1. **Introductory concepts of DBMS**
2. [What are the disadvantages of File processing?](http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/461-problems-with-file-system.html)

**Data redundancy and inconsistency**

Multiple file formats, duplication of information in different files

**Difficulty in accessing data**

Need to write a new program to carry out each new task

**Data isolation**

Multiple files and formats

**Integrity problems**

Integrity constraints (e.g., account balance > 0) become “buried” in

program code rather than being stated explicitly

Hard to add new constraints or change existing ones

**Atomicity of updates**

es may leave database in an inconsistent state with partial updates

carried out

Example: Transfer of funds from one account to another should either

complete or not happen at all

**Concurrent access by multiple users**

Concurrent access needed for performance

Uncontrolled concurrent accesses can lead to inconsistencies

Example: Two people reading a balance (say 100) and updating it by

withdrawing money (say 50 each) at the same time

**Security problems**

Hard to provide user access to some, but not all, data

**Advantages of Database Management System (DBMS)**

**1. Improved data sharing**

An advantage of the database management approach is, the DBMS helps to create an environment in which end users have better access to more and better-managed data.

Such access makes it possible for end users to respond quickly to changes in their environment.

**2. Improved data security**

The more users access the data, the greater the risks of data security breaches. Corporations invest considerable amounts of time, effort, and money to ensure that corporate data are used properly. A DBMS provides a framework for better enforcement of data privacy and security policies.

**3. Better data integration**

Wider access to well-managed data promotes an integrated view of the organization’s operations and a clearer view of the big picture. It becomes much easier to see how actions in one segment of the company affect other segments.

**4. Minimized data inconsistency**

Data inconsistency exists when different versions of the same data appear in different places. For example, data inconsistency exists when a company’s sales department stores a sales representative’s name as “Bill Brown” and the company’s personnel department stores that same person’s name as “William G. Brown,” or when the company’s regional sales office shows the price of a product as $45.95 and its national sales office shows the same product’s price as $43.95. The probability of data inconsistency is greatly reduced in a properly designed database.

**5. Improved data access**

 The DBMS makes it possible to produce quick answers to ad hoc queries. From a database perspective, a query is a specific request issued to the DBMS for data manipulation—for example, to read or update the data. Simply put, a query is a question, and an ad hoc query is a spur-of-the-moment question. The DBMS sends back an answer (called the query result set) to the application. For example, end users, when dealing with large amounts of sales data, might want quick answers to questions (ad hoc queries) such as:

- What was the dollar volume of sales by product during the past six months?  
- What is the sales bonus figure for each of our salespeople during the past three months?  
- How many of our customers have credit balances of 3,000 or more?

**6. Improved decision making**

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.

**7. Increased 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.

Till now we have seen different benefits of database management systems. But it has certain limitations or disadvantages.

Let's find various disadvantages of database system.

**Disadvantages of Database Management System (DBMS):**

 Although the database system yields considerable advantages over previous data management approaches, database systems do carry significant disadvantages. For example:

**1. Increased costs**

one of the disadvantages of dbms is Database systems require sophisticated hardware and software and highly skilled personnel. The cost of maintaining the hardware, software, and personnel required to operate and manage a database system can be substantial. Training, licensing, and regulation compliance costs are often overlooked when database systems are implemented.

**2. Management complexity**

Database systems interface with many different technologies and have a significant impact on a company’s resources and culture. The changes introduced by the adoption of a database system must be properly managed to ensure that they help advance the company’s objectives. Given the fact that database systems hold crucial company data that are accessed from multiple sources, security issues must be assessed constantly.

**3. Maintaining currency**

To maximize the efficiency of the database system, you must keep your system current. Therefore, you must perform frequent updates and apply the latest patches and security measures to all components.

Because database technology advances rapidly, personnel training costs tend to be significant. Vendor dependence. Given the heavy investment in technology and personnel training, companies might be reluctant to change database vendors.

As a consequence, vendors are less likely to offer pricing point advantages to existing customers, and those customers might be limited in their choice of database system components.

**4. Frequent upgrade/replacement cycles**

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.

**Levels of Abstraction**

**Physical level**: describes how a record (e.g., instructor) is stored.

**Logical level**: describes data stored in database, and the relationships among

the data.

type instructor = record

ID : string;

name : string;

dept\_name : string;

salary : integer;

end;

**View level**: application programs hide details of data types. Views can also

hide information (such as an employee’s salary) for security purposes.

**View of Data**

An architecture for a database system

**Instances and Schemas**

Instances and Schemas: A collection of information which is stored in the DB.

DB change over a time as information is inserted and deleted.

Collection of information store in the database at a particular moment is known as

instances of DB.

Overall design is known as DB Schema.

**Instances and Schemas**

**Logical Schema** – the overall logical structure of the database

Example: The database consists of information about a set of customers and

accounts in a bank and the relationship between them

Analogous to type information of a variable in a program

**Physical schema**– the overall physical structure of the database

**Instance** – the actual content of the database at a particular point in time

Analogous to the value of a variable

**Physical Data Independence** – the ability to modify the physical schema without

changing the logical schema

**Applications depend on the logical schema**

In general, the interfaces between the various levels and components should

be well defined so that changes in some parts do not seriously influence

others.

**Types of Data Base**

**1) Classification Based on the number of user**

Single user

Multiple work group DB

**2) Classification Based on location**

Centralized DB

Distributed DB

**3) Classification Based on how they use**

Operation DB

Transaction DB

**4) Degree to which Data are structured**

Unstructured data

Semi-structure Data

Structure Data

**5) DB System User**

Naive user

End User

System Analyst & Application Programmer

DBA

**Relational Database Model in DBMS**

The relational model was introduced in 1970 by E. F. Codd. The relational model foundation is a mathematical concept known as a relation.

The relational model is implemented through a very sophisticated relational database management system (RDBMS).

The RDBMS performs the same basic functions provided by the hierarchical and network DBMS systems, in addition to a host of other functions that make the relational data model easier to understand and implement.

Arguably the most important advantage of the RDBMS is its ability to hide the complexities of the relational model from the user. The RDBMS manages all of the physical details, while the user sees the relational database as a collection of tables in which data are stored. The user can manipulate and query the data in a way that seems intuitive and logical.

A relational table stores a collection of related entities. In this respect, the relational database table resembles a file. But there is one crucial difference between a table and a file:

A table yields complete data and structural independence because it is a purely logical structure. How the data are physically stored in the database is of no concern to the user or the designer.

Another reason for the relational data model’s rise to dominance is its powerful and flexible query language.

For most relational database software, the query language is Structured Query Language (SQL), which allows the user to specify what must be done without specifying how it must be done.

The RDBMS uses SQL to translate user queries into instructions for retrieving the requested data. SQL makes it possible to retrieve data with far less effort than any other database or file environment.

From an end-user perspective, any SQL-based relational database application involves three parts: a user interface, a set of tables stored in the database, and the SQL “engine.” Each of these parts is explained below.  
  
**The end-user interface:**  
  
Basically, the interface allows the end user to interact with the data (by auto-generating SQL code).

Each interface is a product of the software vendor’s idea of meaningful interaction with the data.

You can also design your own customized interface with the help of application generators that are now standard fare in the database software  arena.  
  
**A collection of tables stored in the database:**  
  
In a relational database, all data are perceived to be stored in tables.

The tables simply “present” the data to the end user in a way that is easy to understand.

Each table is independent. Rows in different tables are related by common values in common attributes.

**SQL engine:**  
  
Largely hidden from the end user, the SQL engine executes all queries, or data requests. Keep in mind that the SQL engine is part of the DBMS software.

The end user uses SQL to create table structures and to perform data access and table maintenance.

The SQL engine processes all user requests—largely behind the scenes and without the end user’s knowledge.

of Relational Model in DBMS

The different advantages of Relational model are as follows:

1. Structural independence

2. Improved conceptual simplicity

3. Easier database design, implementation, management, and user

4. Ad hoc query capability (SQL)

5. Powerful database management system

## Disadvantages of Relational Model in DBMS

The different disadvantages of Relational model are as follows:

1. Substantial hardware and system software overhead

2. Possibility of poor design and implementation

3. Potential “islands of information” problems

**The Components of Entity Relationship Model in DBMS:**

**Entity:**

An entity was defined as anything about which data are to be collected and stored.

An entity is represented in the ERD by a rectangle, also known as an entity box.

The name of the entity, a noun, is written in the center of the rectangle.

The entity name is generally written in capital letters and is written in the singular form: PAINTER rather than PAINTERS, and EMPLOYEE rather than EMPLOYEES.

**Relationships:**

Relationships describe associations among data. Most relationships describe associations between two entities. when the basic data model components were introduced, three types of relationships among data were illustrated:

one-to-many (1:M),

many-to-many (M:N),

and one-to-one (1:1).

The ER model uses the term connectivity to label the relationship types.

The name of the relationship is usually an active or passive verb.

For example, a PAINTER paints many PAINTINGs; an EMPLOYEE learns many SKILLs; an EMPLOYEE manages a STORE.

The following figure the different types of relationships using two ER notations:

The original Chen notation and the more current Crow’s Foot notation.

The left side of the ER diagram shows the Chen notation, based on Peter chen’s landmark paper.

In this notation, the connectivities are written next to each entity box.

Relationships are represented by a diamond connected to the related entities through a relationship line.

The relationship name is written inside the diamond

## The advantages of ER model in DBMS

1. Exceptional conceptual simplicity  
2. Visual representation  
3. Effective communication tool  
4. Integrated with the relational database model

## The disadvantages of ER Model in DBMS

1. Limited constraint representation  
2. Limited relationship representation  
3. No data manipulation language  
4. Loss of information content

[Explain different types of keys available in Relational Model.](http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/476-types-of-keys-in-dbms.html)

**Types of Keys in DBMS**

In the [Relational Database Model](http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/470-relational-database-model.html), keys are important because they are used to ensure that each row in a table is uniquely identifiable. They are also used to establish relationships among tables and to ensure the integrity of the data.

**Functional dependence:**  
  
The attribute B is functionally dependent on the attribute A if each value in column A determines one and only one value in column B.  
  
**Composite key:**

A key may be composed of more than one attribute. Such a multi attribute key is known as a composite key.  
  
**Relational Database Keys:**

**Superkey:** An attribute (or a combination of attributes) that uniquely identifies each row in a table.  
  
**Candidate key:** A minimal (irreducible) superkey. A superkey that does not contain a subset of attributes that is itself a superkey.  
  
**Primary key:** A candidate key selected to uniquely identify all other attribute values in any given row. Cannot contain null entries.

**Secondary key:** An attribute (or combination of attributes) used strictly for data retrieval purposes.  
  
**Foreign key:** An attribute (or combination of attributes) in one table whose values must either match the primary key in another table or be null.

[Explain Database Integrity rules.](http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/477-integrity-rules-in-dbms.html)

1. **Entity-Relationship model**