window to close.

6.6 QUERY

1. Access Queries Process Data

We have created a table with five records. Now, we are going to create an Access 2007 QUERY, which will process the data which we have entered into our table.

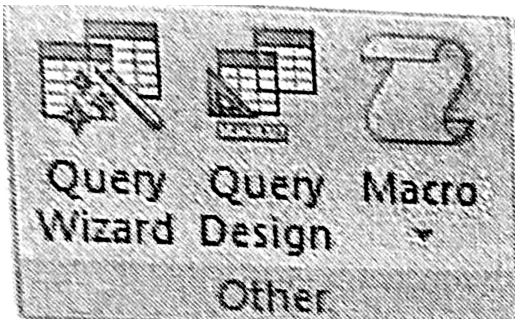
ID Number	FirstName	Surname	Age
1	Amit	Sharma	20
2	Rohit	Pathak	15
3	Arun	Verma	19
4	Rahul	Kumar	12
5	Kiran	Bedi	30
6	Akant	Rai	14
7	Tajinder	Pal	18

8	Tarun	Verma	15

Now we want Access to do is extract all the records in the above table, where the persons age is greater than or equal to 20. Access Queries can do far more than this, of course, but this simple task should serve as a useful learning exercise.

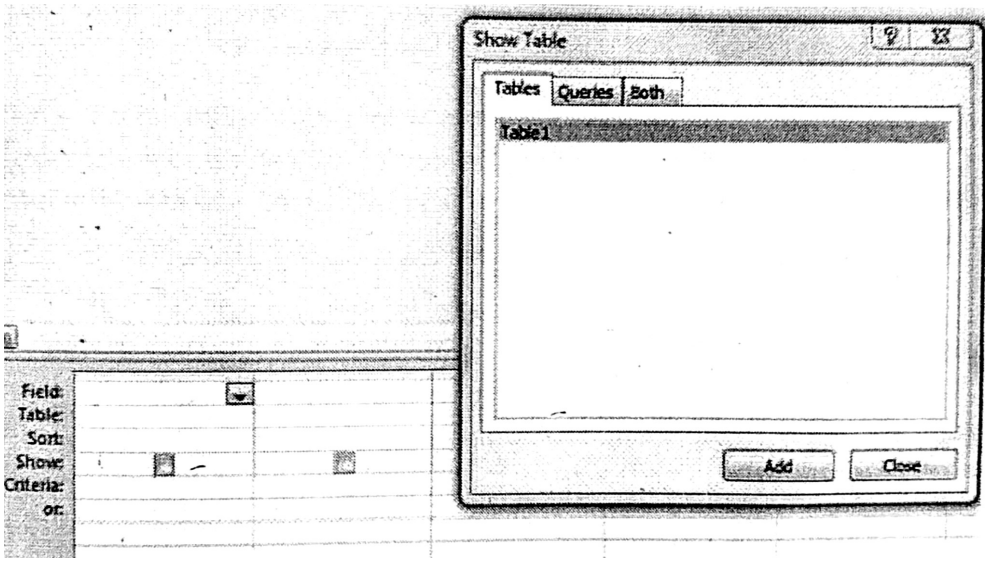
2. Open the Query Design Grid

The first stage is to select the CREATE TAB and then go to the OTHER group on the far right of the ribbon. Then click on the QUERY DESIGN ICON to bring up the query design grid.



3. Select Table for Query

The next stage is to select which table we are going to use in this query. When the query design grid opened, the SHOW TABLE pop up window should also have opened. If necessary we can open this window manually by clicking the SHOW TABLE icon in the QUERY SET UP group on the Access Ribbon.

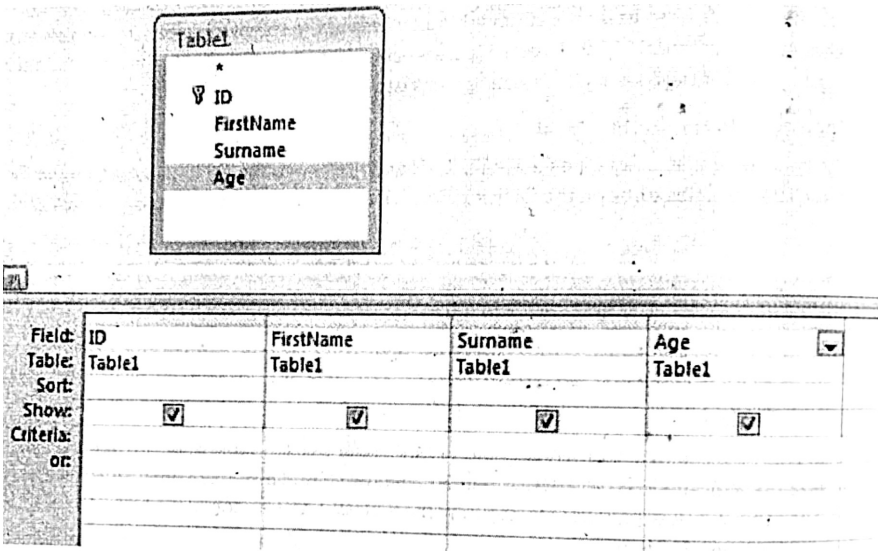


So far there is only one table in our database TABLE 1. This should be highlighted when the window opens, but if not, just click on it once. Next click ADD. When we close the pop-up window by clicking x, we should see a box labeled TABLE 1 above the query design grid.

4. Select fields from Table

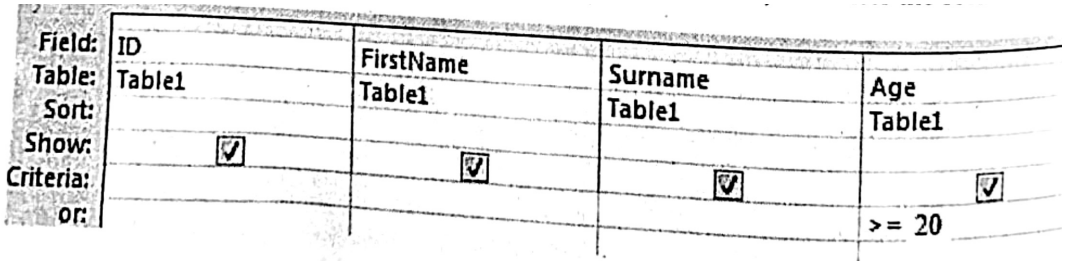
In the table 1 box, we will see a list of its field names. We are going to use all the fields in this query, so select each one individually by double clicking on their names. We will now see

the field names at the top of each column in the Query Design Grid.



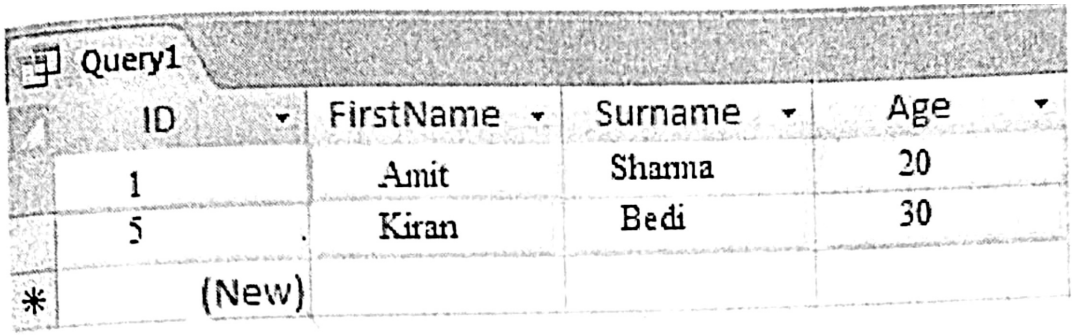
5. Enter query criteria

The purpose of our query is to extract records where the person'sage is greater than or equal to 20. To do this we enter the criteria into the appropriate cell of the query design grid. In this case you need to go to the AGE column of the CRITERIA row, and enter the formula >=20.



6. Save and Run Query

Click the save Icon above the Access Ribbon. When prompted for the query name, just use the default QUERY1. To run the query, click the RUN icon in the RESULTS GROUP of the Access Ribbon. We should now be presented with a datasheet displaying your query results - Amit Sharma 20 and KiranBedi 30.



6.7 REPORTS

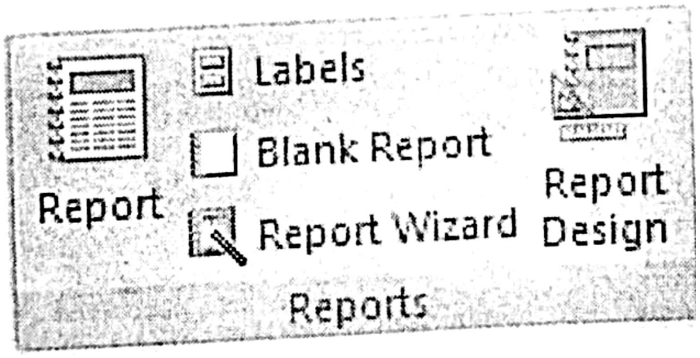
Access REPORTS are a way of displaying and printed information from your

database.Developers often use Reports to display the results of a Query, which is what we are going to do next.

Like other Access database objects, there a number-of ways to create an Access REPORT. We are going use the REPORT WIZARD. Our task is to display the results of QUERY1 which we created in the previous query.

1. Open Report Wizard

Begin by selecting the CREATE TAB on the Access Ribbon. Then click REPORT WIZARD from the REPORTS group to open the pop up window.

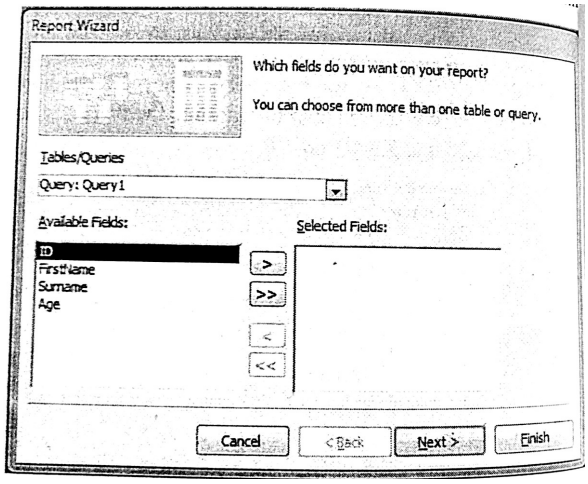


2. Select Report Data Source and Fields

The first page of the Report Wizard is almost identical to that of the Form Wizard. On this page, we will select QUERY1 as the DATA as the DATA SOURCE for the Report. This is done by selecting the query from the list in the TABLES/QUERIES combo box. Then select all fields from QUERY1 by clicking the double arrow to the right of the text box labeled AVAILABLE FIELDS. The field names will now appear in the SELECTED FIELDS text box on the right. Click next to go to the second page of the wizard.

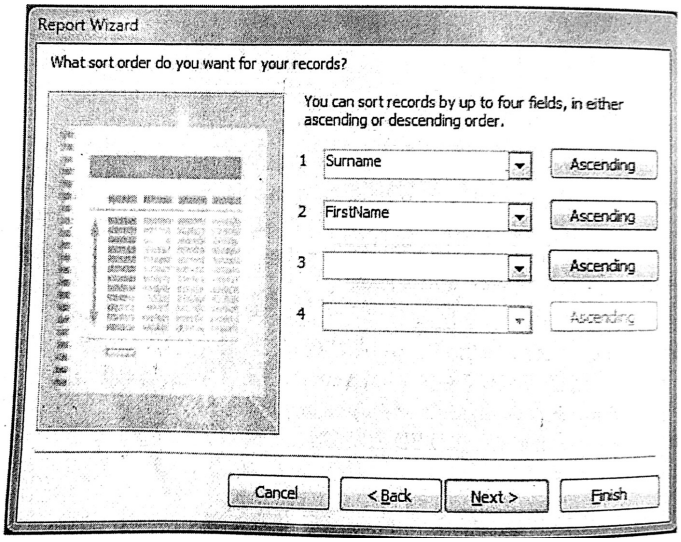
3. Report Grouping Levels

Page two of the wizard is where we can add grouping levels if required. So we could, for example, group each person in our report with others of the same age. We don't need to do this for this particular example, but it is certainly a useful feature that we might want to learn more about later. But for now, just click next.



4. Record Sort Order

Page three of the wizard is where we determine what order the records are displayed in the report. For example: we are going to put our records into alphabetical order for SURNAME and FIRSTNAME. Begin by clicking the top drop down box and selecting SURNAME, and in the box 2below select FIRSTNAME. This means that records are sorted alphabetically by surname, and if two surnames are the same, the FIRSTNAME is used to determine which record comes first etc. and Click Next.

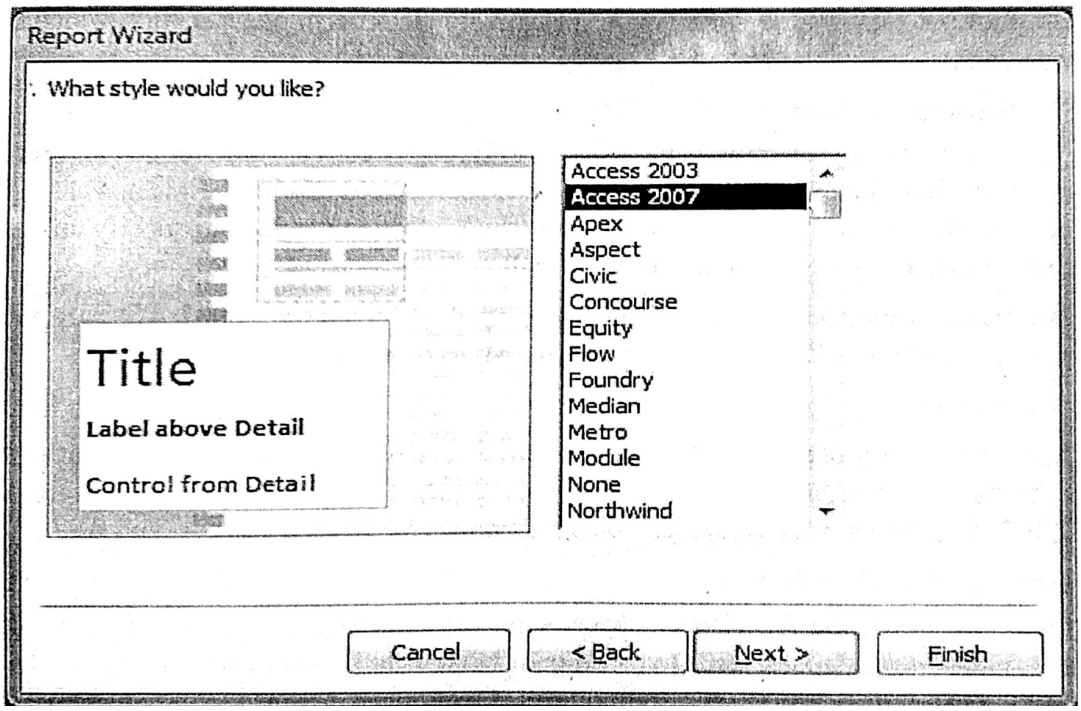


5. Report Lay Out

This page of the Wizard is where you set the lay out for your report. There are various lay out options, but in our case we shall keep to the default setting of TABULAR and Click next.

6. Style

For style, select Access 2007 from the list and click next.



7. Name of Report

Name our report REPORT1 and then click FINISH. We access report will now be displayed on the screen.

6.8 SECURITY

Microsoft Access offers the following ways to secure the application while allowing users to remain productive:

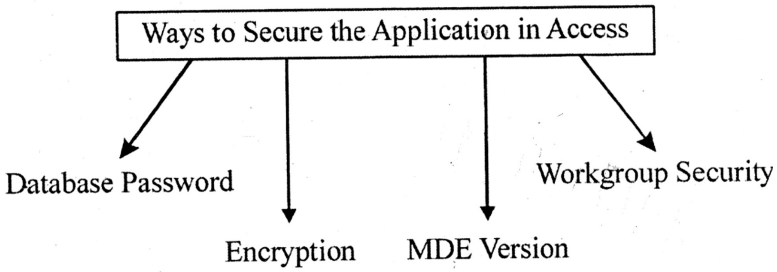


Fig. 6.2: Ways to Secure the Application in Access

- Database Password
- Workgroup Security
- Encryption
- MDE Version

(a) Database Password:

The most basic security method is a database password. Once entered, the user has full control of all the database objects. This is a relatively weak form of protection which can be easily cracked.

(b) Workgroup Security

The next level of protection is the use of workgroup, security which requires a user name and password. Individual users and groups can be specified along with their rights at the object type or individual object level. This is especially helpful when we need to specify people with read only or data entry rights but this may be challenging to specify.

Note: The workgroup security is not supported in the Access 2007 ACCDB database format, however Access 2007 still supports it for MDB databases.

(c) Encryption

Encryption is a process to translate confidential data into unreadable form so that unwanted access to the information is avoided. Access databases can be encrypted. The Access 2007 ACCDB format offers advanced encryption techniques.

(d) Access MDE Version

In Access 2007, the ACCDB database is converted to an ACCDE file. In case of MDE version, the database design needs to be secured in order to prevent changes, the database can be locked/protected and the source code compiled by converting the database to an MDE file.

Questions

1. What is the need of MS Access? Write note on different components.
2. List the name of MS Access components.
3. Write steps to create database.
4. Write steps to create tables in a database.
5. Write steps to design user friendly form for entering inputs into the database.
6. Write customized reports to show data from database.
7. Write steps to use macro.
8. Write note on security tool available in MS Access 2007