# Data Base Management system (DBMS)

## CHAPTER 1

## 1. Introduction of Data, Database, Database system and DBMS

**Data** can be defined as a representation of facts, concepts, or instructions in a formalized manner, which should be suitable for communication, interpretation, or processing by human or electronic machine.

Data is represented with the help of characters such as alphabets (A-Z, a-z), digits (0-9) or special characters (+,-, /,*, <,>, = etc.)

**Information** is organized or classified data, which has some meaningful values for the receiver. Information is the processed data on which decisions and actions are based.

**Database: -** A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.

Let us discuss a database example: An online telephone directory uses a database to store data of people, phone numbers, and other contact details. Your electricity service provider uses a database to manage billing, client-related issues, handle fault data, etc.

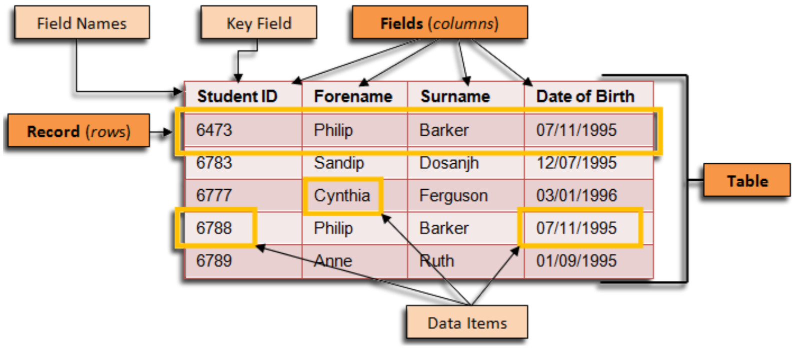
Let us also consider Facebook. It needs to store, manipulate, and present data related to members, their friends, member activities, messages, advertisements, and a lot more. We can provide a countless number of examples for the usage of databases.

**Database** is a collection of interrelated data which helps in the efficient retrieval, insertion, and deletion of data from the database and organizes the data in the form of tables, views, schemas, reports, etc. For Example, a university database organizes the data about students, faculty, admin staff, etc. which helps in the efficient retrieval, insertion, and deletion of data from it.

**Record:** Record is a collection of values or fields of a specific entity. Eg. An employee, Salary account, etc.

**Field:** A field refers to an area within a record which is reserved for a specific piece of data. Eg. Employee ID.

**Table:** Table is the collection of records of specific types. E.g. Employee table is a collection of record related to all the employees.



| Student ID | Forename | Surname | Date of Birth |
|---|---|---|---|
| 6473 | Philip | Barker | 07/11/1995 |
| 6783 | Sandip | Dosanjh | 12/07/1995 |
| 6777 | Cynthia | Ferguson | 03/01/1996 |
| 6788 | Philip | Barker | 07/11/1995 |
| 6789 | Anne | Ruth | 01/09/1995 |

*fig: Showing Fields , record and table*

## 2. Keys In Database

KEYS in DBMS is an attribute or set of attributes which helps you to identify a row (tuple) in a relation (table). They allow you to find the relation between two tables. Keys help you uniquely identify a row in a table by a combination of one or more columns in that table. Key is also helpful for finding unique record or row from the table. Database key is also helpful for finding unique record or row from the table.

- Super Key – A super key is a group of single or multiple keys which identifies rows in a table.
- Primary Key – is a column or group of columns in a table that uniquely identify every row in that table.
- Candidate Key – is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes.
- Alternate Key – is a column or group of columns in a table that uniquely identify every row in that table.
- Foreign Key – is a column that creates a relationship between two tables. The purpose of foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.
- Compound Key – has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database.
- Composite Key – is a combination of two or more columns that uniquely identify rows in a table. The combination of columns guarantees uniqueness, though individual uniqueness is not guaranteed.
- Surrogate Key – An artificial key which aims to uniquely identify each record is called a surrogate key. These kind of key are unique because they are created when you don't have any natural primary key.

## 3. Advantages of Database Management System

In contrast with the File Based Data Management System, Dbms has numerous benefits. We are putting light on some of the considerable benefits here–

**1. Data Integrity**

Data integrity means data is consistent and accurate in the database. It is essential as there are multiple databases in DBMS. All these databases contain data which is visible to multiple users. Therefore, it is essential to ensure that data is consistent and correct in all databases for all users.

**2. Data Security**

Data security is a vital concept in a database. Only users authorized must be allowed to access the database and their identity must be authenticated using username and password. Unauthorized users shouldn't be allowed to access the database under any circumstances as it violets the integrity constraints.

A DBMS provides a better platform for data privacy thus helping companies to offer an improved data security.

**3. Better data integration**

Due to the database management system, we have access to well managed and synchronized form of data making it easy to handle. It also gives an integrated view of how a particular organization is working and keeps track of how one segment of the company affects another segment.

**4. Minimized Data Inconsistency**

Data inconsistency occurs between files when various versions of the same data appear in different places. Data consistency is ensured in the database; there is no data redundancy. Besides, any database changes are immediately reflected by all users, and there is no data inconsistency.

**5. Faster Data Access**

The database management system helps the users to produce quick answers to queries making data accessing accurate and faster.

**6. Recovery and Backup**

DBMS automatically takes care of recovery and backup. The users are not required to take periodical backup as this is taken care of by DBMS. Besides, it also restores a database after a system failure or crash to prevent its previous condition.

**7. Increased end-user productivity**

The available data transform into helpful information with the help of combination tools. It helps end users make better, informative and quick decisions that can make the difference between success and failure in the global economy.

## 4. DDL (Data Definition Language):

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database. DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.

List of DDL commands:

**CREATE:** This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).

***CREATE TABLE [table name] ([column definitions]) [table parameters];***

*For example:*

*CREATE TABLE Employee (Employee Id INTEGER PRIMARY KEY, First name CHAR (50) NULL, Last name CHAR (75) NOT NULL);*

**DROP:** This command is used to delete objects from the database.

*Drop statement syntax is:*

*DROP object type object name;*

*For example:*

*DROP TABLE Employee;*

*In this example, we're deleting the Employee table.*

**ALTER:** This is used to alter the structure of the database.

*An alter command syntax is:*

*ALTER object type object name parameters;*

For example:

ALTER TABLE Employee ADD PRIMARY KEY (employee_pk);

In this example, we added a unique primary key to the table to add a constraint and enforce a unique value. The constraint "employee_pk" is a primary key and is on the Employee table.

**TRUNCATE:** This is used to remove all records from a table, including all spaces allocated for the records are removed.

*Truncate statement syntax is:*

*TRUNCATE TABLE table_name;*

*For example:*

*TRUNCATE TABLE Employee;*

*In this example, we're marking all the extents of the Employee table for deallocation, so they're considered empty for reuse.*

**COMMENT:** This is used to add comments to the data dictionary.

**RENAME:** This is used to rename an object existing in the database.

*RENAME TABLE old_table_name TO new_table_name;*

## 5. DML (Data Manipulation Language):

The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

List of DML commands:

**INSERT :** It is used to insert data into a table.

Syntax:

INSERT INTO NAME_OF_TABLE (1_column, 2_column, 3_column, …. N_column)

VALUES (1_value, 2_value, 3_value, …. N_value);

Or

INSERT INTO NAME_OF_TABLE

VALUES (1_value, 2_value, 3_value, …. N_value);

**Example:**

INSERT INTO Student(Stu_Name, DOB, Phone, Mail)

VALUES(‘Phoebe’, ‘1998-05-26’, 7812865845, ‘user@xyz.com’);

**UPDATE:** It is used to update existing data within a table.

Syntax:

UPDATE name_of_table SET 1_coumn = 1_value, 2_coumn = 2_value, 3_coumn = 3_value, … , N_coumn = N_value

WHERE condition;

And here,

name_of_table: name of the table

1_column, 2_column, 3_column, …. N_column: name of the first, second, third, …. nth column.

1_value, 2_value, 3_value, …. N_value: the new value for the first, second, third, …. nth column.

condition: the condition used to select those rows for which the column values need to be updated.

**Example:**

UPDATE Student SET Phone = 9039462901 WHERE Stu_Name = ’Phoebe’;

The WHERE clause in the preceding query is used to select the rows for which the columns need to be adjusted, and the SET statement has been used to assign new values to a particular column. If the WHERE clause is not used at all, then all of the rows’ columns will be modified. As a result, the WHERE clause is used to pick specific rows from the table.

Thus, the example query would update the phone number of the student with the name ‘Phoebe’.

**DELETE :** It is used to delete records from a database table.

Syntax:

DELETE FROM name_of_table [WHERE condition];

**Example:**

DELETE FROM Student WHERE Stu_Name = ’Phoebe’;

The command given above would delete the record for the student with the name ‘Phoebe’ from the ‘Student’ table. Apart from this, one can also use the LOCK Table statement to explicitly acquire the shared or exclusive table lock on a specified table.

**SELECT**: It is used to select data from a database.

Syntax:

SELECT column1, column2, ...
FROM table_name;

**Example:**

SELECT CustomerName,Address FROM Customers;

## 6. Difference between DDL and DML.

| DDL | DML |
|---|---|
| It stands for Data Definition Language. | It stands for Data Manipulation Language. |
| It is used to create database schema and can be used to define some constraints as well. | It is used to add, retrieve or update the data. |
| It basically defines the column (Attributes) of the table. | It add or update the row of the table. These rows are called as tuple. |
| It doesn’t have any further classification. | It is further classified into Procedural and Non-Procedural DML. |
| Basic command present in DDL are CREATE, DROP, RENAME, ALTER etc. | BASIC command present in DML are UPDATE, INSERT, MERGE etc. |
| DDL does not use WHERE clause in its statement. | While DML uses WHERE clause in its statement. |

## 7. Types of Database Model

**Database Model:** It determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized and manipulated.

There are four common types of database model that are useful for different types of data or information. Depending upon your specific needs, one of these models can be used.

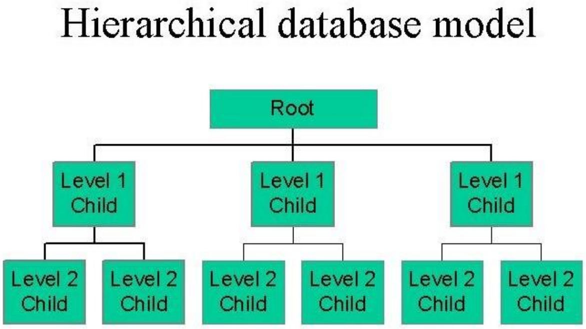
1. Hierarchical databases.
2. Network databases.
3. Relational databases.
4. Object-oriented databases

### 1. Hierarchical databases

It is one of the oldest database model developed by IBM for information Management System. In a hierarchical database model, the data is organized into a tree-like structure. In simple language we can say that it is a set of organized data in tree structure.

This type of Database model is rarely used nowadays. Its structure is like a tree with nodes representing records and branches representing fields. The windows registry used in Windows XP is an example of a hierarchical database. Configuration settings are stored as tree structures with nodes.

The following figure shows the generalized the structure of Hierarchical database model in which data is stored in the form of tree like structure (data represented or stored in root node, parent node and child node).



**Advantage**

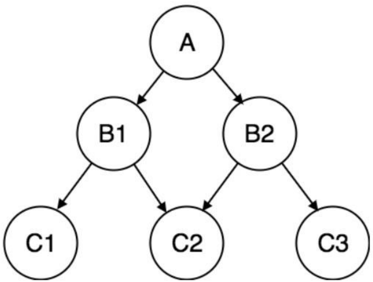
- The model allows us easy addition and deletion of new information.
- Data at the top of the Hierarchy is very fast to access.
- It worked well with linear data storage mediums such as tapes.
- It relates well to anything that works through a one to many relationships.

**Disadvantages**

- It requires data to be repetitively stored in many different entities.
- Now a day there is no longer use of linear data storage mediums such as tapes.
- Searching for data requires the DBMS to run through the entire model from top to bottom until the required information is found, making queries very slow.
- This model support only one to many relationships, many to many relationships are not supported.

### 2. Network databases

This is looks like a Hierarchical database model due to which many time it is called as modified version of Hierarchical database. Network database model organized data more like a graph and can have more than one parent node. The network model is a database model conceived as a flexible way of representing objects and their relationships.



**Advantage**

• The network model is conceptually simple and easy to design.
• The network model can represent redundancy in data more effectively than in the hierarchical model.
• The network model can handle the one to many and many to many relationships which is real help in modelling the real-life situations.
• The data access is easier and flexible than the hierarchical model.
• The network model is better than the hierarchical model in isolating the programs from the complex physical storage details.

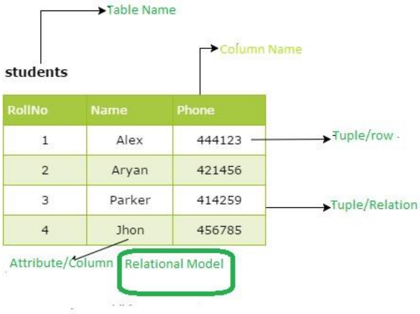
**Disadvantage:**

• All the records are maintained using pointers and hence the whole database structure becomes very complex.
• The insertion, deletion and updating operations of any record require the large number of pointers adjustments.
• The structural changes to the database is very difficult.

## 3. Relational Database

A relational database is developed by E. F. Codd in 1970. The various software systems used to maintain relational databases are known as a relational database management system (RDBMS). In this model, data is organized in rows and column structure i.e., two-dimensional tables and the relationship is maintained by storing a common field. It consists of three major components.

In relational model, three key terms are heavily used such as relations, attributes, and domains. A relation nothing but is a table with rows and columns. The named columns of the relation are called as attributes, and finally the domain is nothing but the set of values the attributes can take. The following figure gives us the overview of rational database model.



| RollNo | Name | Phone |
|---|---|---|
| 1 | Alex | 444123 |
| 2 | Aryan | 421456 |
| 3 | Parker | 414259 |
| 4 | Jhon | 456785 |

**Advantage**

- Relational model is one of the most popular used database model.
- In relational model, changes in the database structure do not affect the data access.
- The revision of any information as tables consisting of rows and columns is much easier to understand.
- The relational database supports both data independence and structure independence concept which makes the database design, maintenance, administration and usage much easier than the other models.
- In this we can write complex query to accesses or modify the data from database.
- It is easier to maintain security as compare to other models.

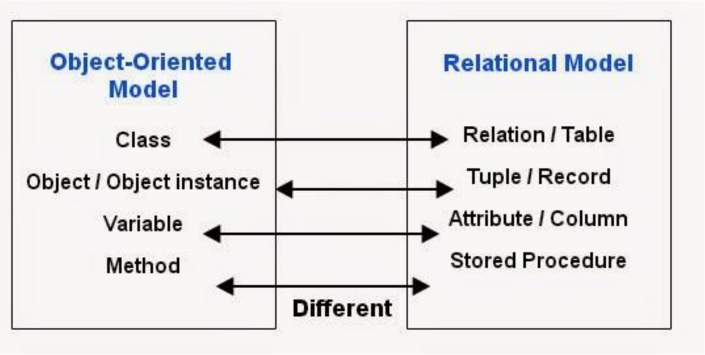
**Disadvantages**

- Mapping of objects in relational database is very difficult.
- Object oriented paradigm is missing in relation model.
- Data Integrity is difficult to ensure with Relational database.
- Relational Model is not suitable for huge database but suitable for small database.
- Hardware overheads are incurred which make it costly.
- Ease of design can lead to bad design.
- Relational database system hides the implementation complexities and the physical data storage details from the users.

### 4. Object-oriented databases

An object database is a system in which information is represented in the form of objects as used in object-oriented programming. Object oriented databases are different from relational databases which are table-oriented. The object-oriented data model is based on the object-oriented-programming language concept, which is now in wide use. Inheritance, polymorphism, overloading. object-identity, encapsulation and information hiding with methods to provide an interface to objects, are among the key concepts of object-oriented programming that have found applications in data modelling. The object-oriented data model also supports a rich type system, including structured and collection types.

The following figure shows the difference between relation and object-oriented database model.



**Advantages**

- Object database can handle different types of data while relational data base handles a single data. Unlike traditional databases like hierarchical, network or relational, the object-oriented databases can handle the different types of data, for example, pictures, voice video, including text, numbers and so on.

- Object-oriented databases provide us code reusability, real world modelling, and improved reliability and flexibility.
- The object-oriented database is having low maintenance costs as compared to other model because most of the tasks within the system are encapsulated, they may be reused and incorporated into new tasks.

**Disadvantages**

- There is no universally defined data model for an OODBMS, and most models lack a theoretical foundation.
- In comparison to RDBMSs the use of OODBMS is still relatively limited.
- There is a Lack of support for security in OODBMSs that do not provide adequate security mechanisms.
- The system more complex than that of traditional DBMSs.

### 5. Entity relationship Model

ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database. The ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.

ER Modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

**Following are the main components and its symbols in ER Diagrams:**

Rectangles: This Entity Relationship Diagram symbol represents entity types

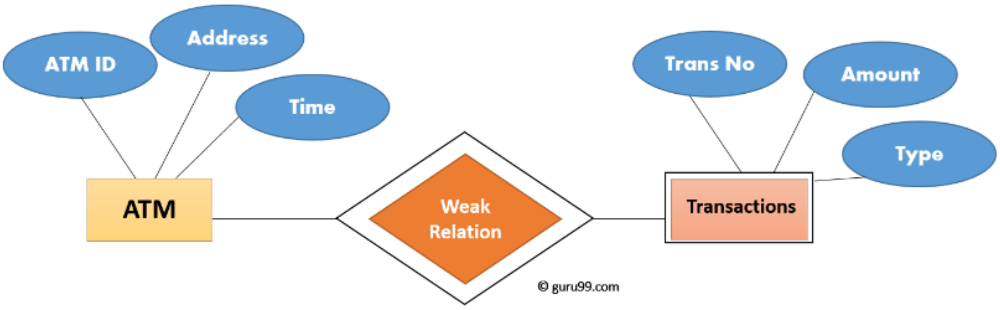
Ellipses: Symbol represent attributes

Diamonds: This symbol represents relationship types

Lines: It links attributes to entity types and entity types with other relationship types

Primary key: attributes are underlined

Double Ellipses: Represent multi-valued attributes



© guru99.com

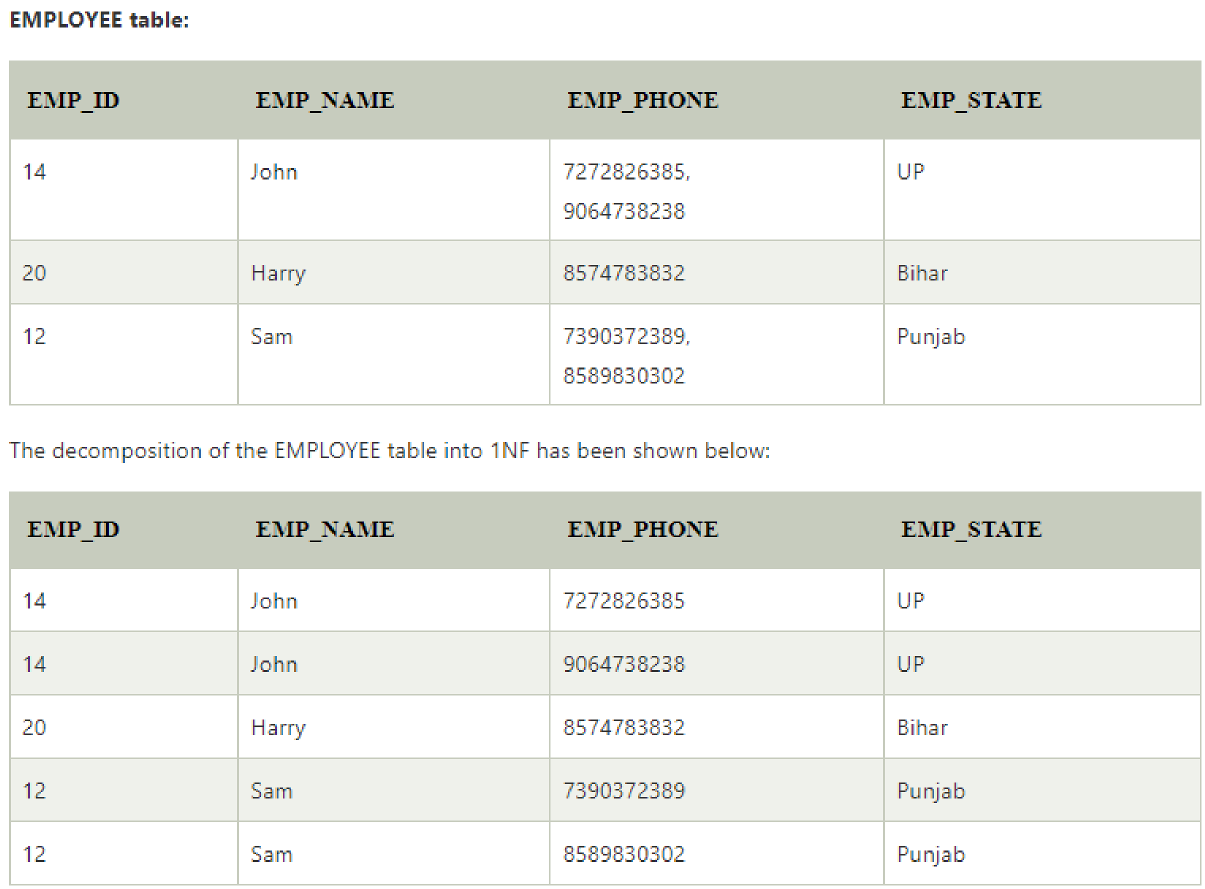
## 8. Concept of Normalization 1NF, 2NF and 3NF

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

### First Normal Form (1NF)

- A relation will be 1NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
- First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

Example: Relation EMPLOYEE is not in 1NF because of multi-valued attribute EMP_PHONE.



**EMPLOYEE table:**

| EMP_ID | EMP_NAME | EMP_PHONE | EMP_STATE |
|---|---|---|---|
| 14 | John | 7272826385,<br>9064738238 | UP |
| 20 | Harry | 8574783832 | Bihar |
| 12 | Sam | 7390372389,<br>8589830302 | Punjab |

The decomposition of the EMPLOYEE table into 1NF has been shown below:

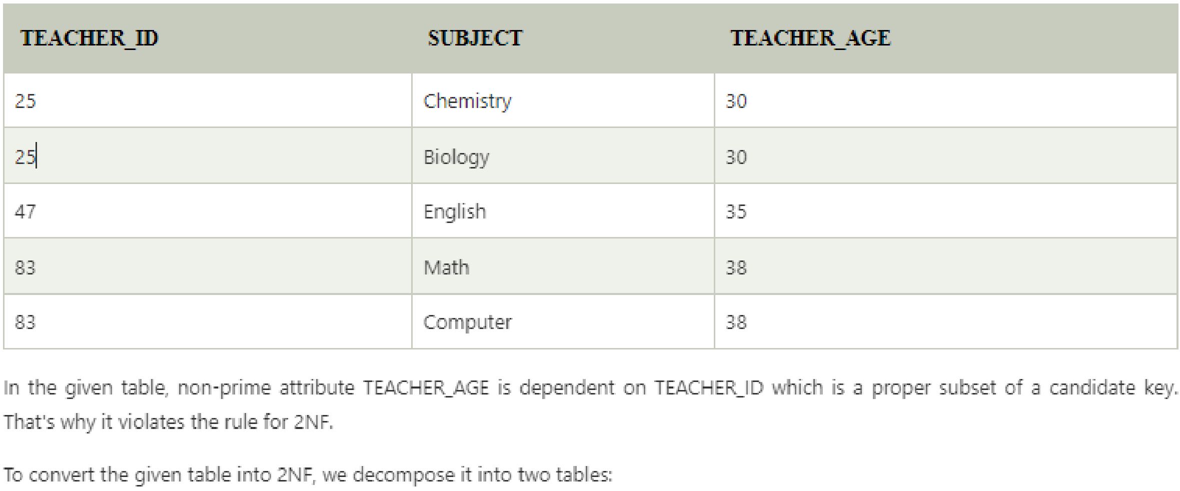
| EMP_ID | EMP_NAME | EMP_PHONE | EMP_STATE |
|---|---|---|---|
| 14 | John | 7272826385 | UP |
| 14 | John | 9064738238 | UP |
| 20 | Harry | 8574783832 | Bihar |
| 12 | Sam | 7390372389 | Punjab |
| 12 | Sam | 8589830302 | Punjab |

### Second Normal Form (2NF)

- In the 2NF, relational must be in 1NF.
- In the second normal form, all non-key attributes are fully functional dependent on the primary key

Example: Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

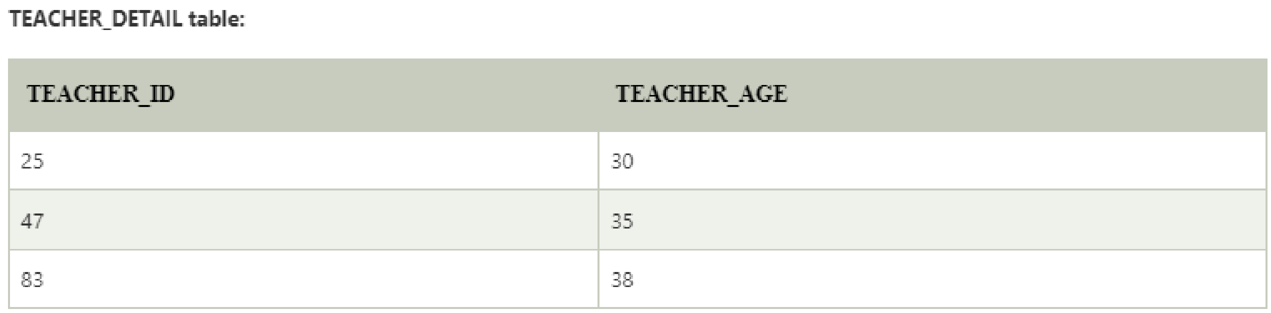
## TEACHER table



| TEACHER_ID | SUBJECT | TEACHER_AGE |
|---|---|---|
| 25 | Chemistry | 30 |
| 25 | Biology | 30 |
| 47 | English | 35 |
| 83 | Math | 38 |
| 83 | Computer | 38 |

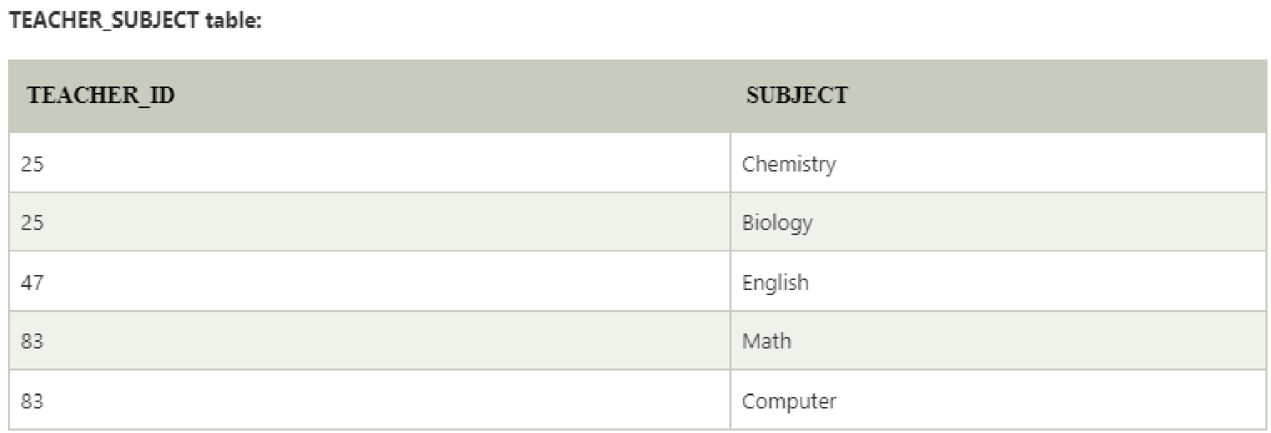
In the given table, non-prime attribute TEACHER_AGE is dependent on TEACHER_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF.

To convert the given table into 2NF, we decompose it into two tables:



**TEACHER_DETAIL table:**

| TEACHER_ID | TEACHER_AGE |
|---|---|
| 25 | 30 |
| 47 | 35 |
| 83 | 38 |



**TEACHER_SUBJECT table:**

| TEACHER_ID | SUBJECT |
|---|---|
| 25 | Chemistry |
| 25 | Biology |
| 47 | English |
| 83 | Math |
| 83 | Computer |

## Third Normal Form (3NF)

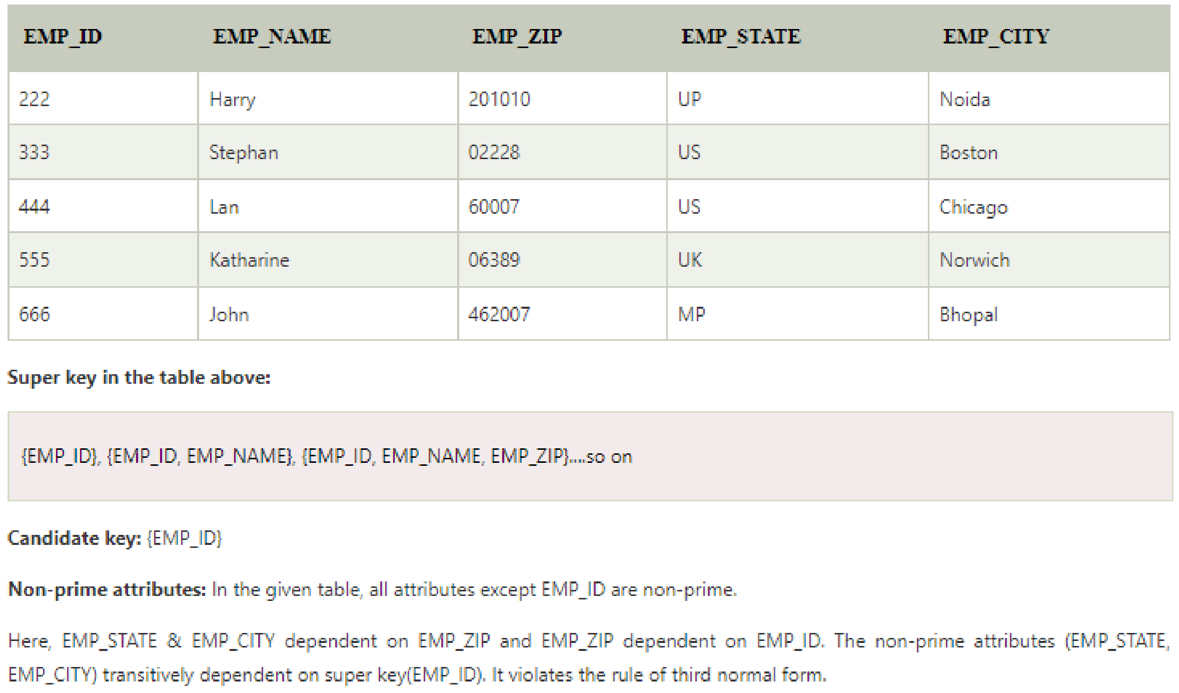
- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

A relation is in third normal form if it holds atleast one of the following conditions for every non-trivial function dependency X → Y.

1. X is a super key.
2. Y is a prime attribute, i.e., each element of Y is part of some candidate key.

Example:

EMPLOYEE_DETAIL table:



| EMP_ID | EMP_NAME | EMP_ZIP | EMP_STATE | EMP_CITY |
|---|---|---|---|---|
| 222 | Harry | 201010 | UP | Noida |
| 333 | Stephan | 02228 | US | Boston |
| 444 | Lan | 60007 | US | Chicago |
| 555 | Katharine | 06389 | UK | Norwich |
| 666 | John | 462007 | MP | Bhopal |

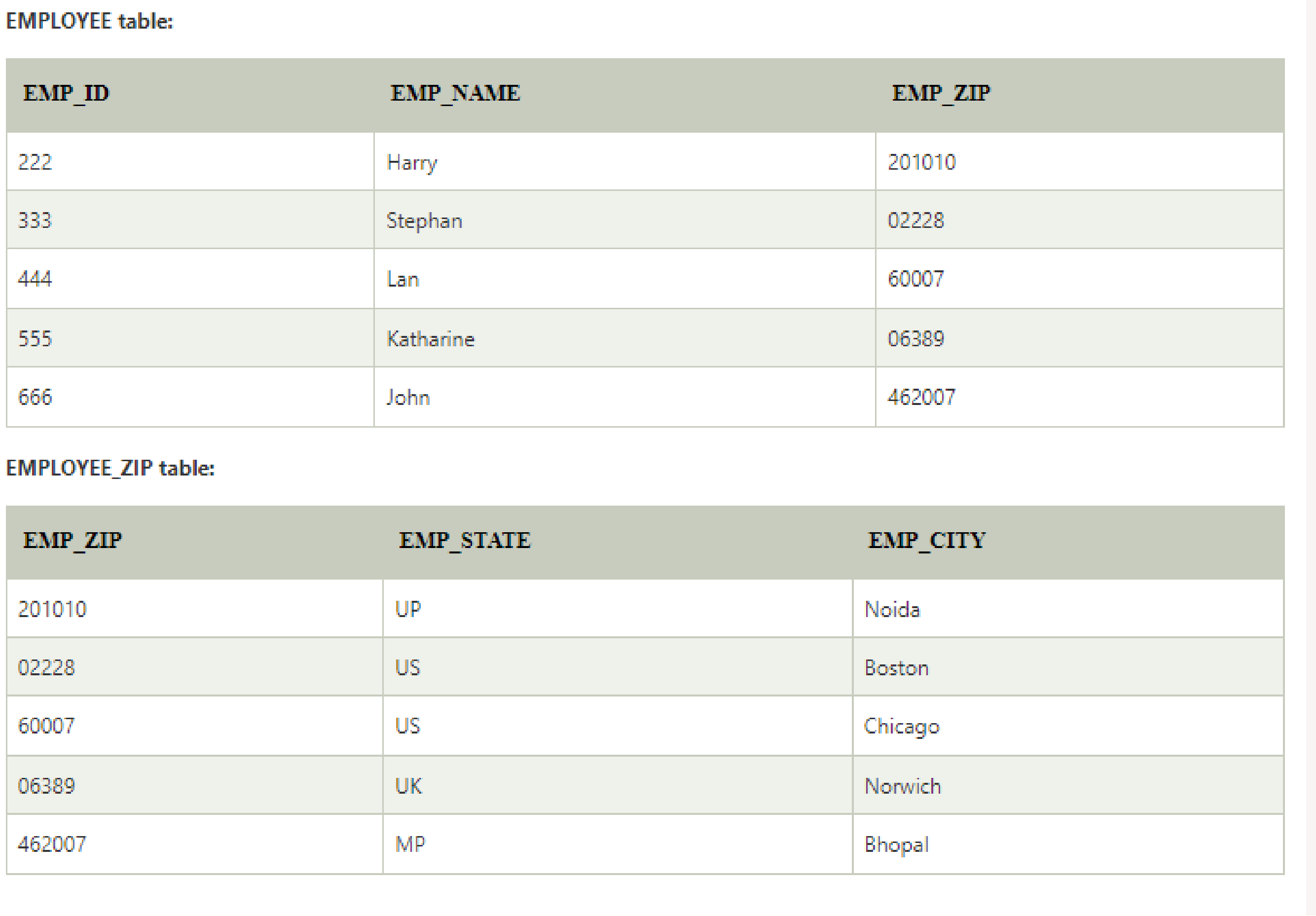
**Super key in the table above:**

```
{EMP_ID}, {EMP_ID, EMP_NAME}, {EMP_ID, EMP_NAME, EMP_ZIP}....so on
```

**Candidate key:** {EMP_ID}

**Non-prime attributes:** In the given table, all attributes except EMP_ID are non-prime.

Here, EMP_STATE & EMP_CITY dependent on EMP_ZIP and EMP_ZIP dependent on EMP_ID. The non-prime attributes (EMP_STATE, EMP_CITY) transitively dependent on super key(EMP_ID). It violates the rule of third normal form.



EMPLOYEE table:

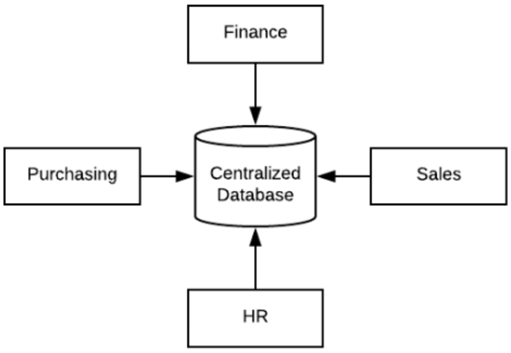
| EMP_ID | EMP_NAME | EMP_ZIP |
|---|---|---|
| 222 | Harry | 201010 |
| 333 | Stephan | 02228 |
| 444 | Lan | 60007 |
| 555 | Katharine | 06389 |
| 666 | John | 462007 |

EMPLOYEE_ZIP table:

| EMP_ZIP | EMP_STATE | EMP_CITY |
|---|---|---|
| 201010 | UP | Noida |
| 02228 | US | Boston |
| 60007 | US | Chicago |
| 06389 | UK | Norwich |
| 462007 | MP | Bhopal |

## 9. Centralized and Distributed Database

A centralized database is stored at a single location such as a mainframe computer. It is maintained and modified from that location only and usually accessed using an internet connection such as a LAN or WAN. The centralized database is used by organizations such as colleges, companies, banks etc.



As can be seen from the above diagram, all the information for the organization is stored in a single database. This database is known as the centralized database.

**Advantages**

Some advantages of Centralized Database Management System are −

- The data integrity is maximised as the whole database is stored at a single physical location. This means that it is easier to coordinate the data and it is as accurate and consistent as possible.
- The data redundancy is minimal in the centralised database. All the data is stored together and not scattered across different locations. So, it is easier to make sure there is no redundant data available.
- Since all the data is in one place, there can be stronger security measures around it. So, the centralised database is much more secure.
- Data is easily portable because it is stored at the same place.
- The centralized database is cheaper than other types of databases as it requires less power and maintenance.
- All the information in the centralized database can be easily accessed from the same location and at the same time.

**Disadvantages**

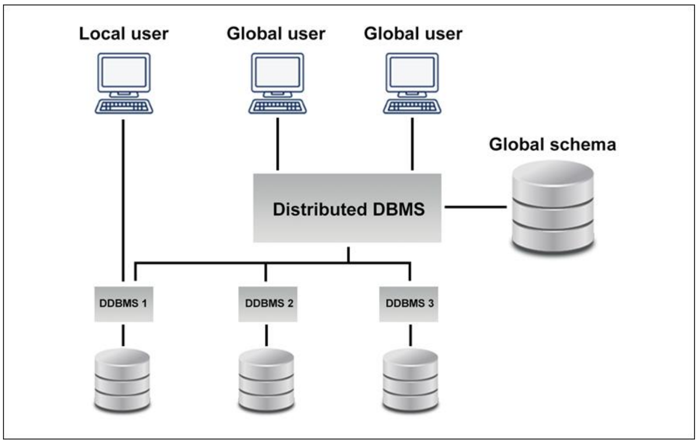
Some disadvantages of Centralized Database Management System are −

- Since all the data is at one location, it takes more time to search and access it. If the network is slow, this process takes even more time.
- There is a lot of data access traffic for the centralized database. This may create a bottleneck situation.
- Since all the data is at the same location, if multiple users try to access it simultaneously it creates a problem. This may reduce the efficiency of the system.
- If there are no database recovery measures in place and a system failure occurs, then all the data in the database will be destroyed.

## Distributed Database

In a distributed database, there are a number of databases that may be geographically distributed all over the world. A distributed DBMS manages the distributed database in a manner so that it appears as one single database to users. In the later part of the chapter, we go on to study the factors that lead to distributed databases, its advantages and disadvantages.

A distributed database is a collection of multiple interconnected databases, which are spread physically across various locations that communicate via a computer network.

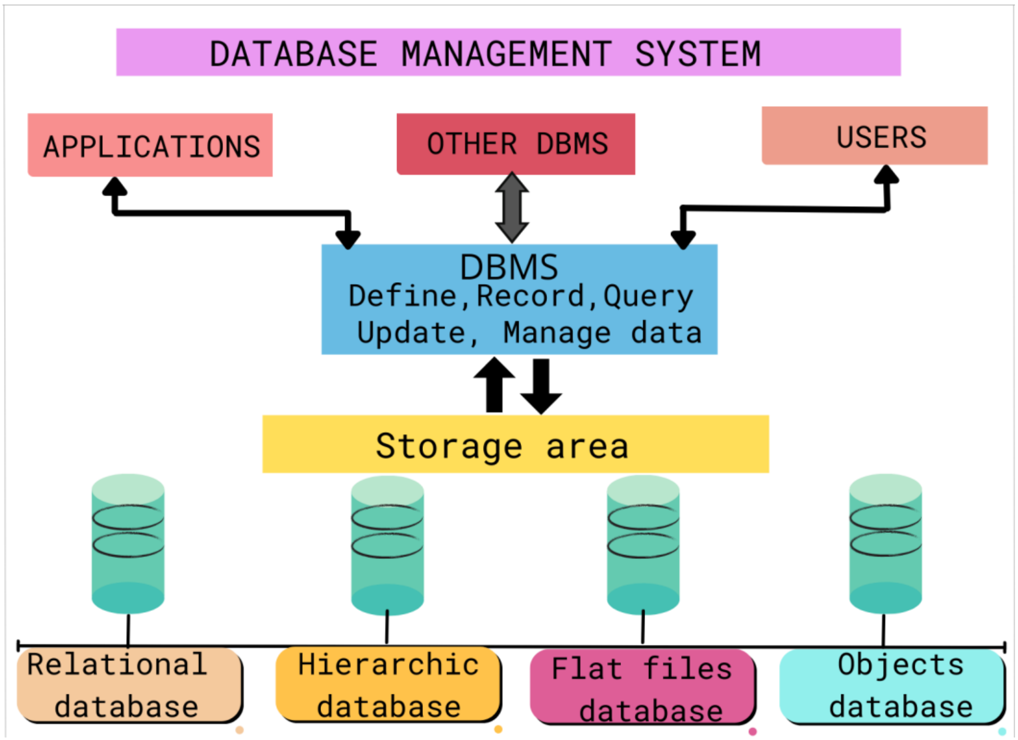


### Advantages of DDBMS

- The database is easier to expand as it is already spread across multiple systems and it is not too complicated to add a system.
- The distributed database can have the data arranged according to different levels of transparency i.e data with different transparency levels can be stored at different locations.
- The database can be stored according to the departmental information in an organisation. In that case, it is easier for a organisational hierarchical access.
- there were a natural catastrophe such as fire or an earthquake all the data would not be destroyed it is stored at different locations.
- It is cheaper to create a network of systems containing a part of the database. This database can also be easily increased or decreased.
- Even if some of the data nodes go offline, the rest of the database can continue its normal functions.

### Disadvantages of DDBMS

- The distributed database is quite complex and it is difficult to make sure that a user gets a uniform view of the database because it is spread across multiple locations.
- This database is more expensive as it is complex and hence, difficult to maintain.
- It is difficult to provide security in a distributed database as the database needs to be secured at all the locations it is stored. Moreover, the infrastructure connecting all the nodes in a distributed database also needs to be secured.
- It is difficult to maintain data integrity in the distributed database because of its nature. There can also be data redundancy in the database as it is stored at multiple locations.
- The distributed database is complicated and it is difficult to find people with the necessary experience who can manage and maintain it.


DATABASE MANAGEMENT SYSTEM
APPLICATIONS
OTHER DBMS
USERS
DBMS
Define,Record,Query
Update, Manage data
Storage area
Relational database
Hierarchic database
Flat files database
Objects database