

MODULE - VI

BIG DATA STORAGE TECHNOLOGY

Q-1 Differentiate between horizontal scaling and vertical scaling.

<u>Ans:</u>	<u>Horizontal scaling</u>	<u>Vertical scaling</u>
	It is also known as <u>scaling out</u> .	It is also known as <u>Scaling up</u> .
	It means that enhancing the performance of server or node by adding more nodes/machine to pool of machines.	It means that increasing the capacity of machine by adding more resources, such as more memory or more processors or adding RAM.
	Horizontal scaling can be achieved with the help of clustering, <u>distributed file system</u> and <u>load balancing</u> .	Vertical scaling can be achieved with the help of <u>RDBMS</u> .
	It provides out of box redundancy and high availability.	It doesn't provide out of the box redundancy and fault tolerance.
	If provides simple, fast access data storage that is capable of storing large datasets.	RDBMS are good for handling transactional workloads involves <u>small amounts</u> of data.

It provides <u>fast</u> read/write capability	It provides <u>random</u> read/write capability
---	---

Distributed File System are not ACID compliant	But RDBMS are ACID compliant
--	------------------------------

It does not provide the ability to search the contents of files	It provides better backup and recovery procedures and ability to search the contents.
---	---

Q-3. Write a note on Distributed File System;

Ans: Distributed File System support Schema-less data storage. It provides out of box redundancy and high availability by copying data to multiple locations via replication. A storage device that is implemented with a distributed file system provides simple, fast access data storage. It provides fast read/write capability.

It is not ideal for datasets comprising a large number of small files. Multiple smaller files are generally combined into a single file. This allows the distributed file systems to have increased performance when data must be accessed in streaming mode with no random reads and writes.

It provides an inexpensive storage option for storing large amounts of data over a long period of time that needs to remain online. The main disadvantage of distributed file systems do not provide the ability to search the contents of files as standard out-of-the box capability.

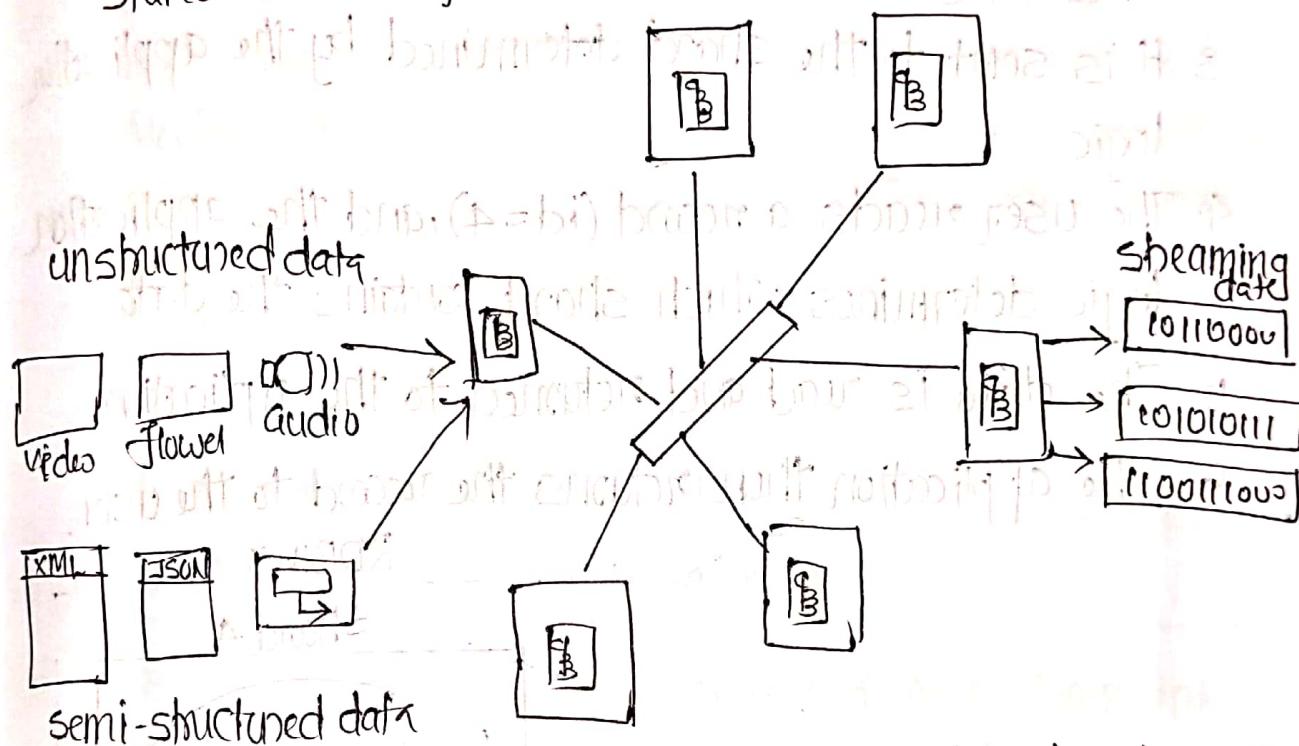


Fig: A distributed file system accessing data in streaming mode with no random reads and writes.

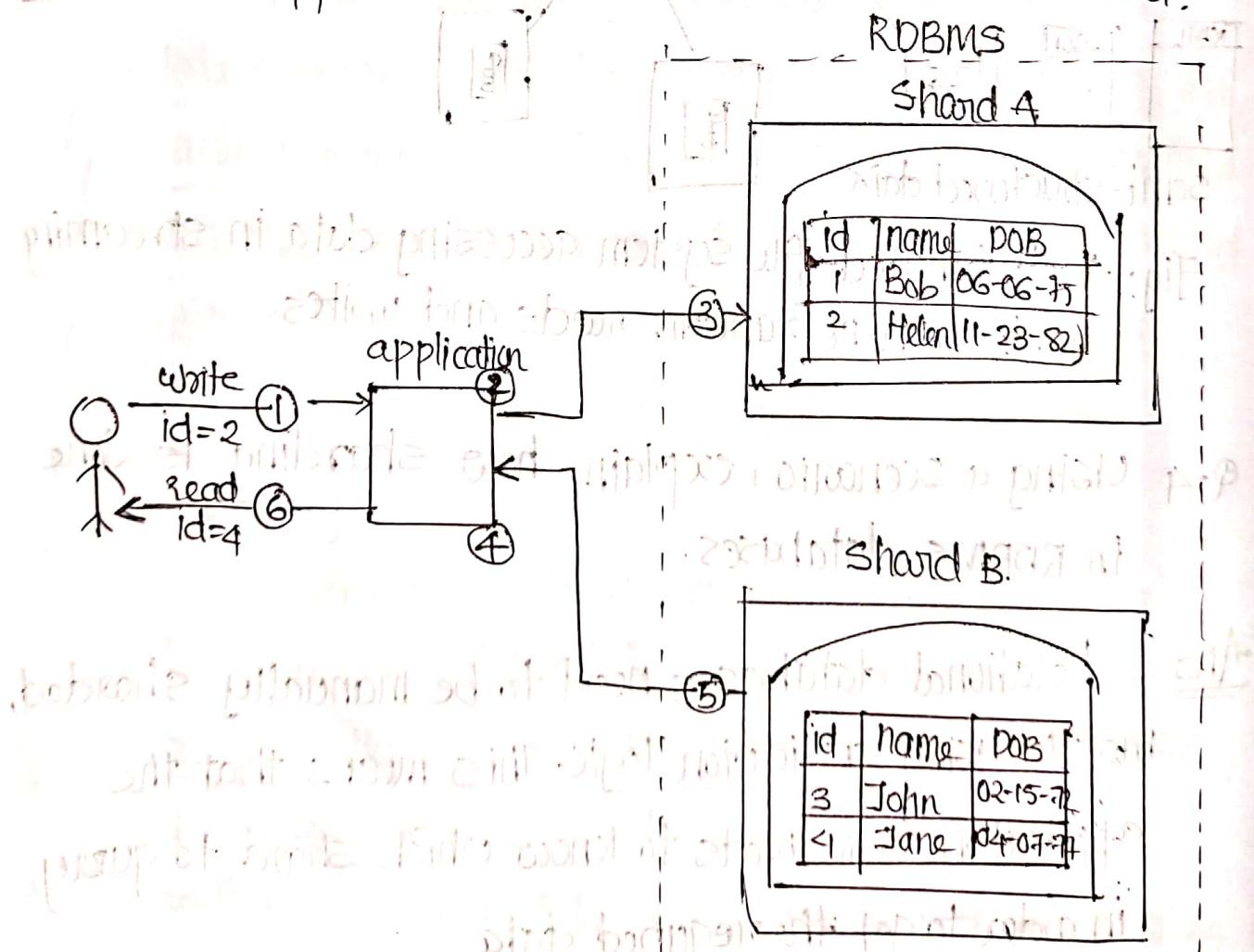
Q-4 Using a scenario, explain how sharding is done in RDBMS databases.

Ans: Relational databases need to be manually sharded, mostly using application logic. This means that the application logic needs to know which shard to query in order to get the required data.

Data processing when data from multiple shards

The following steps are shown in fig:

1. A user writes a record (id=2)
2. The application logic determines which shard it should be written to.
3. It is sent to the shard determined by the application logic.
4. The user reads a record (id=4), and the application logic determines which shard contains the data.
5. The data is read and returned to the application.
6. The application then returns the record to the user.



25

Differences between RDBMS and NoSQL databases.

Ans:

RDBMS	NoSQL
RDBMS stands for Relational Database Management Systems	NoSQL stands for Not-only SQL
Data stored in a relational model, with rows and columns.	Data stored in a host of different databases - each with different data storage models.
It has a fixed schema.	It has no fixed schema (it follows dynamic schema).
It supports a powerful query language.	It supports a very simple query language.
RDBMS is ACID compliant	It is not ACID compliant
Relational database manages only structured data.	NoSQL database can manage structured, unstructured and semi-structured data.
RDBMS do not provide out-of-the box redundancy and fault tolerance.	It provides highly scalable and fault-tolerant.
It has random read/write properties.	It gives both read and write properties.
It supports vertical scaling	It supports horizontal scaling

Figure:

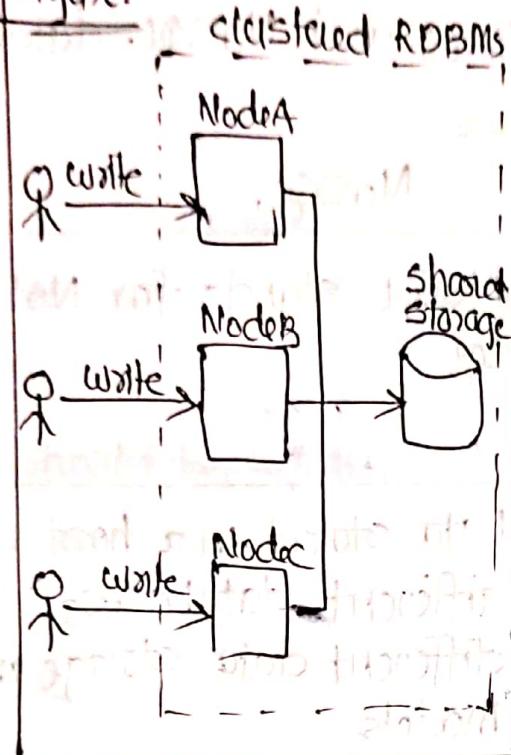
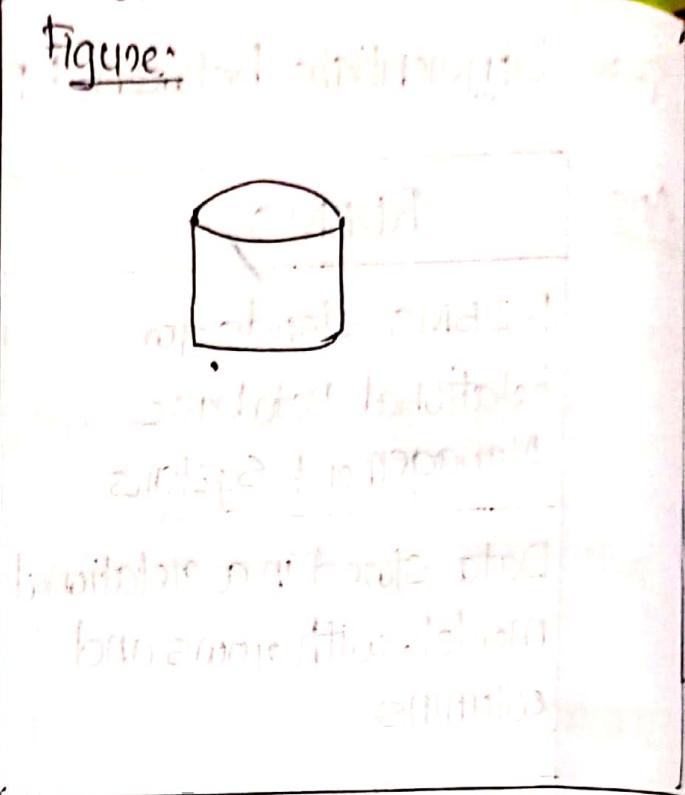


Figure:



To take note on cut lines in the front page

Q-6 What are the features of NoSQL storage devices.

Ans: Not-only SQL (NoSQL) refers to technologies used to develop next generation non-relational databases that are highly scalable and fault-tolerant.

Features

1. Schema-less data model - Data can exist in its raw form.
2. Scale out rather than scale up - More nodes can be added to obtain additional storage with a NoSQL database.
3. Highly available - This is built on cluster-based technologies that provide fault tolerance out-of-the-box.

4. Lower operational costs - Many NoSQL databases are built on open source platforms with no licensing costs.
5. Eventual consistency - Data reads across multiple nodes but may not be consistent immediately after a write. However, all nodes will eventually be in a consistent state.
6. Base, not ACID - BASE compliance requires a database to maintain high availability in the event of network/host failure, while not requiring the database to be in a consistent state whenever an update occurs.
7. API driven data access - Data access is generally supported via API based queries, including RESTful APIs, whereas some implementations may also provide SQL-like query capability.
8. Auto sharding and replication - To support horizontal scaling and provide high availability, a NoSQL storage device automatically employs sharding and replication.
9. Integrated caching - This removes the need for a third-party distributed caching layer.
10. Distributed query support - NoSQL storage devices maintain consistent query behavior across multiple shards.
11. Polyglot persistence - It persisting data using different types of storage technologies within the same solution.

12. Aggregate-focused - NoSQL storage devices store de-normalized aggregate data thereby eliminating the need for joins and extensive mapping between application objects and the data stored in the database.

Q-07 Briefly write about the NoSQL storage device used by MongoDB.

Ans: Document storage device also store data as key-value pairs. However, unlike key-value storage devices, the stored value is a document. These documents can have a complex nested structure.

Key	value
517	{ invoiceId: 1855, date: 1960051, custId: 29317, items: [{ itemId: 471971; 2}, {itemId: 971, qty: 5}] }
419	{ invoiceId: 1853, date: 150091, custId: 25481, items: [{ itemId: 481, qty: 6}, {itemId: 851, qty: 8}] }

- * A document storage device is appropriate when:
 - storing semi-structured document-oriented data comprising flat or nested schema.
 - schema evolution is a requirement as the structure of the document is either unknown or is likely to change.
 - applications require a partial update of the aggregate stored as a document.
 - searches need to be performed on different fields of the documents.
 - storing domain objects, such as customers in serialized object form.
 - query patterns involve insert, select, update and delete operations.

- * A document storage device is inappropriate when:
 - multiple documents need to be updated as part of a single transaction.
 - performing operations that need joins between multiple documents or storing data that is normalized.
 - schema enforcement for achieving consistent query design.
 - the stored value is not self-describing.
 - binary data needs to be stored.

Examples of document storage devices include MongoDB, CouchDB and TeraStore.

Q-8 Explain the different types of NoSQL databases

Ans: NoSQL storage devices can mainly be divided

into four types:

1. key-value

2. document

3. column-family

4. graph

1. key-value

key-value storage devices store data as key-value

pairs and act like hash tables. The table is a list of values where each value is identified by a key. Value look-

up can only be performed via the keys. Partial updates are not possible. Key-value storage devices generally do not maintain any indexes, therefore writes are quite fast. Based on a simple storage model, key-value

storage devices are highly scalable.

* A key-value storage device is appropriate when:

→ unstructured data storage is required.

→ high performance read/writes are required.

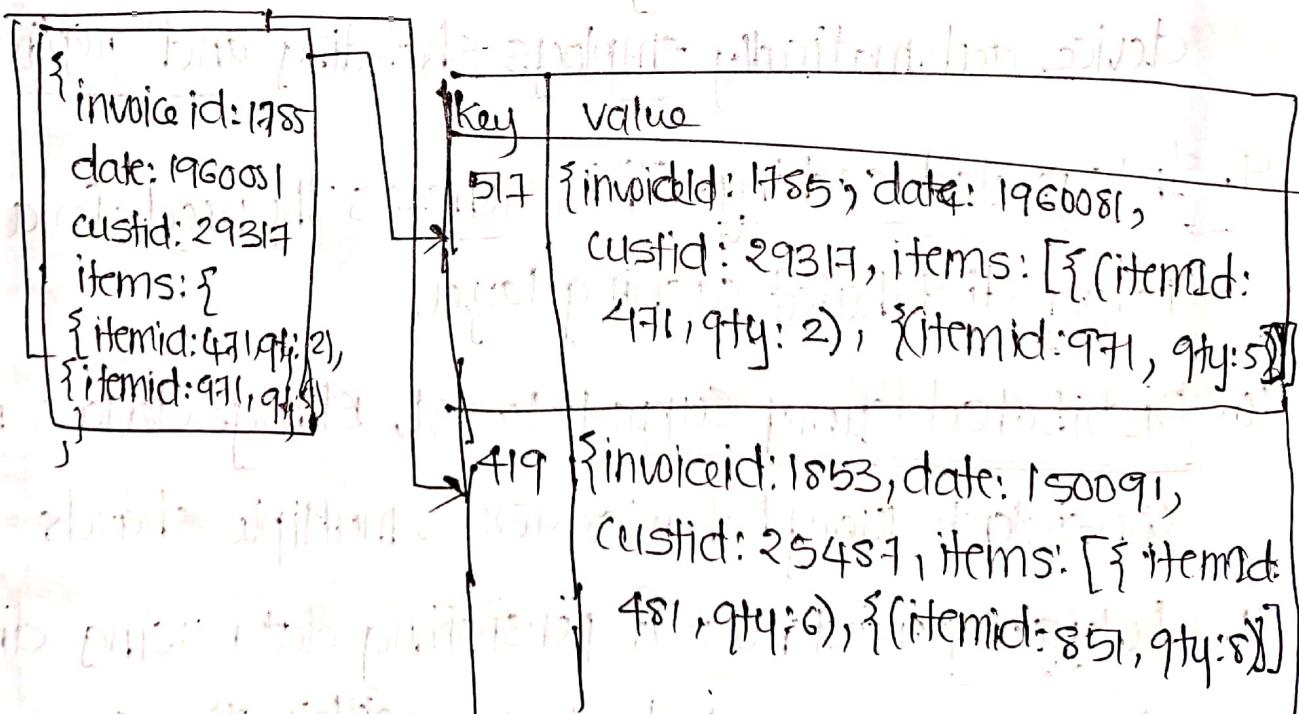
→ the value is fully identifiable via the key alone.

→ the value is a standalone entity that is not dependent on other values.

- * A key-value storage device is inappropriate when:
 - applications require searching or filtering data using attributes of the stored value
 - relationships exist between different key-value entries
 - multiple-keys require manipulation in a single operation.

2. Document

Document storage device also store data as key-value pairs. However, unlike key-value storage devices the stored value is a document. These documents can have a complex nested structure.



- * A document storage device is appropriate, when:
 - storing semi-structured document-oriented data comprising flat or nested schema.
 - schema evolution is a requirement as the structure of the document is either unknown or is likely to change.
 - applications require a partial update of the aggregate stored as a document.
 - searches need to be performed on different fields of the documents.
 - storing domain objects, such as customers in serialized object form.
 - query patterns involve insert, select, update and delete operations.

- * A document storage device is inappropriate when:
 - multiple documents need to be updated as part of a single transaction.
 - performing operations that need joins between multiple documents or storing data that is normalized.
 - schema enforcement for achieving consistent query design.
 - the stored value is not self-describing.
 - binary data needs to be stored.

Examples of document storage devices include MongoDB, CouchDB and TeraStore.

3. Column-family

If stored data much like a traditional RDBMS but group related columns together in a row, resulting in column-families. Each column can be a collection of related columns itself, referred to as a super-column. Each row consists of multiple column families. It provide fast data access with random read/write capability.

- * A column-family storage device is appropriate, when
 - realtime random read/write capability.
 - data represents a tabular structure.
 - support for schema evolution
 - efficient use of storage
- * A column-family storage device is inappropriate when.
 - relational data access is required
 - ACID transactional support is required.
 - SQL-compliant queries need to be executed
 - query patterns are likely to change frequently.

4. Graph

Graph storage devices are used to persist interconnected entities. If emphasis is on the structure of the entities, graph storage devices place emphasis on storing the linkages between entities. Entities are stored as nodes, and are also called vertices.

* A graph storage device is appropriate when:

- interconnected entities need to be stored.
- querying entities based on the type of relationship.
- finding groups of interconnected entities.
- mining data with a view toward finding patterns.

* A graph storage device is inappropriate when:

- involves searching for nodes or edges.
- Binary storage is required.
- queries based on the selection of node/edge attributes dominate node traversal queries.

e.g: Neo4J, OrientDB.

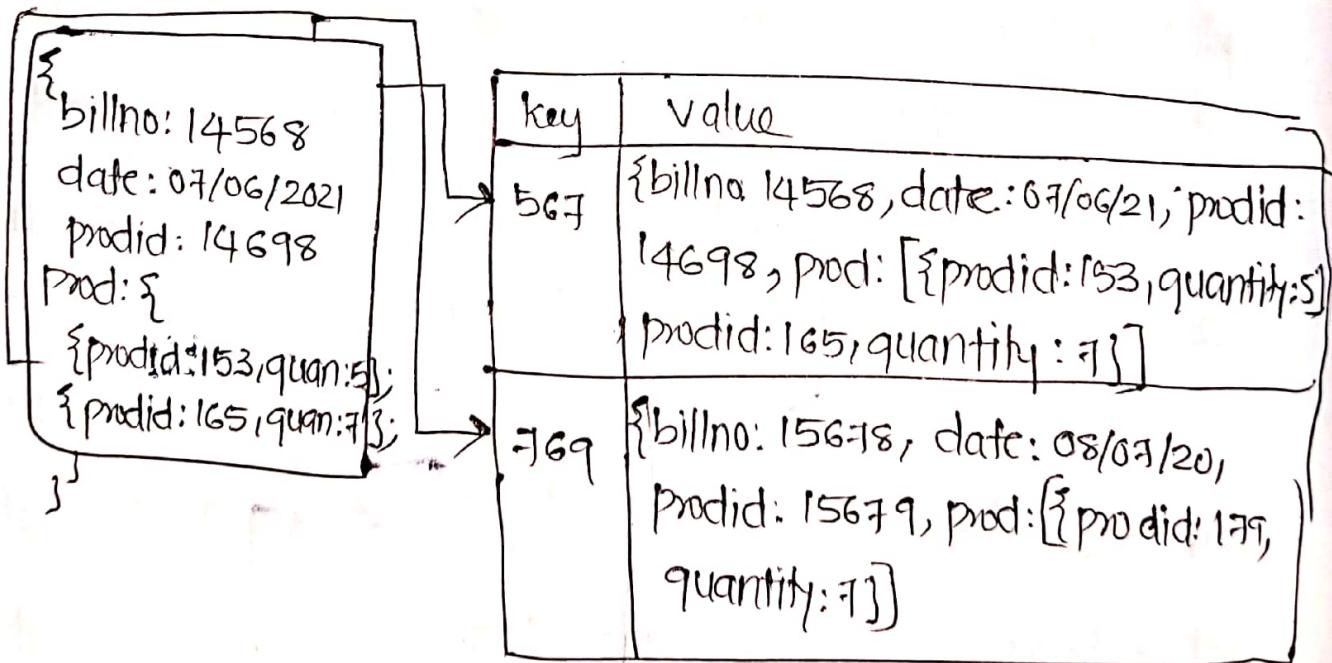
Types of NoSQL - ~~Document~~, e.g.,

1) Key-Value

→ ~~Machine~~ friendly format for application code

key	value	format
859	Miller, 12.10.45, 30, Good Product Service	Text
465	10.1011.00110111000011100110	Image
785	<prod-id>1567</prod-id><Total>1495</Total>	XML

2) Document

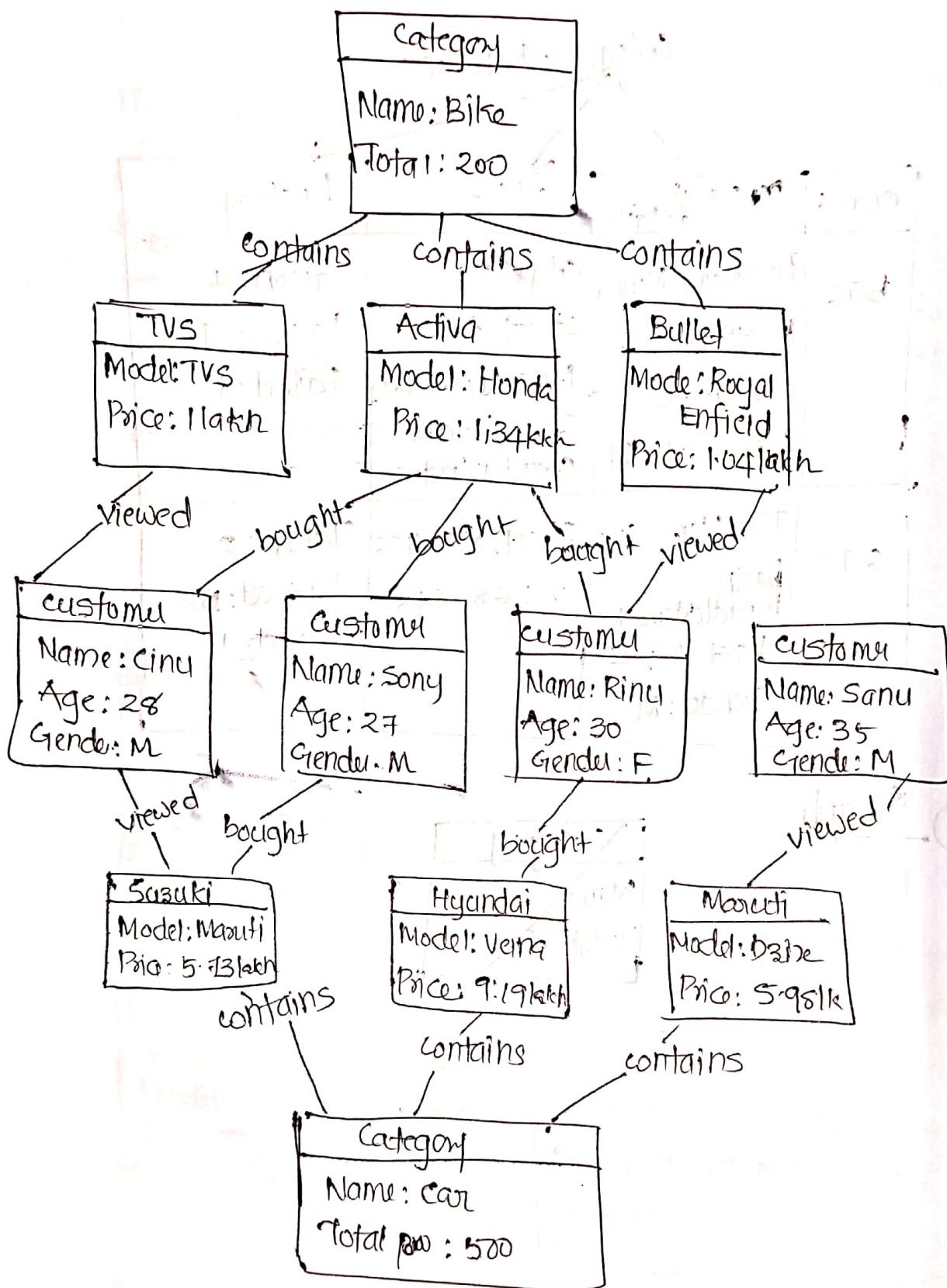


3) Column-Family

column-families

empid	Personal details	address	history
596	FirstName: Minnu LastName: S. Gender: F DOB: 04-12-97	City: PTA Street: ABC 123 Pincode: 689645 State: Kerala Country: India	Taken: 4 Passed: 4 Failed: 1
698	FirstName: Sinc LastName: S. MiddleName: M Gender: M	Street: XYZ 456 Pin: 689643	Taken: 3 Passed: 2 Failed: 1

4) Graph



Q-9

Differentiate between Document storage device and key value storage devices.

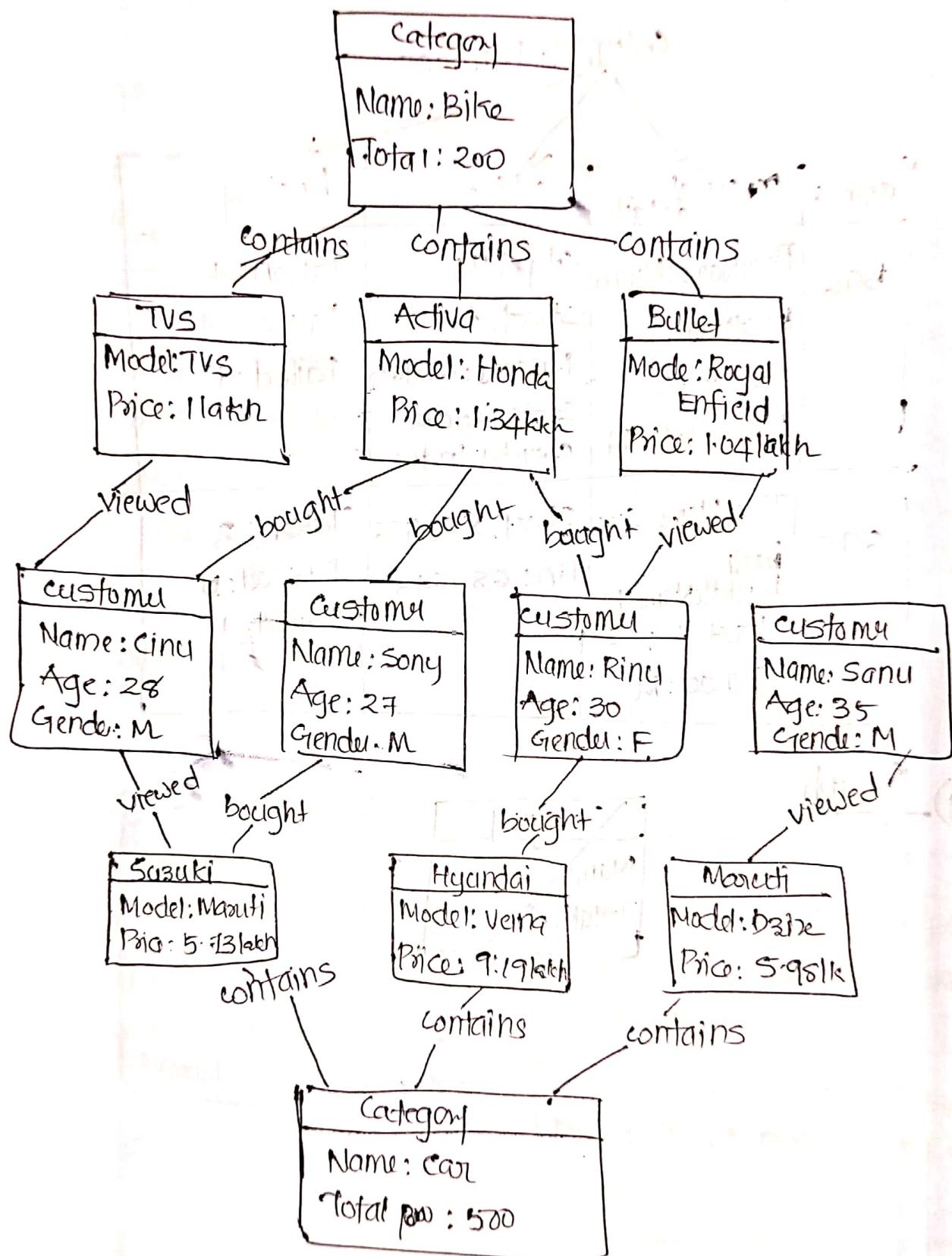
- Ans:
- Document storage devices are value aware
 - the stored value is self-describing.
 - A select operation can reference a field inside the aggregate value
 - A select operation can retrieve a part of the aggregate value
 - Partial updates are supported, therefore a subset of the aggregate can be updated.
 - indexes that speed up searches are generally supported.

Q-10 Write a short note on graph based storage device
i. When it is appropriate to use?

A: Graph storage devices are used to persist interconnected entities. If emphasis is on the structure of the entities, graph storage devices place emphasis on storing the linkages between entities. Entities are stored as nodes and are also called vertices.

- * A graph storage device is appropriate when:
 - interconnected entities need to be stored
 - querying entities based on the type of relationship
 - finding groups of interconnected entities
 - mining data with a view toward finding patterns.

④ Graph - Eg:



Q-11 Differentiate between NewSQL and NoSQL databases.

Ans:

NoSQL	NewSQL
NoSQL is a schema-free database	NewSQL is schema-fixed as well as a schema free database
It is horizontally scalable.	It is also horizontally scalable.
It possesses automatically high-availability	It possesses built-in high availability.
It promotes CAP properties	It promotes ACID properties
It is not used for developing OLTP systems (online Transactional Processing)	It is used for developing OLTP systems with very high volumes of transactions.

There are low-security concerns

Examples of NoSQL databases include DynamoDB, MongoDB

There are moderate security concerns.

Examples of NewSQL databases include VoltDB, Neo4jDB, InnoDB

Q-12. List out the advantages of NewSQL databases.

- Ans: → It is less complex applications, greater consistency.
→ It supports SQL compliant syntax for data definition and data manipulation operations.
→ It uses a logical relational data model for data storage.
→ It introduces new implementations for traditional relational databases.

Q-13. What are NewSQL databases?

- Ans: NewSQL storage devices combine the ACID properties of RDBMS with the scalability and fault tolerance offered by NoSQL storage devices. It generally supports SQL compliant syntax for data definition and data manipulation operations, and they often use a logical

relational data model for data storage.

NoSQL databases can be used for developing OLTP systems with very high volumes of transactions.

Ex: Banking System.

They can also be used for real-time analytics.

Ex: operational analytics, as some implementations leverage in-memory storage.

Q-2 Explain the different methods for implementing on disk storage devices.

Ans: Ref 3rd, 4th, 5th, 6th, 8th, & 13th answers.