Chameli Devi Group of Institutions, Indore
Department of ESH
BT205 Basic Computer Engineering
B. Tech, CSE and IT (II Semester)
Unit -5

……

Data base Management System: Introduction, File Oriented Approach and Database Approach, Data Models, Architecture of Database System, Data Independence, Data Dictionary, DBA, Primary Key, Data Definition Language and Manipulation Languages.

Cloud Computing: Definition, Cloud Infrastructure, Cloud Segments or Service Delivery Models (IaaS, PaaS and SaaS), Cloud Deployment Models/ Types of Cloud (Public, Private, Community and Hybrid Clouds), Pros and Cons of Cloud Computing

Unit Objective: To familiarize the students with basic concepts of database management system and cloud computing.

Unit Outcome: Student should be able to explain concepts of DBMS and cloud with their real-world examples.

……..

Data & Information

Data is raw fact or figures or entity. When activities in the organization take place, the effect of these activities needs to be recorded which is known as Data.

Processed data is called information

The purpose of data processing is to generate the information required for carrying out the business activities.

Database

A database is basically a set of data and it contains interrelated data. The database contains set of algorithm and rules through which data can be stored in it in a systematic manner. So, all the data fields must be related to each other. A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world. It is designed built and populated with data for a specific purpose.

A database consists of 4 elements: (i) Data (ii) Relationship (iii) Constraints (iv) Schema

Database Management System

The software used for the management, maintenance and retrieval of the data stored in the database. A DBMS is a collection of interrelated data and a set of programs to access those data.

Collection of database and DBMS software. A database system consists of 4 major components:

(i) Data (ii) Hardware (iii) Software (iv) Database users

DBMS	File System
DBMS is a collection of data and user is not required to write the procedures for managing the database.	File system is a collection of data. Any management with the file system, user has to write the procedures
DBMS provides an abstract view of data that hides the details.	File system gives the details of the data representation and Storage of data.
DBMS is efficient to use since there are wide varieties of sophisticated techniques to store and retrieve the data.	In File system storing and retrieving of data cannot be done efficiently.
DBMS takes care of Concurrent access using some form of locking.	Concurrent access to the data in the file system has many problems like a. Reading the file while other deleting some information, updating some information
DBMS has crash recovery mechanism DBMS protects user from the effects of system failures.	File system doesn't provide crash recovery mechanism. Eg. While we are entering some data into the file if System crashes then content of the file is lost.
DBMS has a good protection mechanism.	Protecting a file under file system is very difficult.

Table 4.1: Difference between DBMS & File System

Architecture of DBMS

The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. An n-tier architecture divides the whole system into related but independent n modules, which can be independently modified, altered, changed, or replaced.

DBMS architecture helps in design, development, implementation, and maintenance of a database. A database stores critical information for a business. Selecting the correct Database Architecture helps in quick and secure access to this data.

Single Tier Architecture

In this architecture, the database is directly available to the user. It means the user can directly communicate with the database and uses it. Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users. The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

Figure 5.1: Single Tier Architecture

Two Tier Architecture

Applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: ODBC, JDBC are used. The user interfaces and application programs are run on the client-side. The server side is responsible to provide the functionalities like: query processing and transaction management. To communicate with the DBMS, client-side application establishes a connection with the server side.

Example: Contact Management System created using MS- Access.

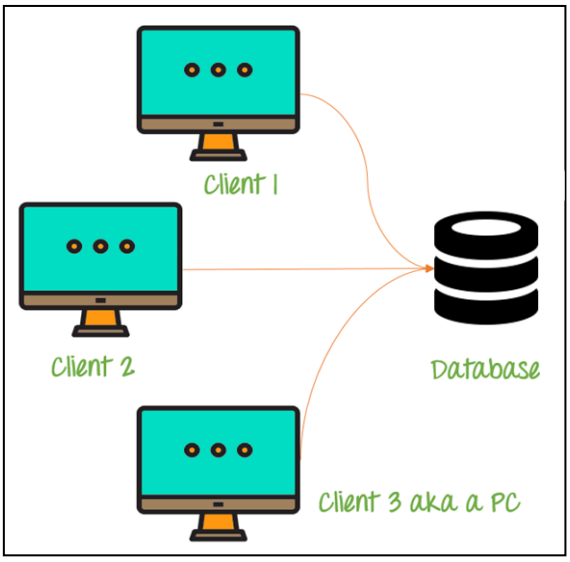

Figure 5.2: Two-Tier Architecture

Three Tier Architecture

The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server. The application on the client-end interacts with an application server which further communicates with the database system. End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.

The goal of Three tier architecture is:

- To separate the user applications and physical database
- Proposed to support DBMS characteristics
- Program-data independence
- Support of multiple views of the data

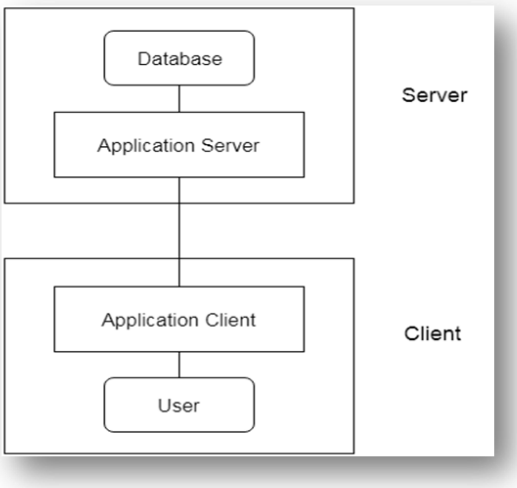

Figure 5.3: Three-Tier Architecture

Database (Data) Tier − The database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.

Application (Middle) Tier − At this tier reside the application server and the programs that access the database. For a user, it presents an abstracted view of the database. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.

User (Presentation) Tier − End-users operate on this tier and they know nothing about any existence of the database beyond this layer. Multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

Database Models

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system.

Relational Model

Data is organized in **two-dimensional tables and the relationship is maintained by storing a common field.** This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model.

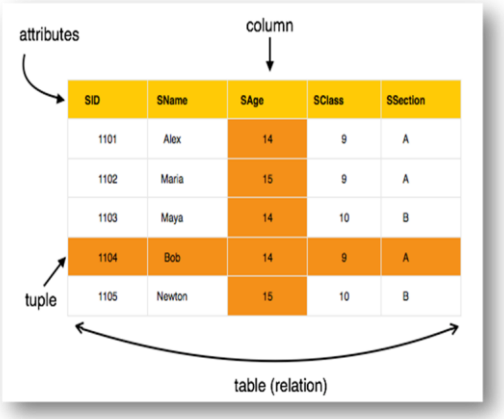

Figure 5.5: Relational Model

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 a same domain.

Hierarchical Model

Organizes data into a tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the Root data, and expands like a tree, adding child nodes to the parent nodes. In this model, a child node will only have a single parent node. This model efficiently describes many real-world relationships like index of a book, recipes etc. In hierarchical model, data is organized into tree-like structure with one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and of course many students.

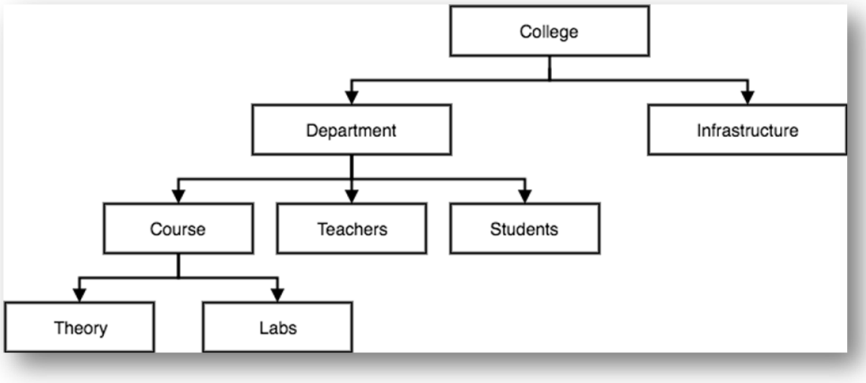

Figure 5.6: Hierarchical Model

Network Model

The network model builds on the hierarchical model by allowing many-to-many relationships between linked records, implying multiple parent records. Based on mathematical set theory, the model is constructed with sets of related records. Each set consists of one owner or parent record and one or more member or child records. A record can be a member or child in multiple sets, allowing this model to convey complex relationships.

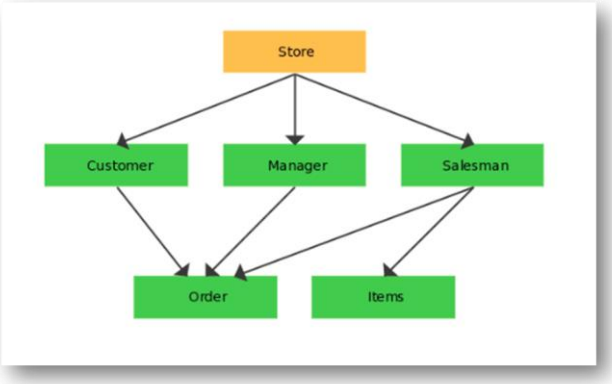

Figure 5.7: Network Model

Entity Relational 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.

This model captures the relationships between real-world entities much like the network model, but it isn't as directly tied to the physical structure of the database. Instead, it's often used for designing a database conceptually.

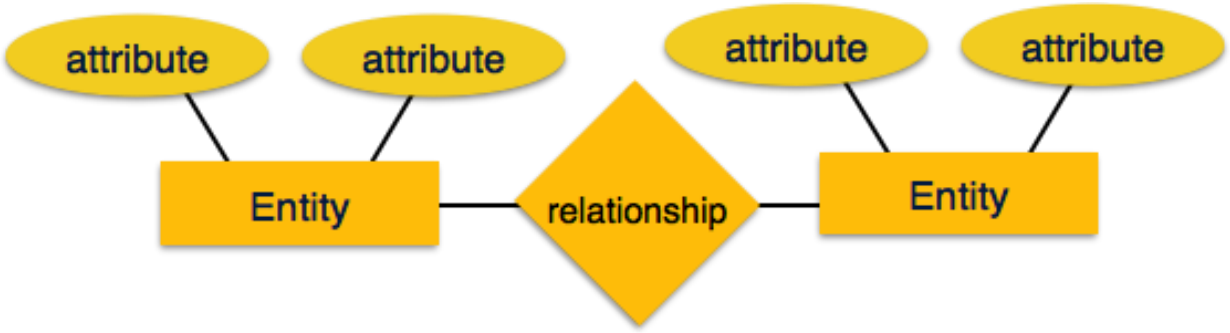

Figure 5.8: E-R Model

Data Independence

Need of Data Independence:

- A database system normally contains a lot of data in addition to users' data.
- For example, it stores data about data, known as metadata, to locate and retrieve data easily.
- It is rather difficult to modify or update a set of metadata once it is stored in the database.
- But as DBMS expands, it needs to change over time to satisfy the requirements of the users.
- If the entire data is dependent, it would become a tedious and highly complex job.

Data Independence is defined as a property of DBMS that helps you to change the Database schema at one level of a database system without requiring to change the schema at the next higher level. Data independence helps you to keep data separated from all programs that make use of it. You can use this stored data for computing and presentation. In many systems, data independence is an essential function for components of the system.

Types of Data Independence

Physical Data Independence

Helps to separate conceptual levels from the internal/physical levels. It allows user to provide a logical description of the database without the need to specify physical structures. User can easily change the physical storage structures or devices with an effect on the conceptual schema. Any change done would be absorbed by the mapping between the conceptual and internal levels.

Examples of changes under Physical Data Independence

- Using a new storage device like Hard Drive or Magnetic Tapes
- Modifying the file organization technique in the Database
- Switching to different data structures

- Changing the access method
- Modifying indexes
- Changes to compression techniques or hashing algorithms
- Change of Location of Database from say C drive to D Drive

Logical Data Independence

Logical Data Independence is the ability to change the conceptual scheme without changing

- External views
- External API or programs

Any change made will be absorbed by the mapping between external and conceptual levels.

Examples of changes under Logical Data Independence

- Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs
- Merging two records into one
- Breaking an existing record into two or more records

Logical Data Independence	**Physical Data Independence**
Logical Data Independence is mainly concerned with the structure or changing the data definition.	Mainly concerned with the storage of the data.
It is difficult as the retrieving of data is mainly dependent on the logical structure of data.	It is easy to retrieve.
Compared to Physical independence it is difficult to achieve logical data independence.	Compared to Logical Independence it is easy to achieve physical data independence.
You need to make changes in the Application program if new fields are added or deleted from the database.	A change in the physical level usually does not need change at the Application program level.
Modification at the logical levels is significant whenever the logical structures of the database are changed.	Modifications made at the internal levels may or may not be needed to improve the performance of the structure.
Concerned with conceptual schema	Concerned with internal schema
Example: Add/Modify/Delete a new attribute	Example : change in compression techniques, hashing algorithms, storage devices, etc.

Table 5.2: Difference between Logical & Physical Data Independence

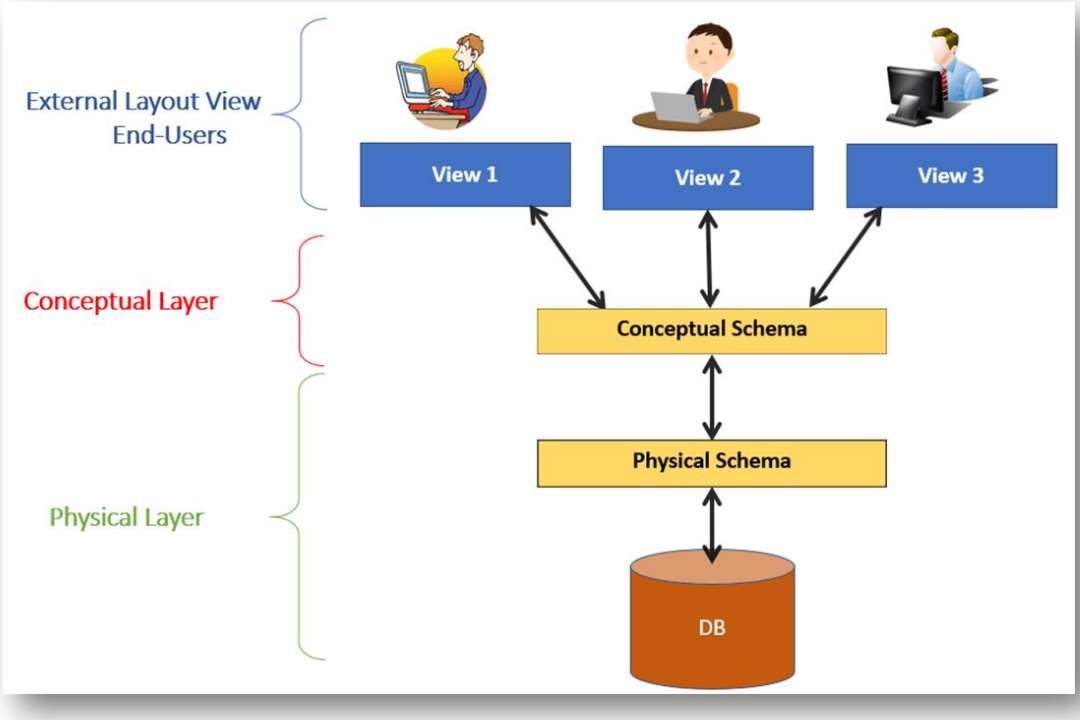

Figure 5.9: Data Independence

Type of Schema	Implementation
External Schema	**View 1**: Course info(cid:int,cname:string) **View 2**: studeninfo(id:int. name:string)

Conceptual Shema	`Students(id: int, name: string, login: string, age: integer)`
	`Courses(id: int, cname.string, credits:integer)`
	`Enrolled(id: int, grade:string)`

Physical Schema	• Relations stored as unordered files. • Index on the first column of Students.

Figure 5.10: Example of Data Independence

Database Administrator

A database administrator (DBA) is a specialized computer systems administrator who maintains a successful database environment by directing or performing all related activities to keep the data secure. The top responsibility of a DBA professional is to maintain data integrity. This means the DBA will ensure that data is secure from unauthorized access but is available to users.

Keys in Relational Model

Primary Key: A column or group of columns in a table which helps us to uniquely identifies every row in that table is called a primary key. Primary key can't be a duplicate. It is must for every row to have a primary key value. The primary key field cannot be null. The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

Foreign Key: A foreign key is a column which is added to create a relationship with another table. Foreign keys help us to maintain data integrity and also allows navigation between two different instances of an entity. Every relationship in the model needs to be supported by a foreign key.

SQL

Structured Query Language is a standard Database language which is used to **create, maintain and retrieve the relational database**. SQL is case insensitive. But it is a recommended practice to use keywords (like SELECT, UPDATE, CREATE, etc) in capital letters and use user defined things (liked table name, column name, etc) in small letters. We can write comments in SQL using "- -" (double hyphen) at the beginning of any line. SQL became a standard of the **American National Standards Institute (ANSI) in 1986**, and of the **International Organization for Standardization (ISO) in 1987**

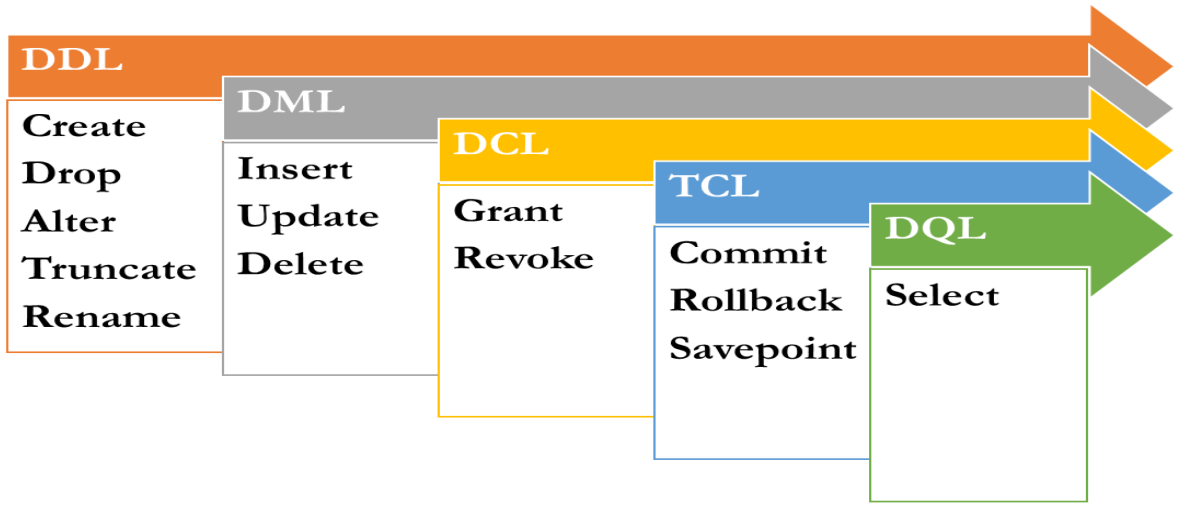

Figure 5.11: SQL Classification

DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc. All the command of DDL is auto-committed that means it permanently save all the changes in the database.

DML commands are used to modify the database. It is responsible for all form of changes in the database. The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

DCL commands are used to grant and take back authority from any database user.

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only. These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

DDL Commands

CREATE: It is used to create a new table in the database.

Syntax: CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);

Example: CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE);

DROP: It is used to delete both the structure and record stored in the table.

Syntax: DROP TABLE;

Example: DROP TABLE EMPLOYEE;

TRUNCATE: It is used to delete all the rows from the table and free the space containing the table

Syntax: TRUNCATE TABLE table_name;

Example: TRUNCATE TABLE EMPLOYEE;

ALTER: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

Syntax:

To add a new column in the table: (ALTER TABLE table_name ADD column_name COLUMN-definition;)

To modify existing column in the table: (ALTER TABLE MODIFY(COLUMN DEFINITION....);

Example:

ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));

ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));

DML Commands

INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax: INSERT INTO TABLE_NAME (col1, col2, col3,.... col N) VALUES (value1, value2, value3, ... valueN);

INSERT INTO TABLE_NAME

VALUES (value1, value2, value3, ... valueN);

Example: INSERT INTO student (Rollno, Subject) VALUES (“12345", "DBMS");

DML Commands

UPDATE: This command is used to update or modify the value of a column in the table.

Syntax: UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [WHERE CONDITION]

Example: UPDATE students SET User_Name = 'Mick' WHERE Student_Id = '3'

DELETE: It is used to remove one or more row from a table.

Syntax: DELETE FROM table_name [WHERE condition];

Example: DELETE FROM student WHERE Rollno="2569";

Cloud Computing

The term cloud refers to a **network or the internet**. It is a technology that uses remote servers on the internet to store, manage, and access data online rather than local drives. The data can be anything such as files, images, documents, audio, video, and more.

Cloud Computing: The term "Cloud Computing" generally refers to the ability of a system to store data or applications on remote servers, process data or applications from servers, and access data and applications via the Internet. Cloud computing provides scalability, flexibility, cost-effectiveness, and security to individuals and organizations to manage their IT operations. Cloud computing works on a Pay-on-Use basis for individuals and organizations. It is an on-demand availability of system resources and computing power without direct active management by the user.

Cloud Services Models

Infrastructure as a Service (IaaS): IaaS is also known as Hardware as a Service (HaaS). It is a computing infrastructure managed over the internet. The main advantage of using IaaS is that it helps users to avoid the cost and complexity of purchasing and managing the physical servers.

There are the following characteristics of IaaS:

- Resources are available as a service
- Services are highly scalable
- Dynamic and flexible
- GUI and API-based access
- Automated administrative tasks

IaaS provider provides the following services:

Compute: Computing as a Service includes virtual central processing units and virtual main memory for the Vms that is provisioned to the end- users.

Storage: IaaS provider provides back-end storage for storing files.

Network: Network as a Service (NaaS) provides networking components such as routers, switches, and bridges for the Vms.

Load balancers: It provides load balancing capability at the infrastructure layer.

Example: DigitalOcean, Linode, Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE), Rackspace, and Cisco Metacloud.

Platform as a Service (PaaS): Platform as a Service (PaaS) provides a runtime environment. It allows programmers to easily create, test, run, and deploy web applications. You can purchase these applications from a cloud service provider on a pay-as-per use basis and access them using the Internet connection. In PaaS, back-end scalability is managed by the cloud service provider, so end- users do not need to worry about managing the infrastructure. PaaS includes infrastructure (servers, storage, and networking) and platform (middleware, development tools, database management systems, business intelligence, and more) to support the web application life cycle.

Example: Google App Engine, Force.com, Joyent, Azure.

PaaS providers provide the Programming languages, Application frameworks, Databases, and other tools:

Programming languages: PaaS providers provide various programming languages for the developers to develop the applications. Some popular programming languages provided by PaaS providers are Java, PHP, Ruby, Perl, and Go.

Application frameworks: PaaS providers provide application frameworks to easily understand the application development. Some popular application frameworks provided by PaaS providers are Node.js, Drupal, Joomla, WordPress, Spring, Play, Rack, and Zend.

Databases: PaaS providers provide various databases such as ClearDB, PostgreSQL, MongoDB, and Redis to communicate with the applications.

Other tools: PaaS providers provide various other tools that are required to develop, test, and deploy the applications.

Software as a Service (SaaS): SaaS is also known as "On-Demand Software". It is a software distribution model in which services are hosted by a cloud service provider. These services are available to end-users over the internet so, the end-users do not need to install any software on their devices to access these services.

There are the following services provided by SaaS providers –

Business Services: SaaS Provider provides various business services to start-up the business. The SaaS business services include ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), billing, and sales.

Document Management: SaaS document management is a software application offered by a third party (SaaS providers) to create, manage, and track electronic documents.

Example: Slack, Samepage, Box, and Zoho Forms.

Social Networks: As we all know, social networking sites are used by the general public, so social networking service providers use SaaS for their convenience and handle the general public's information.

Mail Services: To handle the unpredictable number of users and load on e-mail services, many e-mail providers offering their services using SaaS.

Differences among Models

IaaS	Paas	SaaS
It provides a virtual data center to store information and create platforms for app development, testing, and deployment.	It provides virtual platforms and tools to create, test, and deploy apps.	It provides web software and apps to complete business tasks.
It provides access to resources such as virtual machines, virtual storage, etc.	It provides runtime environments and deployment tools for applications.	It provides software as a service to the end-users.
It is used by network architects.	It is used by developers.	It is used by end users.
IaaS provides only Infrastructure.	PaaS provides Infrastructure + Platform.	SaaS provides Infrastructure + Platform + Software.

Table 5.3: Difference among Cloud Models

Types of Cloud

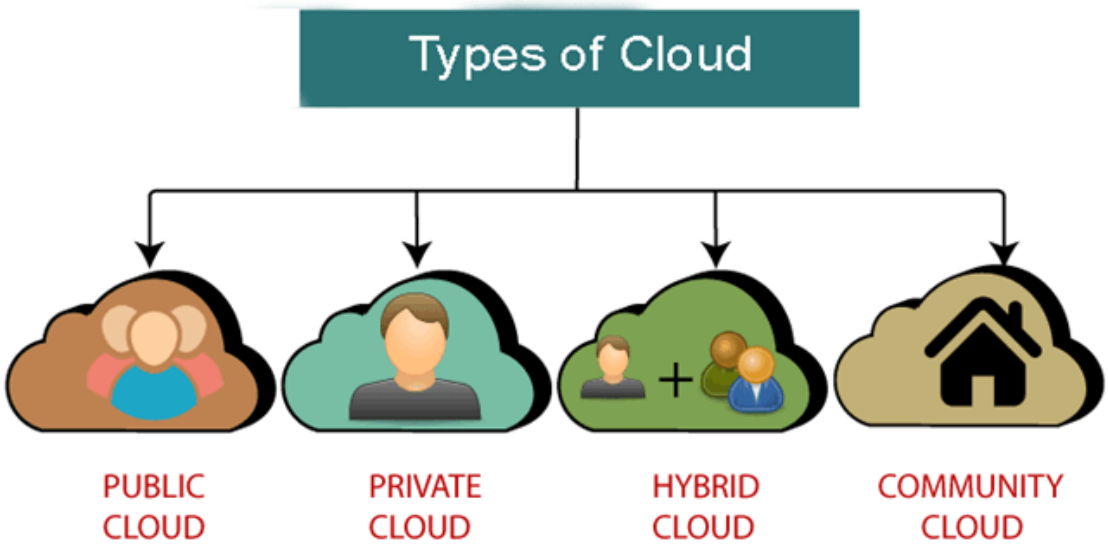

Figure 5.12: Types of Cloud

Public Cloud

Public cloud is open to all to store and access information via the Internet using the pay-per-usage method. In public cloud, computing resources are managed and operated by the Cloud Service Provider (CSP).

Example: Amazon elastic compute cloud (EC2), IBM SmartCloud Enterprise, Microsoft, Google App Engine, Windows Azure Services Platform.

Advantages:

- Public cloud is owned at a lower cost than the private and hybrid cloud.
- Public cloud is maintained by the cloud service provider, so do not need to worry about the maintenance.
- Public cloud is easier to integrate. Hence it offers a better flexibility approach to consumers.
- Public cloud is location independent because its services are delivered through the internet.
- Public cloud is highly scalable as per the requirement of computing resources.
- It is accessible by the general public, so there is no limit to the number of users.

Disadvantages:

- Public Cloud is less secure because resources are shared publicly.
- Performance depends upon the high-speed internet network link to the cloud provider.
- The Client has no control of data.

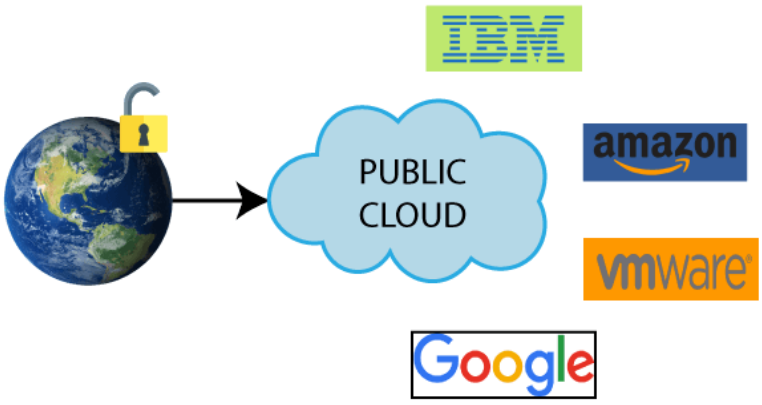

Figure 5.13: Public Cloud

Private Cloud

Private cloud is also known as an internal cloud or corporate cloud. It is used by organizations to build and manage their own data centers internally or by the third party. It can be deployed using Opensource tools such as Openstack and Eucalyptus. Based on the location and management, National Institute of Standards and Technology (NIST) divide private cloud into the following two parts-

- On-premise private cloud
- Outsourced private cloud

Advantages:

- Private cloud provides a high level of security and privacy to the users.
- Private cloud offers better performance with improved speed and space capacity.
- It allows the IT team to quickly allocate and deliver on-demand IT resources.
- The organization has full control over the cloud because it is managed by the organization itself. So, there is no need for the organization to depends on anybody.
- It is suitable for organizations that require a separate cloud for their personal use and data security is the first priority.

Disadvantages:

- Skilled people are required to manage and operate cloud services.
- Private cloud is accessible within the organization, so the area of operations is limited.

- Private cloud is not suitable for organizations that have a high user base, and organizations that do not have the prebuilt infrastructure, sufficient manpower to maintain and manage the cloud.

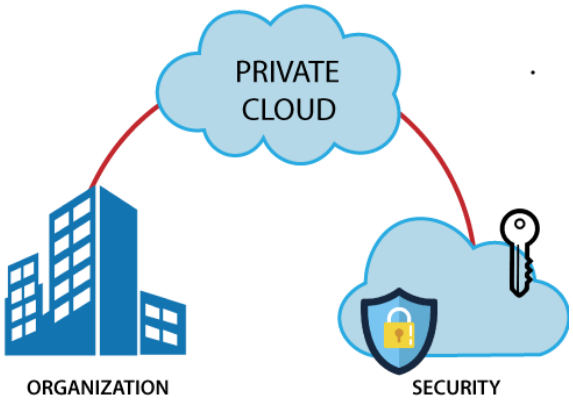

Figure 5.14: Private Cloud

Hybrid Cloud

Hybrid Cloud is a combination of the public cloud and the private cloud. we can say:

Hybrid Cloud = Public Cloud + Private Cloud

Hybrid cloud is partially secure because the services which are running on the public cloud can be accessed by anyone, while the services which are running on a private cloud can be accessed only by the organization's users.

Example: Google Application Suite (Gmail, Google Apps, and Google Drive), Office 365 (MS Office on the Web and One Drive), Amazon Web Services.

Advantages:

- Hybrid cloud is suitable for organizations that require more security than the public cloud.
- Hybrid cloud helps you to deliver new products and services more quickly.
- Hybrid cloud provides an excellent way to reduce the risk.
- Hybrid cloud offers flexible resources because of the public cloud and secure resources because of the private cloud.

Disadvantages:

- In Hybrid Cloud, security feature is not as good as the private cloud.
- Managing a hybrid cloud is complex because it is difficult to manage more than one type of deployment model.
- In the hybrid cloud, the reliability of the services depends on cloud service providers.

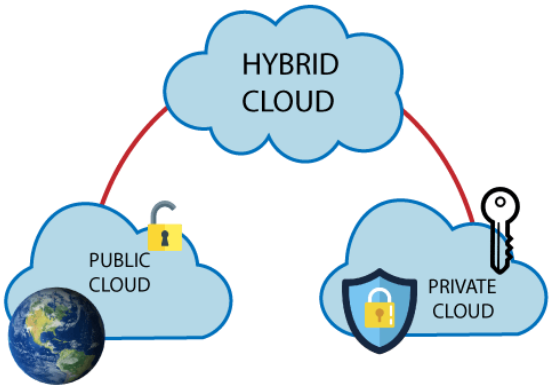

Figure 5.15: Hybrid Cloud

Community Cloud

Community cloud allows systems and services to be accessible by a group of several organizations to share the information between the organization and a specific community. It is owned, managed, and operated by one or more organizations in the community, a third party, or a combination of them.

Example: Health Care community cloud

Advantages:

- Community cloud is cost-effective because the whole cloud is being shared by several organizations or communities.
- Community cloud is suitable for organizations that want to have a collaborative cloud with more security features than the public cloud.
- It provides better security than the public cloud.
- It provides collaborative and distributive environment.
- Community cloud allows us to share cloud resources, infrastructure, and other capabilities among various organizations.

Disadvantages:

- Community cloud is not a good choice for every organization.
- Security features are not as good as the private cloud.
- It is not suitable if there is no collaboration.
- The fixed amount of data storage and bandwidth is shared among all community members.

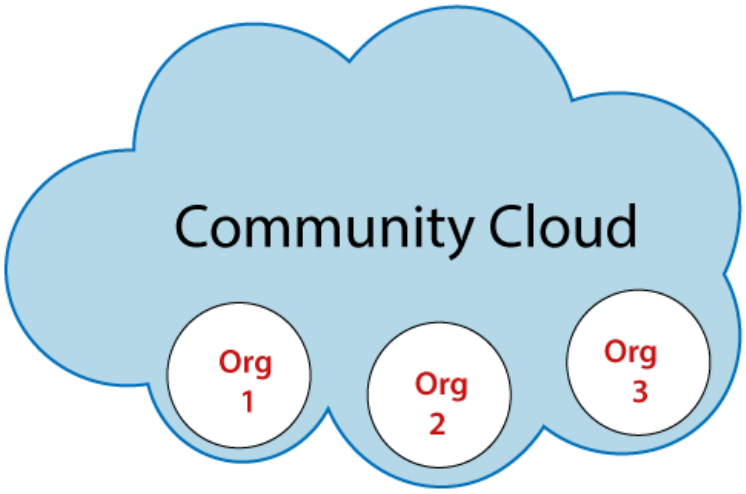

Figure 5.16: Community Cloud

Differences among types of cloud

Parameter	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud
Host	Service provider	Enterprise (Third party)	Enterprise (Third party)	Community (Third party)
Users	General public	Selected users	Selected users	Community members
Access	Internet	Internet, VPN	Internet, VPN	Internet, VPN
Owner	Service provider	Enterprise	Enterprise	Community

Table 5.4: Difference among cloud types

Advantages of Cloud Computing

Back-up and restore data: Once the data is stored in the cloud, it is easier to get back-up and restore that data using the cloud.

Improved collaboration: Cloud applications improve collaboration by allowing groups of people to quickly and easily share information in the cloud via shared storage.

Excellent accessibility: Cloud allows us to quickly and easily access store information anywhere, anytime in the whole world, using an internet connection. An internet cloud infrastructure increases organization productivity and efficiency by ensuring that our data is always accessible.

Low maintenance cost: Cloud computing reduces both hardware and software maintenance costs for organizations.

Mobility: Cloud computing allows us to easily access all cloud data via mobile.

IServices in the pay-per-use model: Cloud computing offers Application Programming Interfaces (APIs) to the users for access services on the cloud and pays the charges as per the usage of service.

Unlimited storage capacity: Cloud offers us a huge amount of storing capacity for storing our important data such as documents, images, audio, video, etc. in one place.

Data security: Data security is one of the biggest advantages of cloud computing. Cloud offers many advanced features related to security and ensures that data is securely stored and handled.

Disadvantages of Cloud Computing

Internet Connectivity: As you know, in cloud computing, every data (image, audio, video, etc.) is stored on the cloud, and we access these data through the cloud by using the internet connection.

If you do not have good internet connectivity, you cannot access these data. However, we have no any other way to access data from the cloud.

Vendor lock-in: Vendor lock-in is the biggest disadvantage of cloud computing.

Organizations may face problems when transferring their services from one vendor to another.

As different vendors provide different platforms, that can cause difficulty moving from one cloud to another.

Limited Control: As we know, cloud infrastructure is completely owned, managed, and monitored by the service provider, so the cloud users have less control over the function and execution of services within a cloud infrastructure.

Security: Although cloud service providers implement the best security standards to store important information.

But, before adopting cloud technology, you should be aware that you will be sending all your organization's sensitive information to a third party, i.e., a cloud computing service provider.

While sending the data on the cloud, there may be a chance that your organization's information is hacked by Hackers.