
RGPVNOTES.IN

Program : **B.Tech**

Subject Name: **Basic Computer Engineering**

Subject Code: **BT-205**

Semester: **2nd**

**UNIT V**

**Database-management system**

A database management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise.

The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

**Purpose of Database Approach**

**File-Based Approach**

The traditional file-oriented approach to information processing has for each application a separate master file and its own set of personal files. COBOL language supported these file-oriented applications. It was used for developing applications such as of payroll, inventory, and financial accounting. However, in general an organization needs flow of information across these applications also and this requires sharing of data, which is very difficult to implement in the traditional file approach. In addition, a major limitation of file-based approach is that the programs are dependent on the files and the files are dependent upon the programs.

These file-based approaches, which came into being as the first commercial applications of computers, suffered from the following significant disadvantages:

**The disadvantages of file system processing**



**(i) Data Redundancy:**

○ Since files and applications are created by the different programmer of various departments over a long period of time, it might lead to several problems:

o Inconsistency in data format

o The same information may be kept in several different places (files).

o Data inconsistency which means various copies of the same data are conflicting; waste storage space and duplication of effort

**(ii) Data Isolation**

o It is difficult for a new application to retrieve the appropriate data which might be stored in various files.

**(iii) Integrity problems**

o Data values must satisfy certain consistency constraints which are specified in the application programs.

o It is difficult to add change the programs to enforce new constraint

**(iv) Security problems**

o There are constraint regarding accessing privileges

o Application is added to the system in an ad-hoc manner so it is difficult to enforce those constraints

**(v) Concurrent – access anomalies**

o Data may be accessed by many applications that have not been coordinated previously so it is not easy to provide a strategy to support multiple users to update data simultaneously.

**Database Approach**

**Fundamental Concepts**

The database is a shared collection of related data which will be used to support the activities of the organization. The database can be viewed as a repository of data that is defined once and then is accessed by various users.



**A database has the following properties**:
- It is a representation of some aspect of the real world; or perhaps, a collection of data elements (facts) representing real-world information.
- The database is logically coherent and internally consistent.
- The database is designed, built, and populated with data for a specific purpose.

**Characteristics of Database approach**

There are many characteristics that distinguish the database approach with the file-based approach. In this section, we describe in detail some of those important characteristics.

**- Self-Describing Nature of a Database System**: Database System contains not only the database itself but also the descriptions of data structure and constraints (meta-data). This information is used by the DBMS software or database users if needed.

- **Insulation between Program and Data**: In the filed base system, the structure of the data files is defined in the application programs so if the user wants to change the structure of a file, all the programs access to that files might need to be changed.

- **Support multiple views of data**: A view is a subset of the database which is defined and dedicated for users of the system. Multiple users in the system might have different views of the system. Each view might contain only the interested data of a user or a group of users.

- **Sharing of data and Multi-user system**: A multi-user database system must allow multiple users to access the database at the same time. As the result, the multi-user DBMS must have concurrency control strategies to ensure that several users tries to access the same data item at a time do so in the manner so that the data always be correct.



**Benefits of Database Approach**

**-To Control Data Redundancy**

o In the Database approach, ideally, each data item is stored in only one place in the database

o However, in some case redundancy is still exists to improving system performance, but such redundancy is controlled and kept to minimum

– **Data Sharing**

o The integration of the whole data in an organization leads to the ability to produce more information from a given amount of data

– **Enforcing Integrity Constraints**

o DBMSs should provide capabilities to define and enforce certain constraints such as data type, data uniqueness.

**- Restricting Unauthorized Access**

o Not all users of the system have the same accessing privileges.

o DBMSs should provide a security subsystem to create and control the user accounts.

**- Data Independence**

o The system data descriptions are separated from the application programs.

o Changes to the data structure are handled by the DBMS and not embedded in the program.

**- Transaction Processing**

o The DBMS must include concurrency control subsystem to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct.

**- Providing multiple views of data**

o A view may be a subset of the database. Various users may have different views of the database itself.

o Users may not need to be aware of how and where the data they refer to is stored

**- Providing backup and recovery facilities**

o If the computer system fails in the middle of a complex update program, the recovery subsystem is responsible for making sure that the database is restored to the stage it was in before the program started executing.

**Data Models in DBMS**

A database model shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed. Individual database models are designed based on the rules and concepts of whichever broader data model the designers adopt. Most data models can be represented by an accompanying database diagram.

**Entity-Relationship Model**

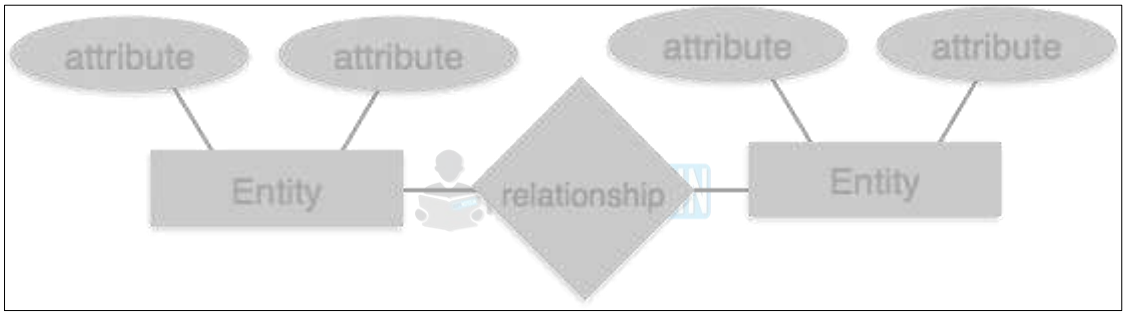
Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

ER Model is best used for the conceptual design of a database.

ER Model is based on −

- **Entities** and their *attributes.*
- **Relationships** among entities.

These concepts are explained below.




E-R Diagram

- **Entity** − An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.
- **Relationship** − The logical association among entities is called ***relationship***. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

  Mapping cardinalities −

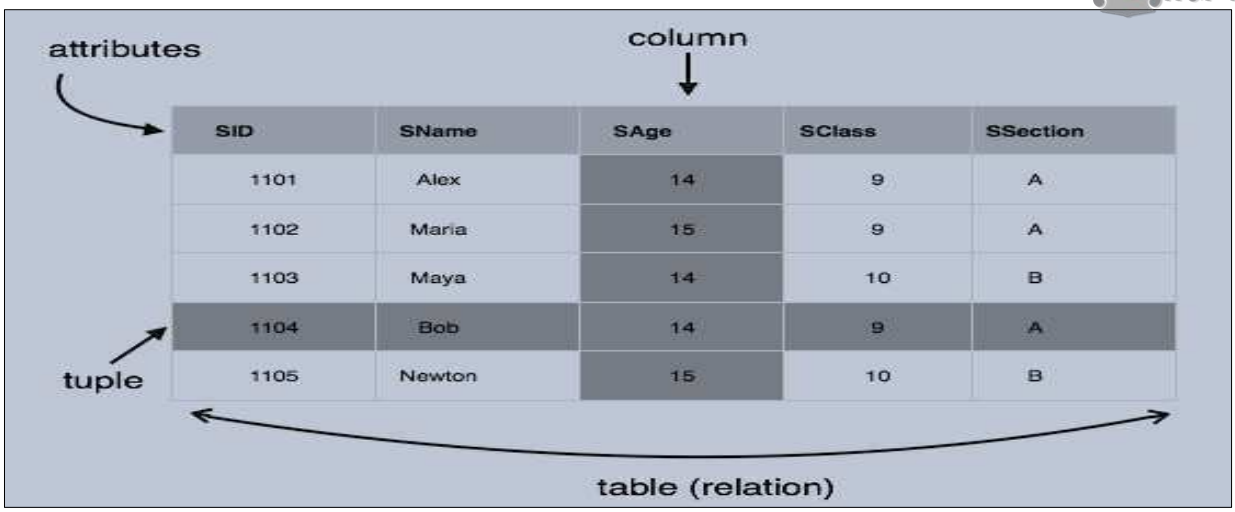
  - one to one
  - one to many
  - many to one
  - many to many

**Relational Model**

The most popular data model in DBMS is the Relational Model. It is more scientific a model than others. This model is based on first-order predicate logic and defines a table as an **n-ary relation**.






**Relational Database Model**

The main highlights of this model are −

- Data is stored in tables called **relations**.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in a relation contains a unique value.
- Each column in a relation contains values from the same domain.
-

**Hierarchical Model**

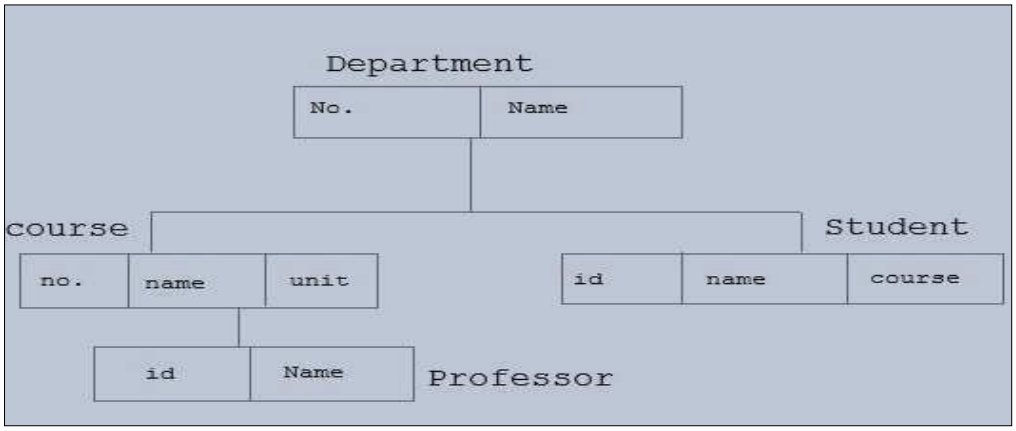
In this model, each entity has only one parent but can have several children. At the top of the hierarchy, there is only one entity which is called **Root**.

Advantages

• Simplicity
• Data Security and Data Integrity
• Efficiency

Disadvantages

• Implementation Complexity
• Lack of structural independence
• Programming complexity




**Hierarchical Model**

**Network Model**

In the network model, entities are organized in a graph, in which some entities can be accessed through several paths.

Advantages

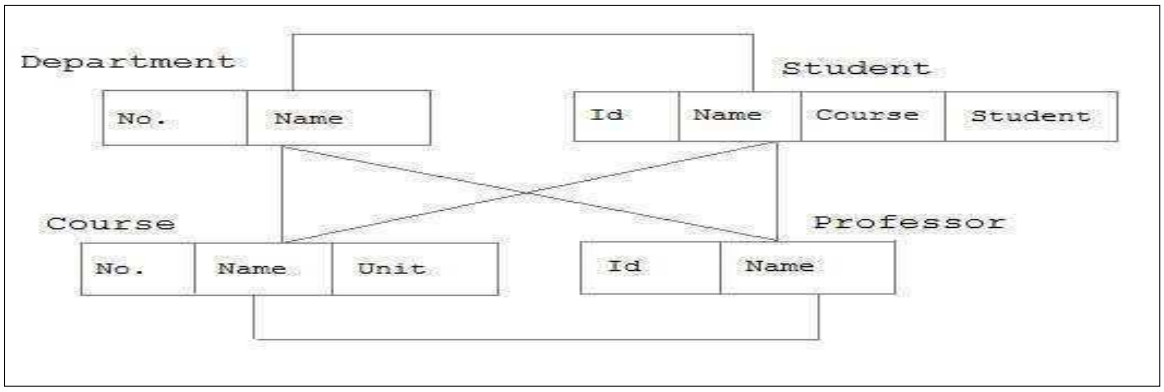


- Conceptual Simplicity
- Ease of data access
- Data Integrity and capability to handle more relationship types
- Data independence
- Database standards

Disadvantages

- System complexity
- Absence of structural independence



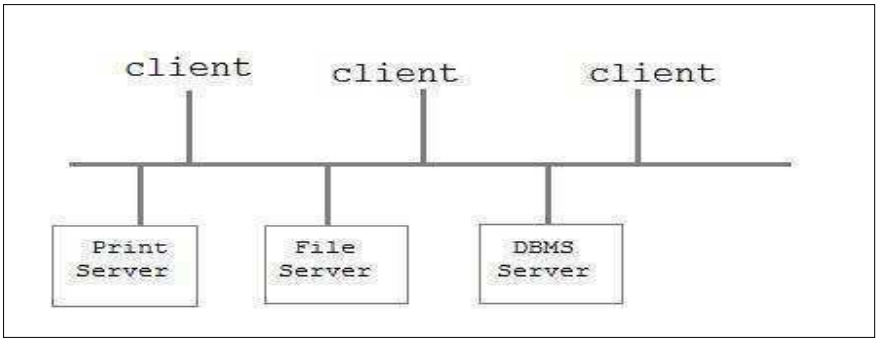

Network Model

**Database Architecture**

Database architecture is logically divided into two types.

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

**Two-tier Client / Server Architecture**



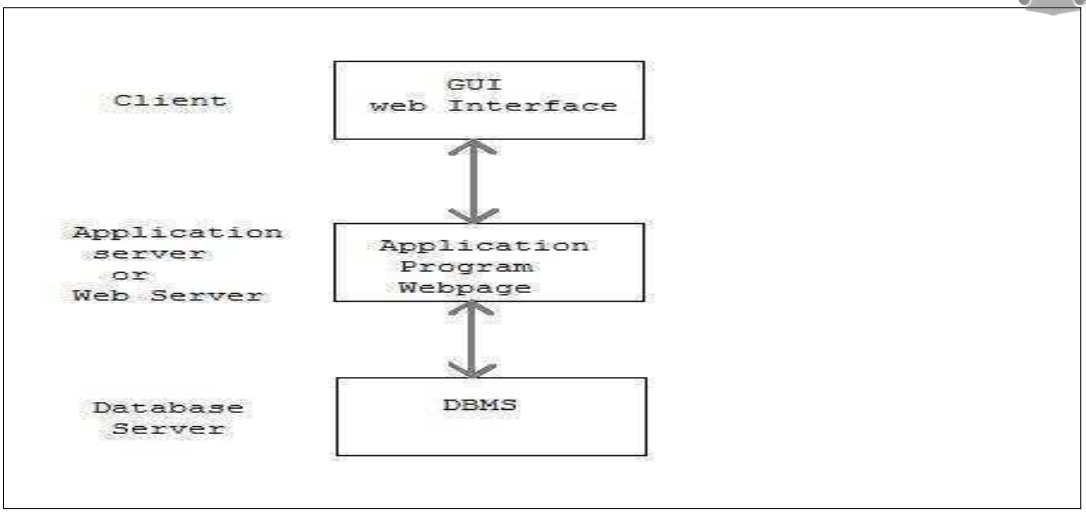

**Two-Tier Architecture**

Two-tier Client / Server architecture is used for User Interface program and Application Programs that run on the client side. An interface called ODBC(Open Database Connectivity) provides an API that allows 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 the 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 Architecture

**Three Tier Architecture**

Three-tier Client / Server database architecture is

commonly used architecture for web applications. An 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 a medium for sending partially processed data between the database server and the client.

**Data Independence**

A major objective for three-level architecture is to provide data independence, which means that upper levels are unaffected by changes in lower levels.

There are two kinds of data independence:

- Logical data independence
- Physical data independence

**Logical Data Independence**

Logical data independence indicates that the conceptual schema can be changed without affecting the existing external schemas. The change would be absorbed by the mapping between the external and conceptual levels. Logical data independence also insulates application programs from operations such as combining two records into one or splitting an existing record into two or more records. This would require a. change in the external/conceptual mapping to leave the external view unchanged.

**Physical Data Independence**

Physical data independence indicates that the physical storage structures or devices could be changed without affecting conceptual schema. The change would be absorbed by the mapping between the conceptual and internal levels.

The Logical data independence is difficult to achieve than physical data independence as it requires the flexibility in the design of database and prograll1iller should foresee the future requirements or modifications in the design.

**DBA (Data Base Administrator)**

**Database Administrator**

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA). The functions of a DBA include:

• Schema definition. The DBA creates the original database schema by executing a set of data definition statements in the DDL.

• Storage structure and access method definition.

• Schema and physical-organization modification. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

• Granting of authorization for data access. By granting different types of authorization, the database administrator can regulate which parts of the database various users can access.

Routine maintenance. Examples of the database administrator's routine maintenance activities are:

– Periodically backing up the database, either onto tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding.
– Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.
– Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users.

**Data Dictionary**

A metadata (also called the data dictionary) 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. It holds the following information about each data element in the databases, it normally includes:

– Name
– Type
– Range of values
– Source
– Access authorization
– Indicates which application programs use the data so that, when a change in a data structure is contemplated, a list of the affected programs can be generated.

The data dictionary is used to control the database operation, data integrity, and accuracy. Metadata is used by developers to develop the programs, queries, controls, and procedures to manage and manipulate the data.

**Active and Passive Data Dictionaries**

The data dictionary may be either active or passive. An active data dictionary (also called integrated data dictionary) is managed automatically by the database management software. Consistent with the current structure and definition of the database. Most of the relational database management systems contain active data dictionaries that can be derived from their system catalog.

The passive data dictionary (also called non-integrated data dictionary) is the one used only for documentation purposes. Data about fields, files, people and so on, in the data processing environment, are. Entered the dictionary and cross-referenced.

**Primary Key**



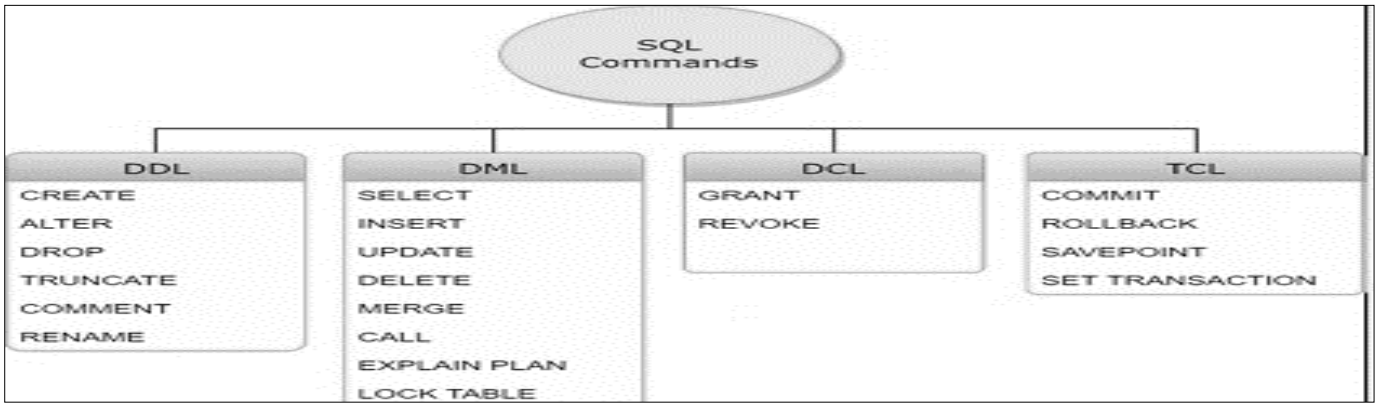
 

A primary key is a special relational database table column (or combination of columns) designated to uniquely identify all table records.

A primary key's main features are:

- It must contain a unique value for each row of data.
- It cannot contain null values.
- A primary key is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

**SQL Commands in DBMS**




**DDL**

DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.



CREATE – to create database and its objects like (table, index, views, stored procedure, function, and triggers)
ALTER – alters the structure of the existing database
DROP – delete objects from the database
TRUNCATE – remove all records from a table, including all spaces allocated for the records are removed
COMMENT – add comments to the data dictionary
RENAME – rename an object

**DML**

DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE etc, and it is used to store, modify, retrieve, delete and update data in the database.

SELECT – retrieve data from a database
INSERT – insert data into a table
UPDATE – updates existing data within a table
DELETE – Delete all records from a database table
MERGE – UPSERT operation (insert or update)
CALL – call a PL/SQL or Java subprogram
EXPLAIN PLAN – interpretation of the data access path
LOCK TABLE – concurrency Control

**Cloud Computing: -**



A cloud is simply a centralized technology platform which provides specific IT services to a selected range of users, offering the ability to log in from anywhere, ideally from any device and over any connection, including the Internet.

Intercept IT believes that a true cloud computing service is one which removes the traditional barriers which exist between software applications, data, and devices. In other words, it is the nirvana of computing from a user's perspective, no need to worry about location, device, or type of connection, all the data and the software applications required by the user are fully available and the experience remains consistent. The highest standards of data protection must be a given, whereby users do not have to think about protecting the integrity of the data they use and store.

**Characteristics of Cloud Computing as per NIST**

Cloud technology is in the news quite often these days, but it still seems to be mysterious and confusing to the non-techie crowd. Cloud options are enticing various industries across the board, which is why it's important to know its essential characteristics as a software offering. Here are the five main characteristics that cloud computing offers businesses today.

**1.On-demand capabilities:**

A business will secure cloud-hosting services through a cloud host provider which could be your usual software vendor. You have access to your services and you have the power to change cloud services through an online control panel or directly with the provider.

**2.Broadnetwork access:**

Your team can access business management solutions using their smart phones, tablets, laptops, and office computers. They can use these devices wherever they are located with a simple online access point.



**3.Resource pooling:**

The cloud enables your employees to enter and use data within the business management software hosted in the cloud at the same time, from any location, and at any time.

**4.Rapid elasticity:**

If anything, the cloud is flexible and scalable to suit your immediate business needs. You can quickly and easily add or remove users, software features, and other resources.

**5.Measured service:**

Going back to the affordable nature of the cloud, you only pay for what you use. You and your cloud provider can measure storage levels, processing, bandwidth, and the number of user accounts and you are billed appropriately.

**Cloud Computing Reference Model**

•The NIST Cloud Computing Reference Architecture consists of five major actors. Each actor plays a role and performs a set of activities and functions. The reference architecture is presented as successive diagrams in increasing level of detail.

•Among the five actors, cloud brokers are optional, as cloud consumers may obtain service directly from a cloud provider.

1. **Cloud Consumer:**

Person or organization that maintains a business relationship with, and uses service from, Cloud Providers.

2. **Cloud Provider:**



A person, organization or entity responsible for making a service available to Cloud Consumers.

3. **Cloud Auditor:**

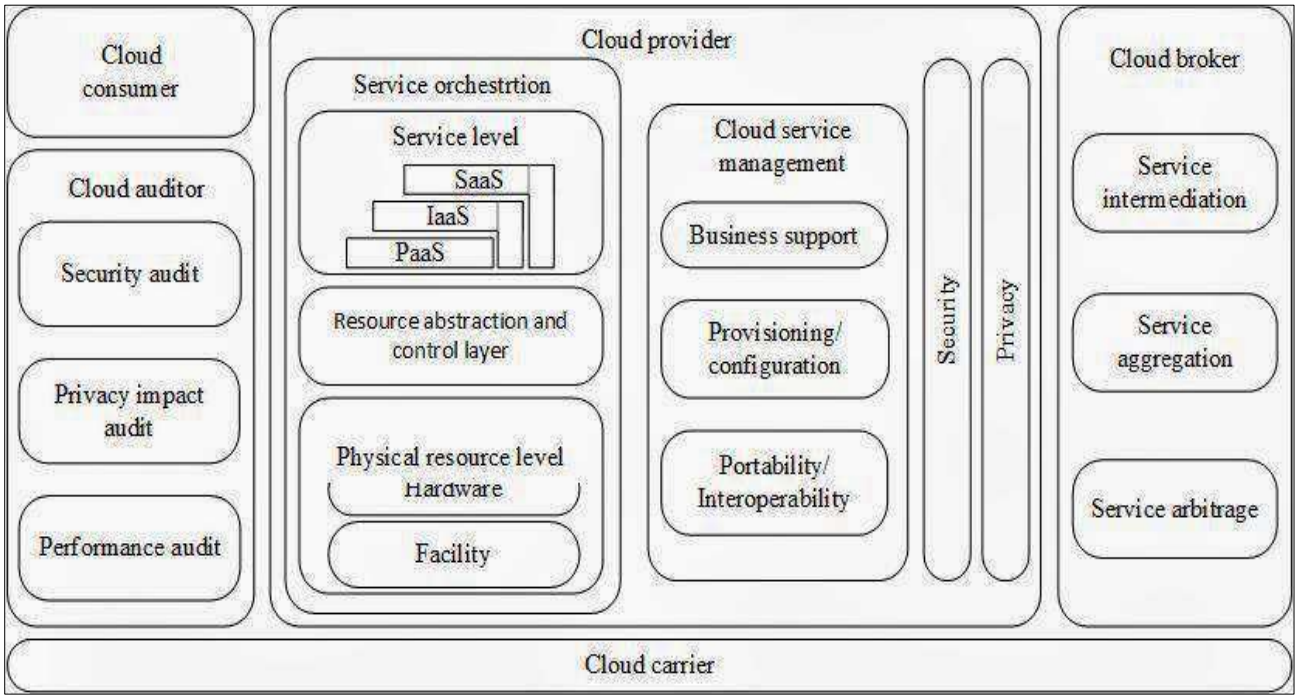
A party that can conduct an independent assessment of cloud services, information system operations, performance and security of the cloud implementation.

4. **Cloud Broker:**

An entity manages the use, performance, and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.

5. **Cloud Carrier:**

The intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.




**Cloud computing service & deployment models**

According to National Institute of Standards and Technology (NIST), Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

**Software as a Service (SaaS).** The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

**Platform as a Service (PaaS).** The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.

**Infrastructure as a Service (IaaS).** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer can deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and



deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

**Deployment Model: -**

**Private cloud**. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

**Community cloud.** The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

**Public cloud.** The cloud infrastructure is provisioned for open use by the public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

**Hybrid cloud.** The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

**Advantages of Cloud Computing**

1. **Usability**: All cloud storage services reviewed in this topic have desktop folders for Mac's and PCs. This allows users to drag and drop files between the cloud storage and their local storage.
2. **Bandwidth**: You can avoid emailing files to individuals and instead send a web link to recipients through your email.
3. **Accessibility**: Stored files can be accessed from anywhere via an Internet connection.
4. **Disaster Recovery**: It is highly recommended that businesses have an emergency backup plan ready in the case of an emergency. Cloud storage can be used as a backup plan by businesses by providing a second copy of important files. These files are stored at a remote location and can be accessed through an internet connection.
5. **Cost Savings:** Businesses and organizations can often reduce annual operating costs by using cloud storage; cloud storage costs about 3 cents per gigabyte to store data internally. Users can see additional cost savings because it does not require internal power to store information remotely.

**Disadvantages of Cloud Computing**

1. **Usability:** Be careful when using drag/drop to move a document into the cloud storage folder. This will permanently move your document from its original folder to the cloud storage location.
2. **Bandwidth**: Several cloud storage services have a specific bandwidth allowance. If an organization surpasses the given allowance, the additional charges could be significant. However, some providers allow unlimited bandwidth. This is a factor that companies should consider when looking at a cloud storage provider.
3. **Accessibility:** If you have no internet connection, you have no access to your data.
4. **Data Security**: There are concerns about the safety and privacy of important data stored remotely. The possibility of private data commingling with other organizations makes some businesses uneasy.
5**. Software**: If you want to be able to manipulate your files locally through multiple devices, you'll need to download the service on all devices.



References: -

1. http://www.sitesbay.com/cpp/cpp-function
2. https://www.tutorialspoint.com/cplusplus/cpp_pointers.htm
3. http://www.cplusplus.com/doc/tutorial/
4. http://www.studytonight.com/cpp/class-and-objects.php






RGPVNOTES.IN

We hope you find these notes useful.

You can get previous year question papers at https://qp.rgpvnotes.in .

If you have any queries or you want to submit your study notes please write us at rgpvnotes.in@gmail.com

LIKE & FOLLOW US ON FACEBOOK
facebook.com/rgpvnotes.in