**Midterm: Part 1: Theory**

**Question 1: Types of Data**

There are three types of data with respect to the structure of data.

* **Structured data:** Any data that can be stored in rows and columns is a structured data.

**A diagram of a computer

Description automatically generated**

**Image Source:** Big Data, Structured, Unstructured-1.pdf ( class lecture ppt, page no 11)

* **Semi-Structured data:** Data that can not be efficiently stored in a table but have a recognizable/organizational property (or structure) which enables it to analyse.

**A screenshot of a computer

Description automatically generated**

**Image Source:** Big Data, Structured, Unstructured-1.pdf ( class lecture ppt, page no 13)

* **Unstructured data:** data that can not be organized in a defined manner/structure.

**A pile of yellow legos

Description automatically generated**

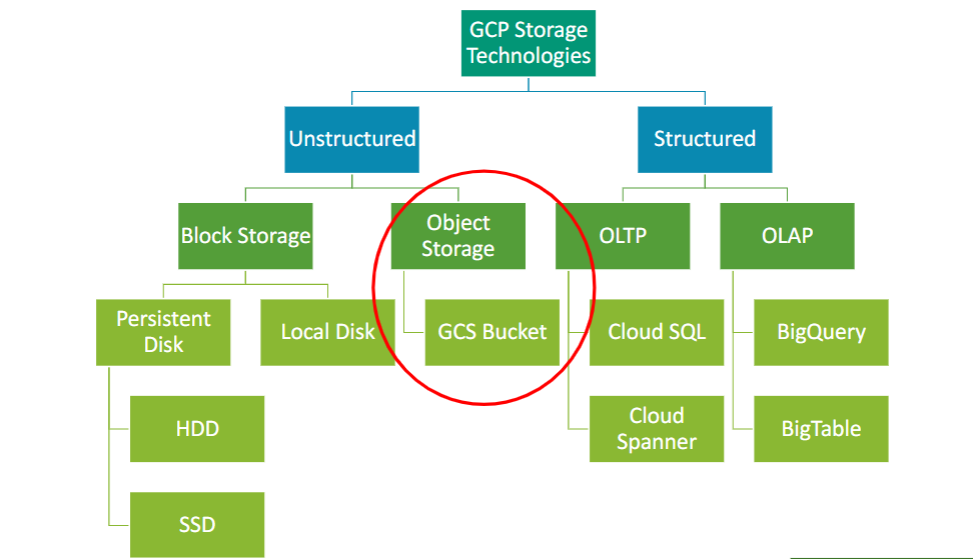
**Image Source:** Big Data, Structured, Unstructured-1.pdf ( class lecture ppt, page no 14)

**1 a. Difference/comparing the three types of data based on the structure.**

The table below compares the properties for Structured, semi-structured and unstructured data to find out the differences.

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Structured data** | **Semi-Structured data** | **Unstructured data** |
| Example | Relational data | XML, JSON | Telemetry, video, image |
| Structure | Rows and columns | XML, JSON etc | character, binary data |
| Transaction management | Robust transaction and concurrency control | low to minimal | No control |
| Versioning | Versioning on rows, columns, tuples | on tuples and graph | versioned as a whole |
| Scalability | limited to no scalability | more scalability than structured data | Most scalable |
| Ability to Query | Complex joins with high performance | much lower than structured data | least to no possibility |

**1 b. Storage Types for structured and unstructured data**

****

**Image Source:** Introduction to cloud storage\_2024.pdf (class lecture ppt, page no 11)

* **Unstructured data:** The unstructured data can be stored in the below two formats/types.
  + **Block Storage:** data is divided into fixed size blocks and stored as single units. Each block is identified by its block number which enables it to perform read/write operations without retrieving the entire dataset.

**Example: Persistent Disk, local disk**

* + **Object Storage:** data is stored as a discrete unit called as object. The metadata stores information about the attributes of the object.

**Example: Google cloud storage bucket**

* **Structured data:** it can be stored in the below listed formats/types.
  + **OLAP:** online analytical processing is a technology used to perform multi-dimensional analysis on aggregated data.

**Example: BigQuery, BigTable**

* + **OLTP:** Online transactional processing is used to process database transactions.

**Example: CloudSQL, CloudSpanner**

**1 c. GCP services for structured and unstructured data**

* **Block Storage (for unstructured data):** Persistent disk is the GCP service used for block storage.
  + **Advantages:** 
    - **Reliability:** minimum to no downtime
    - **High performance:** less metadata so there is lesser data transfer enabling higher data retrieval speed.
    - **Scalability:** blocks can be added as and when needed without any performance impact.
  + **Disadvantages:** 
    - **Expensive:** it uses storage area network (SAN) which has maintenance and management overhead.
    - **Limited use of metadata:** for unstructured data which relies on metadata it can cause some latency for data retrieval.
* **Object Storage (for unstructured data):** bucket is the GCP service used for object storage.
  + **Advantages:** 
    - **High scalability:** we can add storage easily.
    - **AI/ML efficiency:** it can store huge amounts of data or very large files which makes it suitable for analysis using AI/ML.
    - **Pay for what you need:** it is affordable as user pay only for what they are using and can add more storage when required.
    - **Advanced Search capability:** object can be searched with the help of the metadata, contents etc.
  + **Disadvantages:** 
    - **Not efficient for transactional data:** writing data is slow which does not make it suitable for transactional data systems.
    - **Object once created can not be updated:** we need to recreate and upload data when we want to update it.
* **OLAP (for structured data):** BigQuery and BigTable are the GCP services used for OLAP

**BigQuery** : is a enterprise data warehouse for huge amount of relational structured data.

**BigTable** is a NoSql database it is suitable for low latency and huge read/wright operations.

* + **Advantages:** 
    - BigTable
      * **Scalability**: clusters can be scaled up to handle more read and writes.
      * **Managed Service:** limited inputs are required to manage it as it is fully manged service.
      * **No downtime for cluster resizing:** automatically does load balancing for the available nodes.
    - BigQuery
      * **Serverless**: managed service, no infrastructure is required.
      * **Scalability:** can handleextremely large datasets.
      * **High-performance:** uses advanced optimization for high performance.
      * **SQL Support:** supports SQL, hence easy to work with.
      * **AI/ML suitability:** inbuilt functions and integrations for AI/ML models.
      * **Data Security:** offers many security features.
  + **Disadvantages:**
    - BigTable
      * **Limited query capability:** not optimized for complex SQL queries.
      * **Not a cost effective for small queries:** data storage and read/write operation based pricing. So may not be cost effective for small operations.
    - BigQuery
      * **Cost:** BigQuery may not be cost effective for high number of small queries.
      * **Customization limitations:** It is a fully managed service provided by GCP so only a limited number of changes can be made by the user.
      * **Integration limitations:** limited integration options for non-GCP platform applications/systems.
* **OLTP(for structured data):** CloudSQL and Cloud Spanner are the GCP services used for OLTP.
  + **Advantages:**
    - **CloudSQL:**
      * **Low Maintenance:** fully managed service
      * **Backup and recovery:** ensured backup
      * **Security:** Encryption and firewall protection
      * **Integration:** can integrate easily with many applications/systems
    - **Cloud Spanner:**
      * **Suitable for all kind of workloads:** small or big
      * **Scalability and High Availability:** limitless scalability
      * **Managed service:** reduced operational cost
      * **Data Encryption:** high security with data encryption and other features.
      * **Service Level Agreement 99.99%:** so minimal to no downtime
  + **Disadvantages:**
    - **CloudSQL:**
      * **Database engine:** not many database engines are compatible other than MYSQL and PostgreSQL.
      * **Incremental/Transaction logs:** only offers full backup.
      * **Cost:** high performance configurations may incur higher costs
    - **Cloud Spanner:**
      * **Query level challenges due to PK oriented design:** any update or delete query must specify PK.
      * **Views are not supported:** can not create views

**Question 2: Comparing IaaS, PaaS, SaaS**

**IaaS-**Infrastructure as a service, **PaaS –** Platform as a Serviceand **SaaS –** Software as a Service are the three fundamental cloud deployment service models.

**Description:**

* **IaaS: Infrastructure as a service**
  + It provides infrastructure like virtual server, network equipment, data canters space etc as per user requirement to perform their operations or software services.
* **PaaS: Platform as a service**
  + It provides the hardware and the software where the user can build their own applications.
  + Example: Creating a MYSQL Database on GCP and using it for your project.
* **SaaS: Software as a Service**
  + It provides a given software like Rally, outlook etc to be used by the user.
  + The user doesn’t have to worry about hosting and maintain it.

**Differences:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **IaaS** | **PaaS** | **SaaS** |
| Intended user | network team | developer/engineering team | end user (limited to no tech expertise) |
| Access | includes access to VM, storage etc | development tools like GCP MYSQL etc | enables end user to use the application like outlook |
| Technical Expertise | good technical expertise is required | less expertise required than IaaS | no expertise required |
| User/consumer level control | user controls almost every aspect except the hardware | User controls the data | no control by the user |
| Example | Virtual Machine, storage, load balancer, security | GCP MySQL, Cloud Spanner | Outlook, Rally, Instagram |

**Question 3: Cloud storage classes – standard, nearline, coldline, archive**

**Desciption:**

* **Standard:** it is the most suitable storage option for data that is accessed frequently. The data availability of this storage class is the highest which means that there is minimum to no delay in data retrieval.
* **Nearline:** it is a high-durability and low-cost storage class suitable for data that is accessed less frequently compared to data in standard storage class.
* **Coldline:** it is cheaper than nearline storage and highly durable storage for lesser frequently accessed data.
* **Archive:** it is the cheapest storage class suitable for backup and disaster recovery operations related data where the data is accessed on a very large interval of time (once in a year).

**3 a: Comparison between the storage classes**

The below table describes the key difference between them

A table with text and numbers

Description automatically generated

**Image Source:** Introduction to cloud storage\_2024 (class lecture, page 15)

Apart from the above differences, there are some more differences shown in the below table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Storage Class** | **Standard** | **Nearline** | **Coldline** | **Archive** |
| Cost | Most expensive | lesser than Standard | very lesser than coldline | least expensive |
| Data accessibility | frequently accessed data | once in a month | once in three months | once in a year |
| Application suitability | Transactional applications with data intensive processing | Monthly reports or data files | Quarterly reports | backup and archiving |

**3 b: Duration of accessing the data in these storage classes**

**Standard:** Most frequently accessed data, suitable for huge data processing applications

**Nearline:** Suitable for once in a month accessed data.

**Coldline:** Suitable for once in three months accessed data.

**Archive:** Suitable for once in a year accessed data.

**3 c: Cost efficiency**

**Standard:** the most expensive storage class for its high availability and durability

**Nearline:** expensive but cheaper than standard

**Coldline:** relatively cheaper option for quarterly accessed data

**Archive:** least expensive

**Question 4: Relational and non-relational databases, similarity and differences**

**Description:**

**Relational Database** also known as RDBMS or SQL database (data in rows and columns)

* **Relationship/Schema:** we can establish relationship between entities/tables as the data is in tabular (rows and columns) format.
* **Complex query:** due to the entity relationship (ER), we can make complex joins and implement critical business logic to get the required datasets.
* **Normalization:** we can normalize and de-normalize the data based on our use case.

**Non-relational Database** also known as non-RDBMS or noSql database (data is not in rows and columns)

* **Schema/relationship:** This type of data has limited to no schema/relationship hence different types of data can be stored in different formats.
* **Key-Value database:** if thedata has key value pair relationship, then we can use the key-value pair database to save it.
* **Document Databases:** for semi-structured data like JSON, YAML etc
* **Graph Databases:** for data having multidirectional relationships.

**Comparing Relational and non-relational databases**

While both relational and non-relational databases store data and carry utmost significance for their respective use case, there are some similarities and differences.

**Similarity:**

* **Availability:** Both the systems store data and keep it available for query/retrieval.
* **Business Contingency:** Both the database systems should/usually have backup and disaster recovery capability.
* **Load balancing:** Both should/usually have load balancing implemented at system level.
* **Fault tolerant:** Both the systems should/usually be fault tolerant.
* **Interoperability:** both the systems have capability to connect to external systems for inbound and outbound data feeds.

**Difference:**

* The below table describes the differences.

|  |  |  |
| --- | --- | --- |
| **Property** | **Relational Database** | **Non-Relational Database** |
| Structure | fully structured data | semi-structured and unstructured data |
| Query Capability | Complex and highly optimized query performance | limited to no query performance |
| interoperability with other data | schema should match for joins | Schema/relationship matching is not mandatory to append/combine any data to the given data |
| Scalability | restrictions due to schema | can be scaled easily |
| Performance | can be highly optimized with proper indexing | depends on network and underlying hardware |
| Data Integrity | strong controls to establish data integrity | relatively difficult to control/manage consistency |
| Use case | data is defined and size is predicable | data is undefined (unknown schema) and size can change abruptly |

Based on the use case, we decide on what type of database we should use.

**Question 5: five Vs of Big Data**

**Description:**

Five Vs of Big Data are

A diagram of a big data

Description automatically generated

**Image Source:** Big Data, structured, unstructured-1.pdf (class lecture page 3)

* **Volume:** the enormous size of the big data
  + The volume of big data is increasingly growing with each passing day
  + There are millions of logs recorded from various sources every single minute which are stored and analysed to study and predict many possible outputs.
  + Example: 500 million tweets are made every single day.

**Source**: https://www.omnicoreagency.com/twitter-statistics/#:~:text=On%20average%2C%20500%20million%20tweets,200%20billion%20tweets%20every%20year.

* **Velocity:** the speed at which the big data is generated/accumulated
  + In the present-day scenario, we have humongous live streaming and telemetry data generated at a rapid pace.
  + This data needed to be processed and stored in an efficient manner.
  + Example: Netflix, Tesla telemetry data, process execution logs etc
* **Variety:** the data can be structured, semi-structured and unstructured
  + Different type of data needs to be stored and processed differently.
  + Identifying the type of data and redirecting it the relevant system/database for processing is crucial to make the best use of it.
  + Example: Tweets, emails, review comments, videos, images, live streaming etc
* **Veracity:**  data can be inconsistent and uncertain
  + Sometimes too much data can be difficult to analyse and process
  + Sometimes too little data can not be sufficient to derive sensible conclusions
  + Example: on a Tylor Swift concert day, there can be huge traffic in the host city and the same city can have empty roads on a Thanksgiving day because everyone is at home.
* **Value: “**data is the new gold” only when we can find meaningful insights from it
  + it needs expertise to understand the data and get critical information out of it to help solve an existing problem or predict an upcoming situation.
  + the more efficiently we can draw value out of data the better we can utilize it.
  + Example: The hospital patient data can be used to ensure logistical supplies for operational support but the same data can be used to predict/study illness patterns and provide preventive care, vaccines and awareness among people and the government.

**Question 6: OLAP and OLTP**

**Description:**

* **OLAP:** online analytical processing
  + it is basically used for analytical purpose as it can process large amount of data in an efficient manner.
  + involves working on aggregated data and data cubes (three-dimensional data)
  + it helps in decision making and problem solving.
  + it uses data from OLTP systems which gets ingested to OLAP platform via ETL (Extract, Transform, Load) pipelines.
  + it is usually capable of handling more volume of data than the OLTP as analytics may be required to use muti-year data.
  + Example: Sales trends in last 10 years
* **OLTP:** online transactional protocol
  + It is designed to deal with large volumes of transactions from multiple users.
  + This transactional data is moved to the OLAP system via ETL pipelines to perform analytics.
  + A robust OLTP system ensures business contingency and load balancing in peak hours.
  + It is generally a relational database with a capability of executing multiple transactions (read/write/update) efficiently in a concurrent manner.
  + It is also equipped to maintain data/system integrity which means, if a payment is made by an online payment gateway, it should be “complete” or “declined”. It cannot be partially done.
  + Example: Online Banking, Amazon online shopping etc

A simplified explanation of OLTP and OLAP can be, OLTP is used to run the business like banking, hotel booking, online shopping etc and OLAP is used to understand the business like demand, trends, forecast etc.

Both OLAP and OLTP have significant role to for a successful business