Contents

[2 Chapter 3: Database Management System 3](#_Toc470852135)

[2.1 Introduction 3](#_Toc470852136)

[2.2 Components of Database System 3](#_Toc470852137)

[2.2.1 Functions of DBMS 4](#_Toc470852138)

[2.2.2 Advantages of DBMS 4](#_Toc470852139)

[2.2.3 Disadvantages of DBMS 4](#_Toc470852140)

[2.3 Database Architecture 4](#_Toc470852141)

[3 Logical two-tier Client / Server Architecture 4](#_Toc470852142)

[3.1.1 Three-tier Client / Server Architecture 5](#_Toc470852143)

[3.2 Types of Database Management Systems: 5](#_Toc470852144)

[3.2.1 Hierarchical Databases 6](#_Toc470852145)

[3.2.2 Network Database 7](#_Toc470852146)

[3.2.3 Relational Database 7](#_Toc470852147)

[3.2.4 Object-oriented Model 9](#_Toc470852148)

[3.3 Normalization of Database 10](#_Toc470852149)

[3.3.1 Problem Without Normalization 10](#_Toc470852150)

[3.4 Normalization Rule 11](#_Toc470852151)

[3.4.1 First Normal Form (1NF) 11](#_Toc470852152)

[3.4.2 Second Normal Form 14](#_Toc470852153)

[3.4.3 Third Normal Form 18](#_Toc470852154)

[3.4.4 Boyce Codd Normal Form (BCNF) 20](#_Toc470852155)

[3.4.5 4NF (Fourth Normal Form) Rules 21](#_Toc470852156)

[3.4.6 5NF (Fifth Normal Form) Rules 21](#_Toc470852157)

[3.4.7 6NF (Sixth Normal Form) Proposed 21](#_Toc470852158)

[4 Summary 21](#_Toc470852159)

[5 Operating System 22](#_Toc470852160)

[5.1 Introduction 22](#_Toc470852161)

[5.2 Function Of Operating System 22](#_Toc470852162)

[5.2.1 Booting 22](#_Toc470852163)

[5.2.2 Memory Management 22](#_Toc470852164)

[5.2.3 Loading And Execution 22](#_Toc470852165)

[5.2.4 Data Security 22](#_Toc470852166)

[5.2.5 Disk Management 22](#_Toc470852167)

[5.2.6 Process Management 22](#_Toc470852168)

[5.2.7 Device Controlling 23](#_Toc470852169)

[5.2.8 Printing Controlling 23](#_Toc470852170)

[5.2.9 Providing Interference 23](#_Toc470852171)

[5.3 Classification Of Operating System 23](#_Toc470852172)

[5.3.1 Multiuser OS: 24](#_Toc470852173)

[5.3.2 Multiprocessing OS: 24](#_Toc470852174)

[5.3.3 Multiprogramming OS: 24](#_Toc470852175)

[5.3.4 Multitasking OS: 24](#_Toc470852176)

[5.3.5 Multithreading: 24](#_Toc470852177)

[5.3.6 Batch Processing: 24](#_Toc470852178)

[5.3.7 Online Processing: 25](#_Toc470852179)

[6 Introduction To IT 25](#_Toc470852180)

[6.1 Introduction 25](#_Toc470852181)

[6.2 I/O Devices 25](#_Toc470852182)

[6.2.1 Input Device 25](#_Toc470852183)

[6.2.2 Output Devices 26](#_Toc470852184)

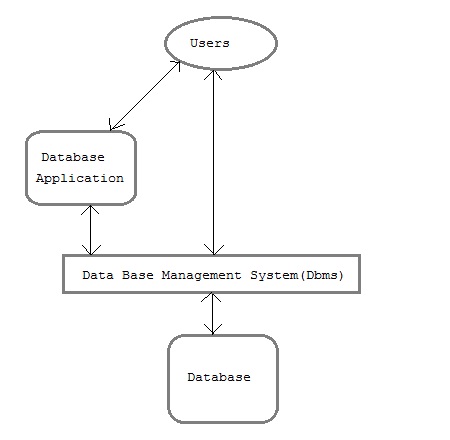
# Chapter 3: Database Management System

## Introduction

Database Management System or DBMS in short refers to the technology of storing and retrieving users’ data with utmost efficiency along with appropriate security measures. A Database is a collection of records. Database management systems are designed as the means of managing all the records. Database Management is a software system that uses a standard method and running queries with some of them designed for the oversight and proper control of databases.

## Components of Database System

The database system can be divided into four components.



**Users:** Users may be of various type such as DB Administrator, System Developer and End Users.

**Database Application:** Database application may be Personal, Departmental, Enterprise and Internal Software that allow users to define, create and manages database access, Ex: MySQL, Oracle etc.

**Database:** Collection of logical data

### Functions of DBMS

* Provides data Independence
* Concurrency Control
* Provides Recovery services
* Provides Utility services
* Provides a clear and logical view of the process that manipulates data.

### Advantages of DBMS

* Segregation of application program.
* Minimal data duplicacy.
* Easy retrieval of data.
* Reduced development time and maintenance need.

### Disadvantages of DBMS

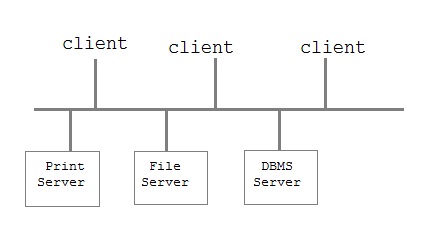
* Complexity
* Costly
* Large in size

## Database Architecture

Database architecture is logically divided into two types:

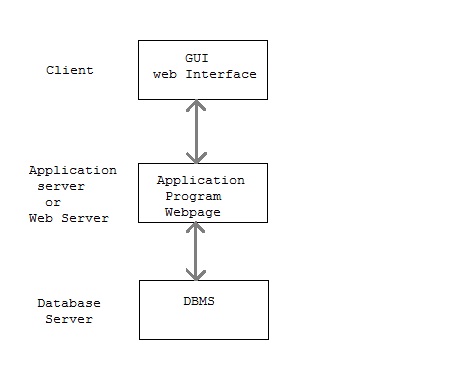
1. Logical two-tier Client / Server Architecture
2. Logical three-tier Client / Server Architecture

# Logical two-tier Client / Server Architecture



Two-tier Client / Server architecture is used for User Interface program and Application Programs that runs on client side. An interface called ODBC (Open Database Connectivity) provides an API that allow client side program to call the dbms. Most DBMS vendors provide ODBC drivers. A client program may connect to several DBMS's. In this architecture some variation of client is also possible for example in some DBMS's more functionality is transferred to the client including data dictionary, optimization etc. Such clients are called **Data Server**.

### Three-tier Client / Server Architecture



Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called **Application Server** or Web Server stores the web connectivity software and the business logic(constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending partially processed data between the database server and the client.

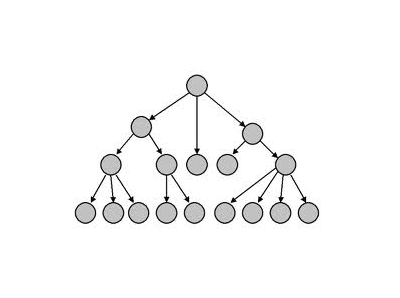
## Types of Database Management Systems:

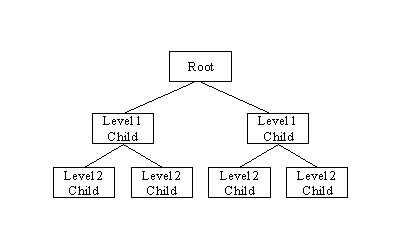
There are four structural types of database management systems:

* HierarchicalDatabase
* Network Database
* Relational Database
* Object Oriented Database

### Hierarchical Databases

In the Hierarchical Database Model, we have to learn about the databases. It is very fast and simple. In a hierarchical database, records contain information about their groups of parent/child relationships, just like as a tree structure. The structure implies that a record can have also a repeating information. In this structure Data follows a series of records, it is a set of field values attached to it. It collects all records together as a record type. These record types are the equivalent of tables in the relational model, and with the individual records being the equivalent of rows. To create links between these record types, the hierarchical model uses these type Relationships.



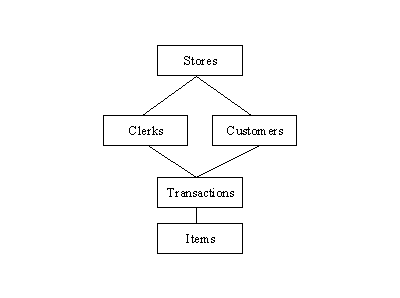


**Advantage:** Hierarchical database can be accessed and updated rapidly because in this model structure is like as a tree and the relationships between records are defined in advance. This feature is a two-edged.

**Disadvantage:**This type of database structure is that each child in the tree may have only one parent, and relationships or linkages between children are not permitted, even if they make sense from a logical standpoint. Hierarchical databases are so in their design. it can adding a new field or record requires that the entire database be redefined.

### Network Database

A network databases are mainly used on a large digital computer. It more connections can be made between different types of data, network databases are considered more efficiency It contains limitations must be considered when we have to use this kind of database. It is Similar to the hierarchical databases, network databases. Network databases are similar to hierarchical databases by also having a hierarchical structure. A network database looks more like a cobweb or interconnected network of records. In network databases, children are called members and parents are called occupier. The difference between each child or member can have more than one parent.

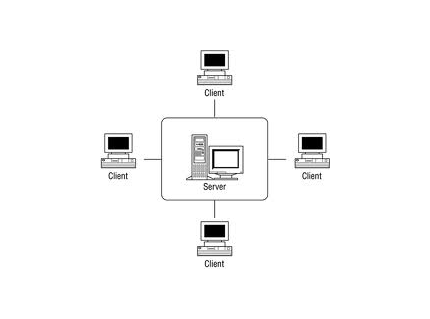


The Approval of the network data model similar with the esteem of the hierarchical data model. Some data were more naturally modeled with more than one parent per child. The network model authorized the modeling of many-to-many relationships in data.

### Relational Database

In relational databases, the relationship between data files is relational. Hierarchical and network databases require the user to pass a hierarchy in order to access needed data. These databases connect to the data in different files by using common data numbers or a key field. Data in relational databases is stored in different access control tables, each having a key field that mainly identifies each row. In the relational databases are more reliable than either the hierarchical or network database structures. In relational databases, tables or files filled up with data are called relations (tuples) designates a row or record, and columns are referred to as attributes or fields.

Relational databases work on each table has a key field that uniquely indicates each row, and that these key fields can be used to connect one table of data to another.



**The relational database has two major reasons**

Relational databases can be used with little or no training.

Database entries can be modified without specify the entire body.

**Properties of Relational Tables:**

In the relational database we have to follow some properties which are given below.

It's Values is Atomic

In Each Row is alone.

Column Values are of the Same thing.

Columns is undistinguished.

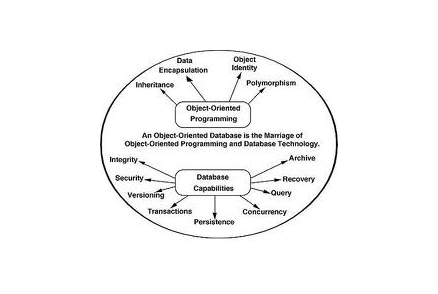
Sequence of Rows is Insignificant.

Each Column has a common Name.

### Object-oriented Model

In this Model we have to discuss the functionality of the object oriented Programming. It takes more than  storage of programming language objects. Object DBMS's increase the semantics of the C++ and Java. It provides full-featured database programming capability, while containing native language compatibility. It adds the database functionality to object programming languages. This approach is the analogical of the application and database development into a constant data model and language environment. Applications require less code, use more natural data modeling, and code bases are easier to maintain. Object developers can write complete database applications with a decent amount of additional effort.

The object-oriented database derivation is the integrity of object-oriented programming language systems and consistent systems. The power of the object-oriented databases comes from the cyclical treatment of both consistent data, as found in databases, and transient data, as found in executing programs.



Object-oriented databases use small, recyclable separated of software called objects. The objects themselves are stored in the object-oriented database. Each object contains of two elements:

Piece of data (e.g., sound, video, text, or graphics).

Instructions, or software programs called methods, for what to do with the data.

**Disadvantage of Object-oriented Databases**

1. Object-oriented databases have these disadvantages
2. Object-oriented database is more expensive to develop.
3. In the Most organizations are unwilling to abandon and convert from those databases.

They have already invested money in developing and implementing. The benefits to object-oriented databases are compelling. The ability to mix and match reusable objects provides incredible multimedia capability.

## Normalization of Database

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purpose,

* Eliminating redundant (useless / duplicate) data.
* Ensuring data dependencies make sense i.e. data is logically stored.

### Problem Without Normalization

Without Normalization, it becomes difficult to handle and update the database, without facing data loss. Insertion, Updation and Deletion Anomalies are very frequent if Database is not Normalized. To understand these anomalies let us take an example of student table.

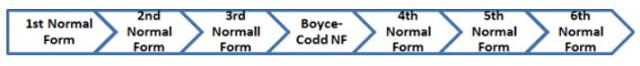
|  |  |  |  |
| --- | --- | --- | --- |
| **Student ID** | **Student Name** | **Student Address** | **Subject** |
| **645** | **Sudarshan** | **Pepsicola** | **Physics** |
| **616** | **Bandana** | **Gonganbu** | **Chemistry** |
| **640** | **Seema** | **Goganbu** | **Mathematics** |
| **641** | **Shrayash** | **Gonganbu** | **English** |

* **Updation Anomaly:** To update address of a student who occurs twice or more than twice in a table, we will have to update student address column in all the rows, else data will become inconsistent.
* **Insertion Anomaly:** Suppose for a new admission, we have a student id, name and address of a student but if student has not opted for any subjects yet then we have to insert **NULL** there, leading to Insertion Anamoly.
* **Deletion Anomaly:** If student id 641 has only one subject and temporarily he drops it, when we delete that row, entire student record will be deleted along with it.

## Normalization Rule

Normalization rule are divided into following normal form.

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. BCNF

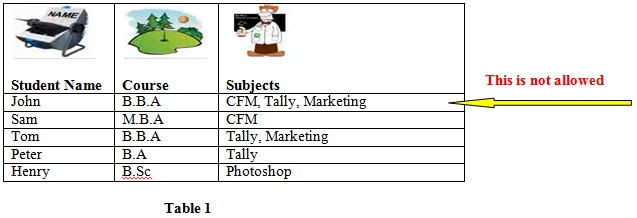


### First Normal Form (1NF)

As per First Normal Form, no two rows of data must contain repeating group of information i.e. each set of column must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique.

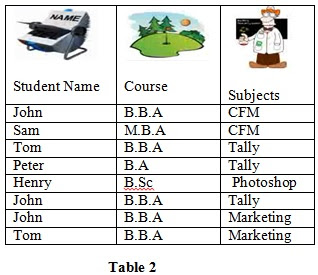
The **Primary key** is usually a single column, but sometimes more than one column can be combined to create a single primary key. For example, consider a table which is not in First normal form

**Student Table:**



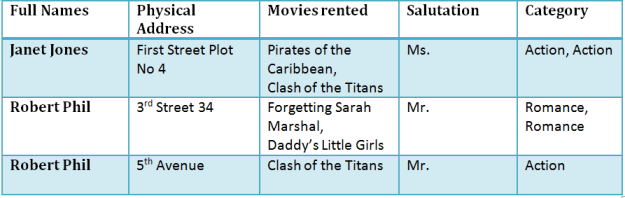
In First Normal Form, any row must not have a column in which more than one value is saved, like separated with commas. Rather than that, we must separate such data into multiple rows.

Students Table following 1NF will be:



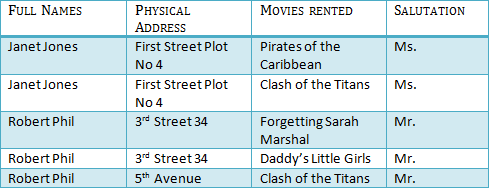
Using the First Normal Form, data redundancy increases, as there will be many columns with same data in multiple rows but each row as a whole will be unique.

Let’s look at another example:



Now let’s move it to 1st Normal Forms, then, according to **1NF Rules**

1. Each table cell should contain single value.
2. Each record needs to be unique.



**Table B: In 1NF Form**

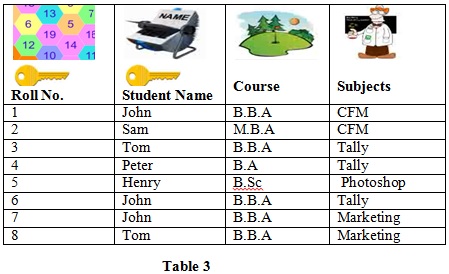
#### Key

A key is a value used to uniquely identify a record in a table. A key should be a single column or combination of multiple columns.

**Note:** Columns in a table that are not used to uniquely identify a record are called non-key columns.

#### Primary Key

A primary key is a column (or columns) in a table that uniquely identifies the rows in that table.



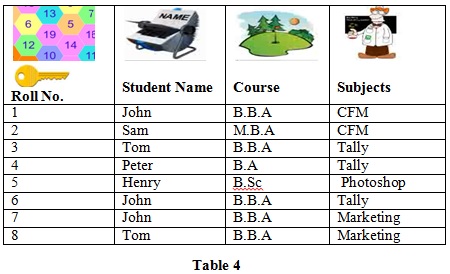
For example, in the table above, we have created one new field called Roll no which represents the primary key.

The values placed in primary key columns must be unique for each row: no duplicates can be tolerated. In addition, nulls are not allowed in primary key columns.

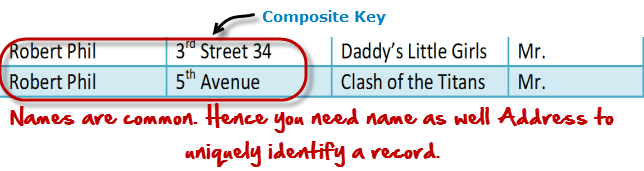
#### Composite Key

A composite key is a primary key composed of multiple columns used to identify a record uniquely.

In Table 4, both **Roll No** and **Student Name** are Primary Keys, combinedly a **Composite key.**



In our database, we have two people with the same name Robert Phil but they live at different places.

Hence we require both Full Nam and Address to uniquely identify record. This is a composite key.

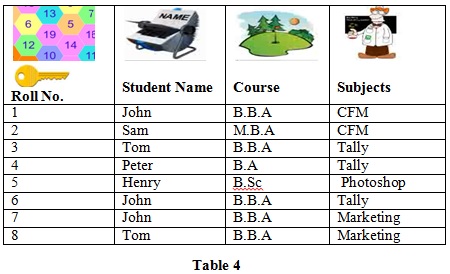
### Second Normal Form

As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails **Second normal form**.

#### 2NF Rules

* Be in 1NF
* Single Column Primary Key

2NF enforces that only one Field must be Primary Key. So Table 3 is not in 2NF. Table 4 below represents 2NF. Note that a Table in 2NF must obey all the rules of 1NF plus One Field Primary Key.



It is clear that we can’t move forward to make our simple database in 2nd Normalization form unless we partition the above table.

[](http://cdn.guru99.com/images/Table2.png)

**Table A**

[](http://cdn.guru99.com/images/Table1.png)

**Table B**

We have divided our 1NF table into two tables viz. Table A and Table B. Table A contains member information. Table B contains information on movies rented.

We have introduced a new column called membership id which is the primary key for the table A. Records can be uniquely identified in Table A using membership id.

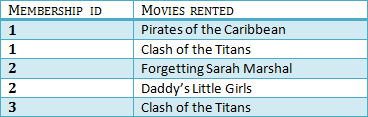
#### Foreign Keys

Foreign keys are columns that point to primary key columns. So, for example, Order No is the primary key of the table ORDERS below and Customer No is a foreign key that points to the primary key in the CUSTOMERS table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order No** | **Employee No** | **Customer No** | **Supplier** | **Price** | Item |
| 1 | 1 | 42 | Harrison | $235 | Desk |
| 2 | 4 | 1 | Ford | $234 | Chair |
| 3 | 1 | 68 | Harrison | $415 | Table |
| 4 | 2 | 112 | Ford | $350 | Lamp |
| 5 | 3 | 42 | Ford | $234 | Chair |
| 6 | 2 | 112 | Ford | $350 | Lamp |
| 7 | 2 | 42 | Harrison | $235 | Desk |

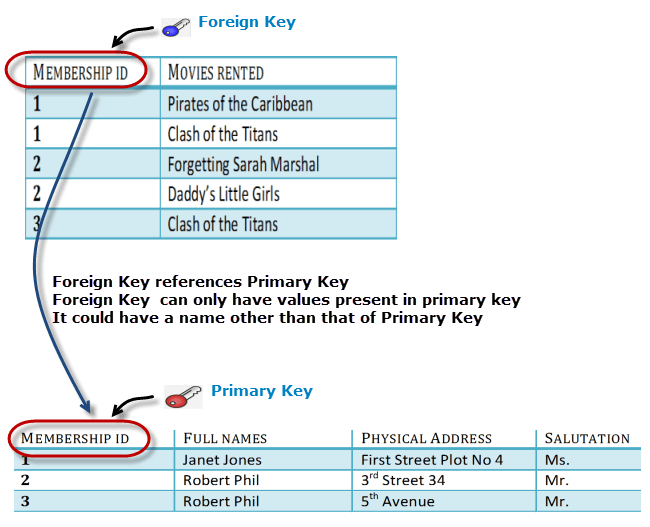
In the context of relational database, a **foreign key** is a field (or collection of fields) in one table that uniquely identifies a row of another table or the same table.In simpler words, the foreign key is defined in a second table, but it refers to the primary key in the first table. A table may have multiple foreign keys, and each foreign key can have a different parent table. Each foreign key is enforced independently by the database system. Therefore, cascading relationships between tables can be established using foreign keys.

In Table 2, Membership ID is the foreign Key

[](http://cdn.guru99.com/images/foreign_key_table.png)

Foreign Key references primary key of another Table! It helps connect your Tables

* A foreign key can have a different name from its primary key
* It ensures rows in one table have corresponding rows in another
* Unlike Primary key they do not have to be unique. Most often they aren't
* Foreign keys can be null even though primary keys can not

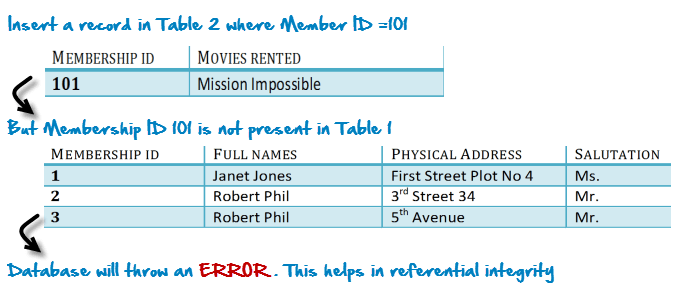


#### Why do you need a foreign key?

When we use foreign key we get

* data integrity in the database is achieved
* amount of data duplication is reduced
* faster queries

Suppose a record is inserted in Table in a such a way that we will only be able to insert values into our foreign key that exist in the unique key in the parent table. This helps in referential integrity.



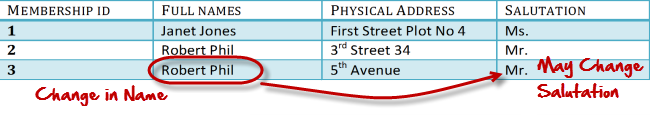
The above problem can be overcome by declaring membership id from Table2  as foreign key of membership id  from Table1.

Now, if somebody tries to insert a value in the membership id field that does not exist in the parent table, an error will be shown.

#### Transitive Dependency

In Database Management System, a **transitive dependency** is a functional dependency which holds by virtue of transitivity. A transitive dependency can occur only in a relation that has three or more attributes. A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change.

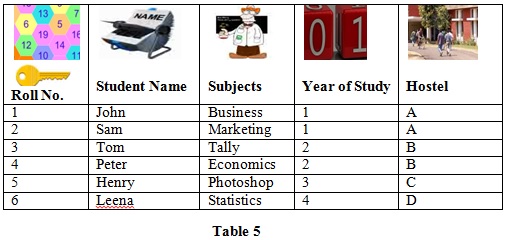
Consider the table 1, changing the non-key column Full Name, may change Salutation.

[](http://cdn.guru99.com/images/transitive_functional_dependencies.png)

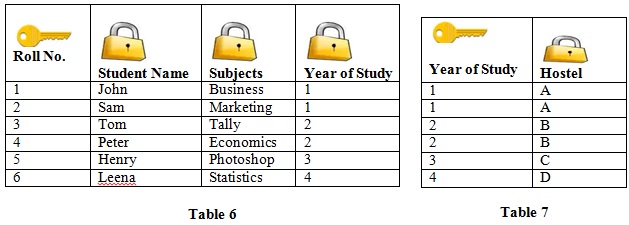
### Third Normal Form

**Third Normal form** applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So this transitive functional dependency should be removed from the table and also the table must be in **Second Normal form**.

If we give close attention to the Table 5 there is a functional dependency between “Year of Study” and “Hostel”. This means that a non-primary key column is controlling another non-primary key column. Here dependency is that 1st year student is staying in Hostel A, 2nd year student is staying in Hostel B and so on. This violates the second rule of **3NF.**



To represent Table 5 in **3NF**, we need to divide Table 5 into two other tables as shown and establish relationship between the two.



For example, consider a table with following fields.

[](http://cdn.guru99.com/images/2NFTable1.png)

**Table A**

[](http://cdn.guru99.com/images/2NFTable2.png)

**Table B**

[](http://cdn.guru99.com/images/2NFTable3.png)

**Table C**

We have again divided our tables and created a new table which stores Salutations.

There are no transitive functional dependencies and hence our table is in 3NF

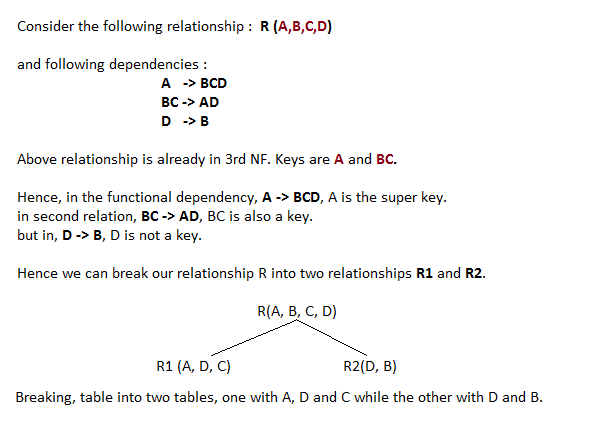
In Table C Salutation ID is primary key and in Table A Salutation ID is foreign to primary key in Table C

Now our little example is in a level that cannot further be decomposed to attain higher forms of normalization. In fact it is already in higher normalization forms. Separate efforts for moving in to next levels of normalizing data are normally needed in complex databases. However, we will be discussing about next levels of normalizations in brief in the following.

### Boyce Codd Normal Form (BCNF)

**Boyce Codd Normal Form** is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

* R must be in 3rd Normal Form
* and, for each functional dependency (X -> Y), X should be a super Key.



### 4NF (Fourth Normal Form) Rules

If no database table instance contains two or more, independent and multivalued data describing the relevant entity, then it is in 4th Normal Form.

### 5NF (Fifth Normal Form) Rules

A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed in to any number of smaller tables without loss of data.

### 6NF (Sixth Normal Form) Proposed

6th Normal Form is not standardized yet however it is being discussed by database experts for some time. Hopefully we would have clear standardized definition for 6th Normal Form in near future.

# Summary

* Database designing is critical to the successful implementation of a database management system that meets the data requirements of an enterprise system.
* Normalization helps produce database systems that are cost effective, cost effective and have better security models.
* Functional dependencies are a very important component of the normalize data process
* Most database systems are normalized database up to the third normal forms.
* A primary uniquely identifies are record in a Table and cannot be null
* A foreign key helps connect table and references a primary key

# Operating System

## Introduction

An operating system is a system software which is used to interact with the application software and the computer hardware. The computer cannot be functional without an operating system.

## Function Of Operating System

### Booting

Booting is a process of starting the computer operating system starts the computer to work. It checks the computer and makes it ready to work.

### Memory Management

It is also an important function of operating system. The memory cannot be managed without operating system. Different programs and data execute in memory at one time. if there is no operating system, the programs may mix with each other. The system will not work properly.

### Loading And Execution

A program is loaded in the memory before it can be executed. Operating system provides the facility to load programs in memory easily and then execute it.

### Data Security

Data is an important part of computer system. The operating system protects the data stored on the computer from illegal use, modification or deletion.

### Disk Management

Operating system manages the disk space. It manages the stored files and folders in a proper way.

### Process Management

CPU can perform one task at one time. if there are many tasks, operating system decides which task should get the CPU.

### Device Controlling

Operating System also controls all devices attached to computer. The hardware devices are controlled with the help of small software called device drivers.

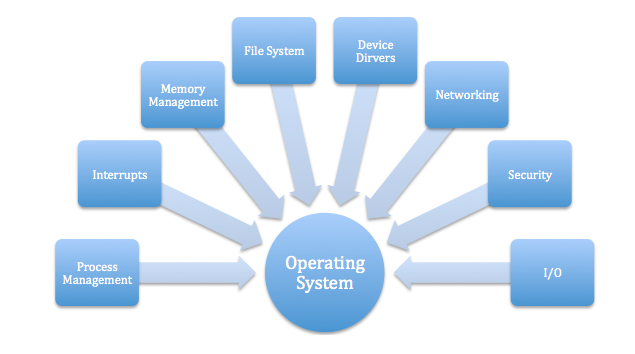
### Printing Controlling

Operating System also controls printing function. It a user issues two print commands at a time, it does not mix data of these files and prints them separately.

### Providing Interference

t is used in order that user interface acts with a computer mutually. User interface controls how you input data and instruction and how information is displayed on screen. The operating system offers two types of the interface to the user:

* **Graphical-line interface**: It interacts with of visual environment to communicate with the computer. It uses windows, icons, menus and other graphical objects to issues commands.
* **Command-line interface**: it provides an interface to communicate with the computer by typing commands.



## Classification Of Operating System

OS can be classified based on numbers of users, processing and scalable parameters.

### ****Multiuser OS:****

In a multiuser OS, more than one user can use the same system at a same time through the multi I/O terminal or through the network.

For example: windows, Linux, Mac, etc.

A multiuser OS uses timesharing to support multiple users.

### ****Multiprocessing OS:****

A multiprocessing OS can support the execution of multiple processes at the same time. It uses multiple number of CPU. It is expensive in cost however, the processing speed will be faster. It is complex in its execution. Operating system like Unix, 64-bit edition of windows, server edition of windows, etc. are multiprocessing.

### ****Multiprogramming OS:****

In a multiprogramming OS more than one programs can be used at the same time. It may or may not be multiprocessing. In a single CPU system, multiple program are executed one after another by dividing the CPU into small time slice. Example: Windows, Mac, Linux, etc.

### ****Multitasking OS:****

In a multitasking system more than one task can be performed at the same time but they are executed one after another through a single CPU by time sharing. For example: Windows, Linux, Mac, Unix etc. Multitasking OS are of two types:

* Pre-empetive multitasking
* Co-operative multitasking

In the pre-empetive multitasking, the OS allows CPU times slice to each program. After each time slice, CPU executes another task. Example: Windows XP

In co-operative multitasking a task can control CPU as long as it requires . However, it will free CPU to execute another program if it doesn’t require CPU. Exaample: windows 3.x, multifinder,etc.

### ****Multithreading:****

A program in execution is known as process. A process can be further divided into multiple sub-processers. These sub-processers are known as threads. A multi-threading OS can divide process into threads and execute those threads. This increases operating speed but also increases the complexity. For example: Unix, Server edition of Linux and windows.

### ****Batch Processing:****

A batch processing is a group of processing system in which all the required input of all the processing task is provided initially. The result of all the task is provided after the completion of all the processing. Its main functions are:

* Multiple task are processed
* User cannot provide input in between the processing
* It is appropriate only when all the inputs are known in advance
* It requires large memory
* CPU ideal time is less
* Printer is the appropriate output device
* It is old processing technique and rarely used at present

### ****Online Processing:****

It is an individual processing system in which the task is processed on individual basis as soon as they are provided by the user. It has features like:

* Individual task is processed at a time
* User can provide input in between processing
* It is appropriate when all inputs ate not known in advance
* It doesn’t require large memory
* CPU ideal time is more
* Monitor is appropriate output device
* It is modern processing technique and mostly used in present

# Introduction To IT

## Introduction

Information technology (IT) is the application of computers and internet to store, study, retrieve, transmit, and manipulate data, or information, often in the context of a business or other enterprise. IT is considered a subset of information and communication technology.

## I/O Devices

### Input Device

An input device is a peripheral device (piece of computer hardware equipment) used to provide data and control signals to an information processing system such as a computer or information appliance. Examples of input devices includes Keyboard, Mouse, Microphone, Image scanner, Microphone.



### Output Devices

An output device is any piece of computer hardware item which utilizes whatever data and commands from your computer in order to perform a task. This leads to the results of data processing carried out by an information processing system (such as a computer) which converts the electronically generated information into human-readable form. Examples of output devices include Monitor, Speakers, Printers, Headphones, Projectors.

