

Unit 4

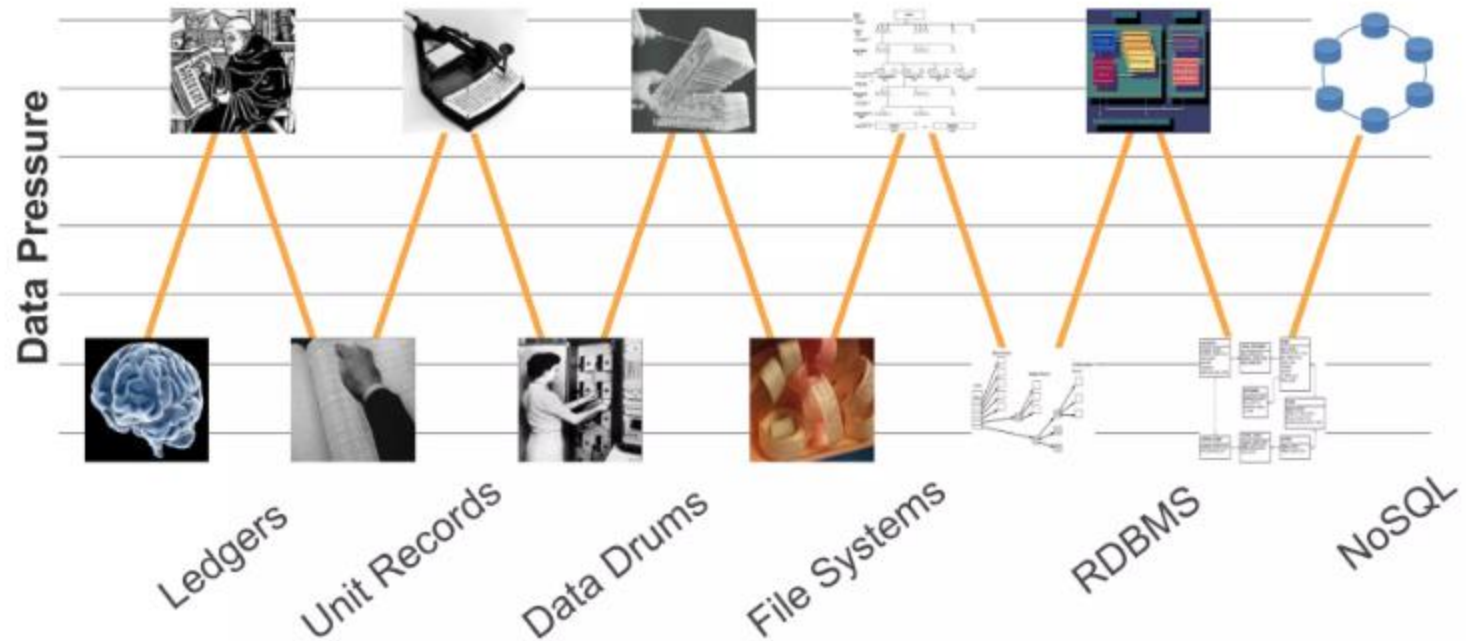
Data Storage and Security in Cloud

Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo Cloud data stores: Datastore and Simple DB, Cloud Storage-Overview, Cloud Storage Providers.

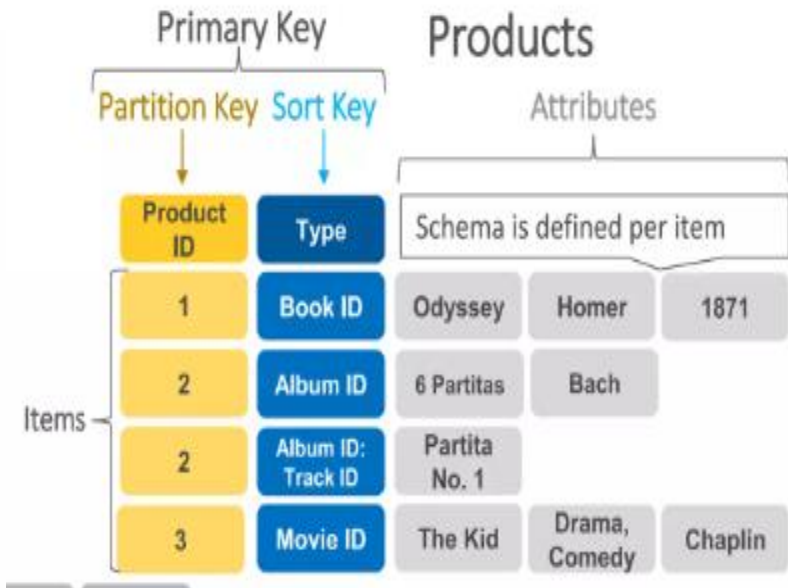
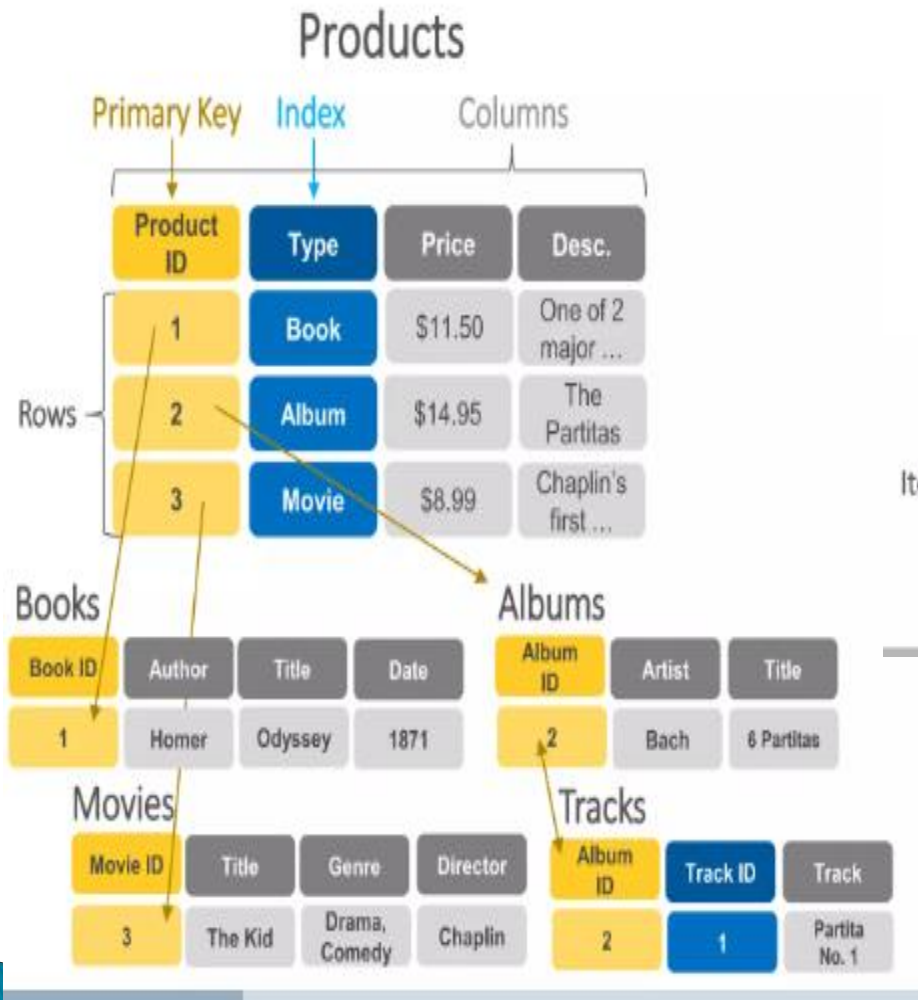
Securing the Cloud- General Security Advantages of Cloud-Based Solutions, Introducing Business Continuity and Disaster Recovery. Disaster Recovery- Understanding the Threats.

Dynamo DB

Timeline of database technology



SQL vs. NoSQL



SQL vs. NoSQL



SQL vs. NoSQL

- ▶ What is SQL?
 - Structured Query Language (SQL) databases store data in structured tables with predefined schema.
- ▶ What is NoSQL?
 - NoSQL databases store unstructured or semi-structured data and support flexible schema designs
- ▶ SQL relational (Normalized) databases are **optimized for storage** while NoSQL is **optimized for Compute** (De-normalized).
- ▶ SQL **scale vertically** while NoSQL **scale horizontally**.

SQL Databases - Characteristics

- ▶ Uses structured tables with relationships.
- ▶ Ensures data integrity through ACID properties.
- ▶ Examples: MySQL, PostgreSQL, Oracle, SQL Server.
- ▶ Best for applications requiring strong consistency (e.g., banking, ERP systems)

NoSQL Databases - Characteristics

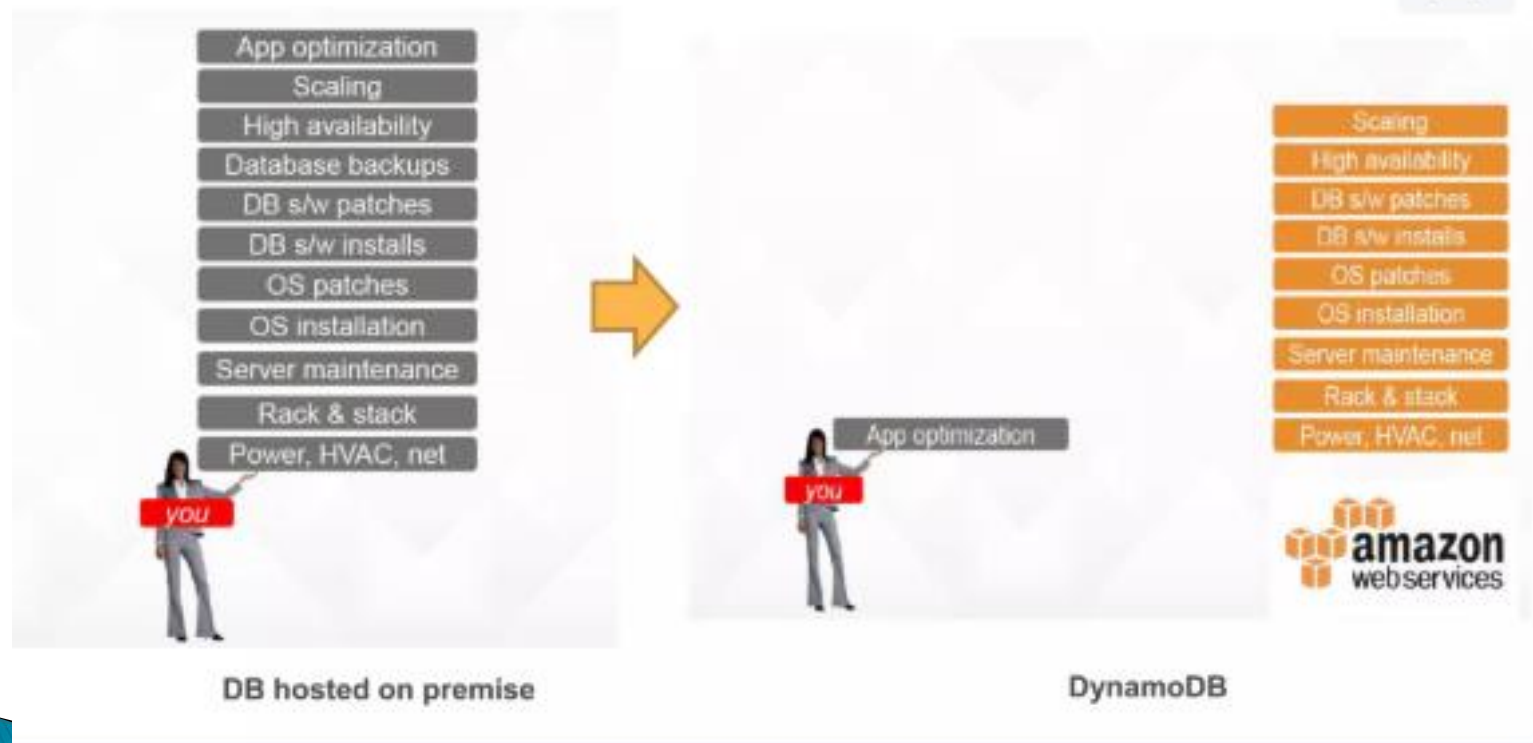
- ▶ Flexible data storage (JSON, XML, key-value pairs, graphs, etc.).
- ▶ Supports high availability and horizontal scaling.
- ▶ Examples: MongoDB, Cassandra, Redis, Neo4j.
- ▶ Best for big data, real-time applications, and distributed systems.

SQL vs. NoSQL

NOSQL VS MYSQL

- Data:
 - NoSQL:
 - Offers flexibility as not every record needs to store the same properties.
 - New properties can be added on the fly.
 - Good for semi structure, complex and nested data.
 - Relation captured by denormalizing data and presenting data in single object in a single record.
 - SQL:
 - Used where the solution for every record has same property.
 - Adding properties may require altering schema or backfilling of data.
 - Good for structured data.
 - Relations are captured in normalized model using joins to resolve reference across the tables.

Fully managed service = automated operations



DynamoDB – Introduction

- ▶ **NoSQL database** service that is offered by **Amazon.com**
- ▶ Amazon released DynamoDb on January 18, 2012.
- ▶ Supports both **document and key-value** store models.
- ▶ DynamoDB allows users to create databases capable of storing and retrieving any amount of data and comes in handy while serving any amount of traffic.
- ▶ It dynamically manages each customer's requests and provides high performance by automatically distributing data and traffic over servers.
- ▶ It is a fully managed NoSQL database service that is fast, predictable in terms of performance, and seamlessly scalable.

DynamoDB – Introduction

- ▶ Amazon DynamoDB is a fully managed NoSQL database service that lets you offload the administrative burdens of operating and scaling a distributed database.
- ▶ It relieves the user from the administrative burdens of operating and scaling a distributed database as the user doesn't have to worry about hardware provisioning, patching Softwares, or cluster scaling.
- ▶ It also eliminates the operational burden and complexity involved in protecting sensitive data by providing encryption at REST.

Advantages of DynamoDB

- ▶ It has fast and predictable performance.
- ▶ It is highly scalable.
- ▶ It offloads the administrative burden operation and scaling.
- ▶ It offers encryption at REST for data protection.
- ▶ Its scalability is highly flexible.
- ▶ AWS Management Console can be used to monitor resource utilization and performance metrics.
- ▶ It provides on-demand backups.
- ▶ It enables point-in-time recovery for your Amazon DynamoDB tables.
- ▶ It can be highly automated.

Limitations of DynamoDB

- ▶ It has a low read capacity unit of 4kB per second and a write capacity unit of 1KB per second.
- ▶ All tables and global secondary indexes must have a minimum of one read and one write capacity unit. Maximums depend on region.
- ▶ Table sizes have no limits, but accounts have a 256 table limit unless you request a higher cap.
- ▶ Only Five local and five global secondary indexes per table are permitted.
- ▶ DynamoDB does not prevent the use of reserved words as names.
- ▶ Partition key length and value minimum length sits at 1 byte, and maximum at 2048 bytes, however, DynamoDB places no limit on values.

Usability

- ▶ Supports various platforms (.NET, PHP, Java, Python, Ruby, etc)
- ▶ The pricing is very simple
- ▶ The data model is very flexible, use of JSON
- ▶ Auto Scaling
- ▶ DAX, in-memory cache that can reduce DynamoDB response times from milliseconds to microseconds making its usability very easy
- ▶ Fully managed cloud database

DynamoDB Components

- ▶ **Tables** - Similar to other database systems, DynamoDB stores data in tables. A table is a collection of data.
- ▶ **Items** - Each table contains zero or more items. An item is a group of attributes that is uniquely identifiable among all of the other items. In DynamoDB, there is no limit to the number of items you can store in a table. Items are like rows in a relational database.
- ▶ **Attributes** - Each item is composed of one or more attributes. An attribute is a fundamental data element, something that does not need to be broken down any further. Attributes in DynamoDB are similar in many ways to fields or columns in other database systems.

DynamoDB Components

Table

id = 100	date = 2012-05-16-09-00-10	total = 25.00
id = 101	date = 2012-05-15-15-00-11	total = 35.00
id = 101	date = 2012-05-16-12-00-10	total = 100.00
id = 102	date = 2012-03-20-18-23-10	total = 20.00
id = 102	date = 2012-03-20-18-23-10	total = 120.00

Item

id = 100	date = 2012-05-16-09-00-10	total = 25.00
id = 101	date = 2012-05-15-15-00-11	total = 35.00
id = 101	date = 2012-05-16-12-00-10	total = 100.00
id = 102	date = 2012-03-20-18-23-10	total = 20.00
id = 102	date = 2012-03-20-18-23-10	total = 120.00

Attribute

id = 100	date = 2012-05-16-09-00-10	total = 25.00
id = 101	date = 2012-05-15-15-00-11	total = 35.00
id = 101	date = 2012-05-16-12-00-10	total = 100.00
id = 102	date = 2012-03-20-18-23-10	total = 20.00
id = 102	date = 2012-03-20-18-23-10	total = 120.00

DynamoDB Structure

- ▶ Each item in the table has a unique identifier, or primary key, that distinguishes the item from all of the others in the table.
- ▶ Other than the primary key, a table is **schemaless**, which means that neither the attributes nor their data types need to be defined beforehand. Each item can have its own distinct attributes.
- ▶ Most of the attributes are scalar, which means that they can have **only one value**. Strings and numbers are common examples of scalars.
- ▶ Some of the items have a **nested attribute (Address)**.
- ▶ DynamoDB supports nested attributes up to 32 levels deep.

DynamoDB vs. RDBMS

Common Tasks	RDBMS	DynamoDB
Connect to the Source	It uses a persistent connection and SQL commands.	It uses HTTP requests and API operations
Create a Table	Its fundamental structures are tables, and must be defined.	It only uses primary keys, and no schema on creation. It uses various data sources.
Get Table Info	All table info remains accessible	Only primary keys are revealed.
Load Table Data	It uses rows made of columns.	In tables, it uses items made of attributes
Read Table Data	It uses SELECT statements and filtering statements.	It uses GetItem, Query, and Scan.
Manage Indexes	It uses standard indexes created through SQL statements. Modifications to it occur automatically on table changes.	It uses a secondary index to achieve the same function. It requires specifications (partition key and sort key).
Modify Table Data	It uses an UPDATE statement.	It uses an UpdateItem operation.
Delete Table Data	It uses a DELETE statement.	It uses a DeleteItem operation.
Delete a Table	It uses a DROP TABLE statement.	It uses a DeleteTable operation.

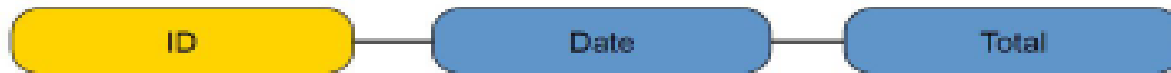
DynamoDB Structure

- ▶ **Partition Key** /hash key – This simple primary key consists of a single attribute referred to as the “partition key.” -determine the partition in which the item will be stored.
- ▶ **Partition Key and Sort Key** – This key, known as the “Composite Primary Key”, consists of two attributes.
 - The partition key and
 - The sort key/Range Key–Allows multiple items to share the same partition and defines how data is ordered and queried within that partition.

```
CREATE TABLE Orders (  
    UserID STRING,      -- Partition Key  
    OrderDate STRING,   -- Sort Key  
    OrderAmount DECIMAL,  
    PRIMARY KEY (UserID, OrderDate)  
);
```

This allows you to store multiple orders for the same user, each identified by a unique combination of UserID and OrderDate.

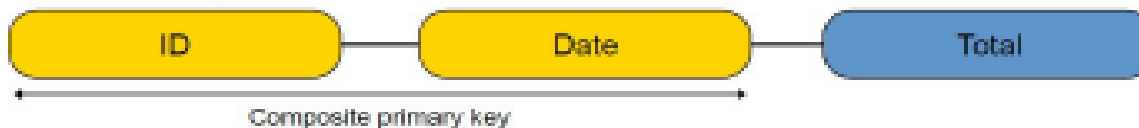
Hash key



id = 100	date = 2012-05-16-09-00-10	total = 25.00
id = 101	date = 2012-05-15-15-00-11	total = 35.00
id = 101	date = 2012-05-16-12-00-10	total = 100.00
id = 102	date = 2012-03-20-18-23-10	total = 20.00
id = 102	date = 2012-03-20-18-23-10	total = 120.00

Hash key

Range key



id = 100	date = 2012-05-16-09-00-10	total = 25.00
id = 101	date = 2012-05-15-15-00-11	total = 35.00
id = 101	date = 2012-05-16-12-00-10	total = 100.00
id = 102	date = 2012-03-20-18-23-10	total = 20.00
id = 102	date = 2012-03-20-18-23-10	total = 120.00

DynamoDB Structure

- ▶ The primary key (Partition Key and optionally a Sort Key) uniquely identifies items in the table,
- ▶ Indexes allow you to create alternative access paths to retrieve data based on different attributes, which is especially useful for complex queries.
- ▶ DynamoDB uses two types of secondary indexes –
 - **Global Secondary Index** – allows you to create an index with a different **Partition Key** and **Sort Key** from the base table's primary key.
 - **Local Secondary Index** – allows you to create an index with the **same Partition Key** as the base table but a **different Sort Key**.

```
CREATE TABLE Orders (  
  UserID STRING,      -- Partition Key  
  OrderDate STRING,   -- Sort Key  
  OrderAmount DECIMAL,  
  Status STRING,  
  PRIMARY KEY (UserID, OrderDate)  
);
```

you want to query orders based on the Status attribute (e.g., "Shipped", "Pending"). Since Status is not part of the primary key, you can create a **Global Secondary Index (GSI)** on Status as the Partition Key.

```
CREATE TABLE Orders (  
  UserID STRING,      -- Partition Key  
  OrderDate STRING,   -- Sort Key (Primary Key)  
  OrderAmount DECIMAL,  
  Status STRING,  
  PRIMARY KEY (UserID, OrderDate)  
)  
WITH LSI = (  
  Status      -- New Sort Key for LSI  
);
```

let's say you want to query orders by UserID, but also want to sort them by Status (instead of OrderDate as in the primary key). You can create a **Local Secondary Index (LSI)** on Status as the Sort Key, keeping UserID as the Partition Key.

DynamoDB

- ▶ API—The API operations offered by DynamoDB include those of the **control plane**, **data plane** (e.g., creation, reading, updating, and deleting), and **streams**.
- ▶ control plane operations
 - CreateTable
 - DescribeTable
 - ListTables
 - UpdateTable
 - DeleteTable

DynamoDB

In the data plane, you perform CRUD operations with the following tools –

Create	Read	Update	Delete
PutItem	GetItem	UpdateItem	DeleteItem
BatchWriteItem- inserting multiple items,	BatchGetItem- fetch data from multiple tables at once.		BatchWriteItem- deleting multiple items,
	Query-retrieve multiple items that share the same partition key . Scan- retrieve all items from a table		

DynamoDB

- ▶ DynamoDB Streams are **useful when you need real-time data change tracking**. They allow you to capture **INSERT, MODIFY, and REMOVE** events in a DynamoDB table and process them asynchronously.
- ▶ Track changes in data for analytics or dashboards.-An e-commerce platform tracks **inventory updates** in real time.
- ▶ Automate workflows when data changes.-Send an **email notification** when a new order is placed., Automatically **update user rankings** when a game score is updated

DynamoDB

- ▶ The stream operations control table streams
- ListStreams–Get a List of Available Streams
- DescribeStream–details about a specific stream,
- GetShardIterator–A **shard iterator** is required to read stream records.
- GetRecords–Fetch item from the stream

Cloud Datastore



Cloud Datastore

- ▶ highly scalable, fully managed NoSQL document database service provided by Google Cloud Platform (GCP).
- ▶ intended to store and retrieve structured data in a schematic way
- ▶ **high availability, scalability, and automatic indexing.**
- ▶ optimized for applications that require **structured data storage with strong consistency and flexible querying.**

Why Use Cloudstore?

- ▶ **Scalability and Performance:** increases its resources automatically as your application grows to accommodate more read and write operations.
- ▶ **Fully Managed Service:** Google Cloud controls the infrastructure, including the creation, scalability, and upkeep of the database,
- ▶ **Flexible Data Model:** As your data requirements change over time, it enables you to store and retrieve structured data without the use of established schemas, giving you flexibility.

How Cloud Datastore Works?

- ▶ **Data is stored as entities** (like rows in a traditional database).
- ▶ **Each entity has a unique key** (similar to a primary key).
- ▶ **Entities belong to a kind** (like tables in SQL).
- ▶ **Properties store data in key-value pairs** (like columns in SQL)

Concept	Datastore	Firestore	Relational database
Category of object	Kind	Collection group	Table
One object	Entity	Document	Row
Individual data for an object	Property	Field	Column
Unique ID for an object	Key	Document ID	Primary Key

Use Cases of Cloud Datastore

- ▶ **Web & Mobile Apps** – User profiles, app configurations, session storage.
- ▶ **E-commerce** – Product catalogs, customer data.
- ▶ **Gaming** – Player scores, game state management.
- ▶ **IoT & Analytics** – Sensor data storage, real-time processing.







Simple DB

- ▶ Amazon SimpleDB is a managed NoSQL database service
- ▶ provides schema-less, key-value pair storage with automatic indexing and query capabilities
- ▶ designed for storing and retrieving small amounts of structured data

Simple DB

Concept	Description	Equivalent in RDBMS
Domain	A collection of items, similar to a table in relational databases	Table
Item	A unique record within a domain, identified by a unique name (key).	Row
Attribute	A key-value pair associated with an item, similar to a column.	Column
Value	The actual data stored in an attribute.	Cell (Column Value)
Query	A SQL-like statement used to retrieve data.	SQL Query

Applications of Amazon SimpleDB

- ▶  Web Applications – Storing user profiles, preferences, and session data.
- ▶  E-commerce – Product catalogs, shopping carts, and order management.
- ▶  IoT & Sensor Data – Storing time-series data from IoT devices
- ▶  Content Management – Managing metadata, blog articles, and tags.
- ▶  Metadata Storage – Storing metadata for files stored in S3
- ▶  Gaming – Storing player scores, game states, and inventory.

cloud storage

- ▶ cloud computing model that enables storing data and files on the internet through a cloud computing provider that you access either through the public internet or a dedicated private network connection
- ▶ securely stores, manages, and maintains the storage servers, infrastructure, and network to ensure you have access to the data
- ▶ It enables **scalability, reliability, and cost-efficiency**, making it ideal for **backups, big data, application hosting, and disaster recovery**.

types of cloud storage

- ▶ **Object storage:** Storing unstructured data, backups, media files, and data lakes.
- ▶ Data is stored as objects (key–value pairs).
- ▶ **Use cases**–Backup, archiving, web content, big data.
- ▶ Examples: AWS S3, Google Cloud Storage, Azure Blob Storage, IBM Cloud Object Storage.
- ▶ A video streaming service storing thousands of videos in AWS S3.

types of cloud storage

- ▶ File Storage: Stores data in a hierarchical file system (folders and directories).
- ▶ Use cases: Enterprise file sharing, content management, web applications.–Suitable for applications that require file-based access.
- ▶ Examples: AWS EFS, Google Filestore, Azure Files, IBM File Storage.
- ▶ A company using **Azure Files** to store shared team documents across remote offices.

types of cloud storage

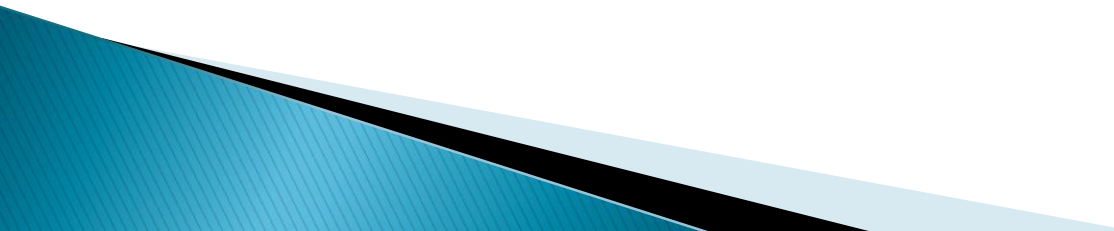
- ▶ Block Storage: Data is divided into fixed-size blocks, similar to a hard drive.
- ▶ Use cases: Databases, virtual machines, high-performance workloads.
- ▶ Example: AWS EBS, Google Persistent Disk, Azure Disk Storage, IBM Block Storage.
- ▶ A cloud-based **MySQL** database using **Google Persistent Disk**

Storage Type	Best For	Structure	Access Method	Performance	Examples
Object Storage	Backup, media, big data	Key-value objects	API-based (HTTP)	High scalability, low latency	AWS S3, GCP Cloud Storage, Azure Blob
File Storage	Shared team storage, applications	Folder and directory system	NFS/SMB protocols	Medium performance	AWS EFS, Google Filestore, Azure Files
Block Storage	Databases, VMs, high-performance apps	Fixed-size blocks	Low-latency network protocols	High performance	AWS EBS, Google Persistent Disk, Azure Disk

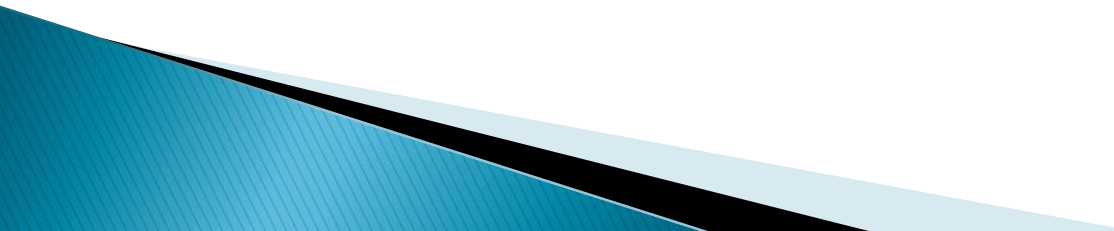
System Security

- ▶ Security issue in Cloud Computing :
 - Cloud security is an evolving sub-domain of computer security, network security, and, more broadly, information security.
 - It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing.

Cloud Security

- ▶ Cloud security, also known as cloud computing security, is a collection of security measures designed to protect cloud-based infrastructure, applications, and data.
 - ▶ These measures ensure user and device authentication, data and resource access control, and data privacy protection.
 - ▶ They also support regulatory data compliance. Cloud security is employed in cloud environments to protect a company's data from distributed denial of service (DDoS) attacks, malware, hackers, and unauthorized user access or use.
- 

System Security

- ▶ Important security and privacy issues :
 - **Data Protection** - To be considered protected, data from one customer must be properly segregated from that of another.
 - **Identity Management** - Every enterprise will have its own identity management system to control access to information and computing resources.
 - **Application Security** - Cloud providers should ensure that applications available as a service via the cloud are secure.
 - **Privacy** - Providers ensure that all critical data are masked and that only authorized users have access to data in its entirety.
- 

Security in Cloud Computing

- ▶ Cloud computing security or cloud security is an important concern which refers to the act of protecting cloud environments, data, information and applications against unauthorized access, DDOS attacks, malwares, hackers and other similar attacks.
- ▶ Three main factors on which planning of cloud security depends.
 - **Resources that can be moved to the cloud and test its sensitivity risk are picked.**
 - **The type of cloud is to be considered.**
 - **The risk in the deployment of the cloud depends on the types of cloud and service models.**

Cloud Computing Security Controls

- ▶ **Preventive Controls** : Preventive controls make the system resilient to attacks by eliminating vulnerabilities in it.-Multi-Factor Authentication (MFA), IAM, Firewalls.
- ▶ **Detective Controls** : It identifies and reacts to security threats and control. Some examples of detective control software are Intrusion detection software and network security monitoring tools.
- ▶ **Corrective Controls** : In the event of a security attack these controls are activated. They limit the damage caused by the attack.
- ▶ **Deterrent/Compensatory Controls** : Deterrent controls are designed to block nefarious attacks on a cloud system. These come in handy when there are insider attackers.

Preventive Control in Cloud Computing Security

- ▶ **Hardening** - process of reducing security exposure and tightening security controls
- ▶ **Security Awareness Training** - process of providing formal cybersecurity education to your workforce
- ▶ **Security Guards** - A person employed by a public or private party to protect an organization's assets.
- ▶ **Change Management** - The methods and manners in which a company describes and implements change within both its internal and external processes.
- ▶ **Account Disablement Policy** - A policy that defines what to do with user access accounts for employees who leave voluntarily, immediate terminations, or on a leave of absence.

Detective Control in Cloud Computing Security

- ▶ Log Monitoring
- ▶ Security Information and Event Management (SIEM) Tool
- ▶ Trend Analysis
- ▶ Security Audits
- ▶ Video Surveillance
- ▶ Motion Detection

Corrective Control in Cloud Computing Security

- ▶ **Intrusion Prevention System (IPS)** - A network security technology that monitors network traffic to detect anomalies in traffic flow. IPS security systems intercept network traffic and can quickly prevent malicious activity by dropping packets or resetting connections.
- ▶ **Backups And System Recovery** - Backups and system recovery is the process of creating and storing copies of data that can be used to protect organizations against data loss.

Compensatory Controls in Cloud Computing Security

- ▶ **Time-based One Time-Password (TOTP)** - A temporary passcode generated by an algorithm that uses the current time of day as one of its authentication factors.
- ▶ **Encryption** - Database security applications, e-mail encryption and other tools.

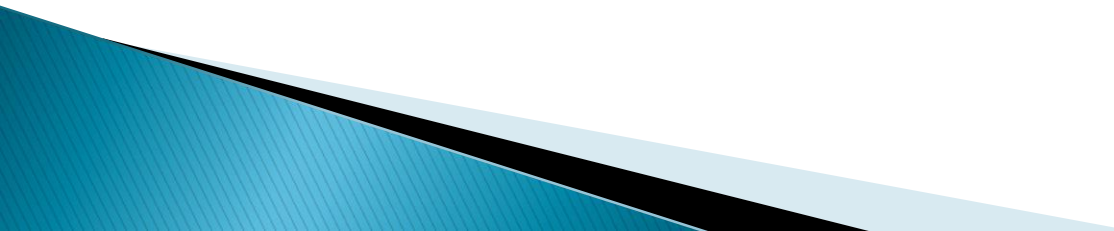
Fault Tolerance

- ▶ What is fault tolerant system ?
 - Fault-tolerance is the property that enables a system to continue operating properly in the event of the failure of some of its components.
 - If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively-designed system in which even a small failure can cause total breakdown.
- ▶ Four basic characteristics :
 - No single point of failure
 - Fault detection and isolation to the failing component
 - Fault containment to prevent propagation of the failure
 - Availability of reversion modes

Fault Tolerance

- ▶ Single Point Of Failure (SPOF)
 - A part of a system which, if it fails, will stop the entire system from working.
 - The assessment of a potentially single location of failure identifies the critical components of a complex system that would provoke a total systems failure in case of malfunction.
- ▶ Preventing single point of failure
 - If a system experiences a failure, it must continue to operate without interruption during the repair process.

Fault Tolerance

- ▶ Fault Detection and Isolation (FDI)
 - A subfield of control engineering which concerns itself with monitoring a system, identifying when a fault has occurred and pinpoint the type of fault and its location.
 - ▶ Isolate failing component
 - When a failure occurs, the system must be able to isolate the failure to the offending component.
- 

Fault Tolerance

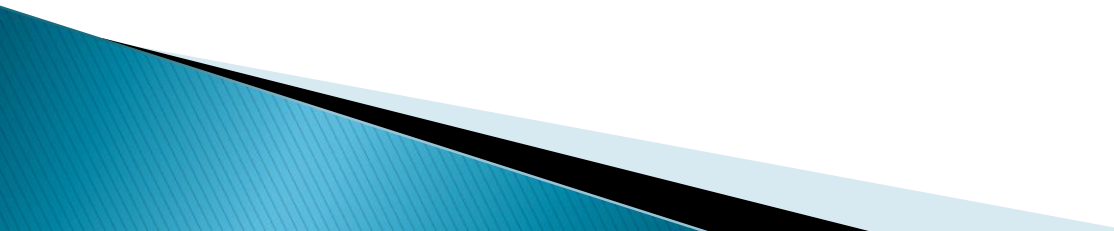
▶ Fault Containment

- Some failure mechanisms can cause a system to fail by propagating the failure to the rest of the system.
- Mechanisms that isolate a rogue transmitter or failing component to protect the system are required.

▶ Available of reversion modes

- System should be able to maintain some check points which can be used in managing the state changes.

System Resilience

- ▶ Resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation.
 - ▶ Resiliency pertains to the system's ability to return to its original state after encountering trouble.
 - ▶ In other words, if a risk event knocks a system offline, a highly resilient system will return back to work and function as planned as soon as possible.
- 

System Resilience

- ▶ Disaster Recovery - Disaster recovery is the process, policies and procedures related to preparing for recovery or continuation of technology infrastructure critical to an organization after a natural or human-induced disaster.
- ▶ Some common strategies :
 - Backup
 - Make data off-site at regular interval
 - Replicate data to an off-site location
 - Replicate whole system
 - Preparing
 - Local mirror systems
 - Surge protector
 - Uninterruptible Power Supply (UPS)

Disaster Recovery in Cloud

- ▶ Data is one of the most valuable assets that any company can hold. One of the best ways to store these assets is within the cloud. However, what can you do if a disaster occurs that affects your cloud data?
- ▶ Disaster recovery in cloud computing can be done through measures such as **a robust backup system** or even by using **multiple servers in different regions** to reduce the harm that a single disaster could cause.

Disaster Recovery in Cloud

- ▶ Disaster recovery (DR) is the process that goes into preparing for and recovering from a disaster.
- ▶ This disaster could take one of a number of forms, but they all end up in the same result: **the prevention of a system from functioning as it normally does**, preventing a business from completing its daily objectives.

Kinds of Disasters

- ▶ **Natural disasters:** Natural disasters such as floods or earthquakes are rarer but not infrequent. If a disaster strikes an area that contains a server that hosts the cloud service you're using, this could disrupt services and require disaster recovery operations.
- ▶ **Technical disasters:** Perhaps the most obvious of the three, technical disasters encompass anything that could go wrong with the cloud technology. This could include power failures or a loss of network connectivity.
- ▶ **Human disasters:** Human failures are a common occurrence and are usually accidents that happen whilst using the cloud services. These could include inadvertent misconfiguration or even malicious third-party access to the cloud service.

Disaster Recovery

- ▶ In the event of a disaster, a company with disaster recovery protocols and options **can minimize the disruption to their services and reduce the overall impact on business performance.**
- ▶ Minimal service interruption means a reduced loss of revenue which, in turn, means user dissatisfaction is also minimized.

Disaster Recovery

- ▶ Having plans for disaster in place also means your company can define its **Recovery Time Objective (RTO)** and its **Recovery Point Objective (RPO)**.
- ▶ The RTO is the maximum acceptable delay between the interruption and continuation of the service and the RPO is the maximum amount of time between data recovery points.

Disaster Examples in Past

- ▶ A data centre run by OVHCloud was destroyed in early 2021 by a fire. All four data centres had been too close, and it took over six hours for firefighters at the scene to put out the blaze.
- ▶ In June 2016, storms in Sydney battered the electrical infrastructure and caused an extensive power outage. This led to **the failure of a number of Elastic Compute Cloud instances and Elastic Block Store volumes** which hosted critical workloads for a number of large companies.
- ▶ In February 2017 an Amazon employee was attempting to debug an issue with the billing system when they accidentally took more servers offline than they needed to.

Disaster Recovery Methods

- ▶ **Backup and restore** - Backing up data and restoring it is one of the easiest, cheapest and fastest ways to recover from a cloud computing disaster. This can be mainly used to mitigate regional disasters such as natural disasters by replicating the data and storing it in a geographically different location.
- ▶ **Pilot Light** – This approach is a method where your company replicates only the minimal and core services it needs to function. This means that only a small part of your IT structure needs to be replicated and provides a minimally functional replacement in case of disaster

Disaster Recovery Methods

- ▶ **Warm Standby** - The warm standby approach is when a scaled down version of your fully functional environment is available and always running in a separate location to your main server. This means that in the event of a disaster, your company can still run a version of the site that is based in a different region.
- ▶ **Multi-site deployment** - Although the most expensive solution of the three, multi-site deployment provides the most comprehensive solution to regional disasters. Multi-site deployment involves running your full workload simultaneously in multiple regions. These regions can be actively used or on a standby in case of disaster in a different region.

Security Issues in Cloud

- ▶ Data Loss
- ▶ Interference of Hackers and Insecure API's
- ▶ User Account Hijacking
- ▶ Changing Service Provider
- ▶ Lack of Skill
- ▶ Denial of Service (DoS) attack

Data Loss

- ▶ Data Loss is one of the issues faced in Cloud Computing. This is also known as Data Leakage.
- ▶ As we know that our sensitive data is in the hands of Somebody else, and we don't have full control over our database.
- ▶ So if the security of cloud service is to break by hackers then it may be possible that hackers will get access to our sensitive data or personal files.

Interference of Hackers and Insecure API's

- ▶ Easiest way to communicate with Cloud is using API. So it is important to protect the Interface's and API's which are used by an external user. But also in cloud computing, few services are available in the public domain.
- ▶ *An* is the vulnerable part of Cloud Computing because it may be possible that these services are accessed by some third parties. So it may be possible that with the help of these services hackers can easily hack or harm our data.

User Account Hijacking

- ▶ Account Hijacking is the most serious security issue in Cloud Computing.
- ▶ If somehow the Account of User or an Organization is hijacked by Hacker. Then the hacker has full authority to perform Unauthorized Activities.

Changing Service Provider

- ▶ Vendor lock In is also an important Security issue in Cloud Computing. Many organizations will face different problems while shifting from one vendor to another.
- ▶ For example, An Organization wants to shift from AWS Cloud to Google Cloud Services then they face various problem's like shifting of all data, also both cloud services have different techniques and functions, so they also face problems regarding that.

Lack of Skill

- ▶ While working, shifting to another service provider, need an extra feature, how to use a feature, etc. are the main problems caused in IT Company who doesn't have skilled Employee.
- ▶ So it requires a skilled person to work with cloud Computing.

Denial of Service (DoS) attack

- ▶ This type of attack occurs when the system receives too much traffic.
- ▶ Mostly DoS attacks occur in large organizations such as the banking sector, government sector, etc. When a DoS attack occurs data is lost. So in order to recover data, it requires a great amount of money as well as time to handle it

THANK YOU!