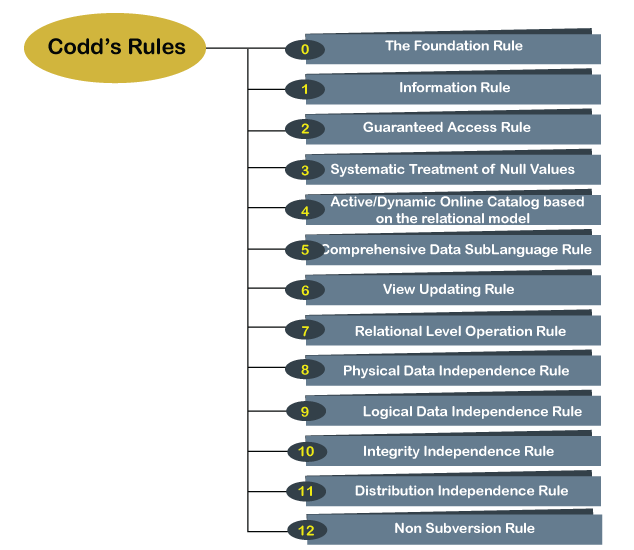
**MN DBT**

codds 12 rules

Text

Description automatically generated

**Error Handling and Exceptions:**

When an error occurs inside a stored procedure, it is important to handle it appropriately, such as continuing or exiting the current code block’s execution and issuing a meaningful error message. MySQL provides an easy way to define handlers that handle general conditions such as warnings or exceptions to specific conditions e.g., specific error codes.

***DECLARE action HANDLER FOR condition\_value statement;***

If a condition whose value matches the condition\_value, MySQL will execute the statement and continue or exit the current code block based on the action. The action accepts one of the following values: CONTINUE : the execution of the enclosing code block ( BEGIN … END ) continues. EXIT : the execution of the enclosing code block, where the handler is declared, terminates. The statement could be a simple statement or a compound statement enclosing by the BEGIN and END keywords.

CREATE PROCEDURE InsertSupplierProduct(

IN inSupplierId INT,

IN inProductId INT

)

BEGIN

DECLARE EXIT HANDLER FOR 1062

BEGIN

SELECT CONCAT('Duplicate key error occurred') AS message;

END;

INSERT INTO SupplierProducts(supplierId,productId)

VALUES(inSupplierId,inProductId);

END$$

DELIMITER ;

**MySQL Storage Engines:**

A storage engine is a software module that a database management system uses to create, read, update data from a database. There are two types of storage engines in MySQL: **transactional** and **non-transactional.** For MySQL 5.5 and later, the default storage engine is **InnoDB.** The default storage engine for MySQL prior to version 5.5 was **MyISAM**.

* InnoDB: widely used, with transaction support, ACID compliant storage engine. It supports row-level locking, crash recovery, multi-version concurrency control, (only engine) foreign key referential integrity constraint. Oracle recommends using InnoDB for tables
* MyISAM is the original storage engine. It is a fast storage engine. It does not support transactions. MyISAM provides table-level locking. It is used in Web and data warehousing.
* Memory storage engine creates tables in memory. It is the fastest engine. It provides table-level locking. It does not support transactions. Memory storage engine is ideal for creating temporary tables or quick lookups. The data is lost when the database is restarted.
* CSV stores data in CSV files. It provides great flexibility because data in this format is easily integrated into other applications.

**NoSQL database**

NoSQL databases (aka "not only SQL") are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.

**When should NoSQL be used:**

* When a huge amount of data needs to be stored and retrieved.
* The relationship between the data you store is not that important
* The data changes over time and is not structured.
* Support of Constraints and Joins is not required at the database level
* The data is growing continuously and the database has to be scaled regularly to handle the data.

|  |  |
| --- | --- |
| **NoSQL Database** | **Relational Database** |
| Supports a very simple query language. | Supports a powerful query language. |
| No fixed schema. | Has a fixed schema. |
| Only eventually consistent. | Follows acid properties. (atomicity, consistency, isolation, and durability) |
| Don't support transactions (support only simple transactions). | Supports transactions (also complex transactions with joins). |
| Used to handle data coming in high velocity. | Used to handle data coming in low velocity. |
| Data arrive from many locations. | Data arrive from one or few locations. |
| Can manage structured, unstructured and semi-structured data. | Manages only structured data. |
| Have no single point of failure. | Have a single point of failure with failover. |
| Can handle big data or data in a very high volume | Used to handle moderate volume of data. |
| Decentralized structure. | Has centralized structure. |
| Gives both read and write scalability. | Gives read scalability only. |
| Deployed in horizontal fashion. | Deployed in vertical fashion. |

**Features of NoSQL Database**

* They have higher scalability.
* They use distributed computing.
* They are cost effective.
* They support flexible schema.
* They can process both unstructured and semi-structured data.
* There are no complex relationships, such as the ones between tables in an RDBMS.

**CAP Theorem**

The CAP Theorem or Brewers Theorem states that it is not possible to guarantee all three of the desirable properties (only 2 is possible)– consistency, availability, and partition tolerance at the same time in a distributed system with data replication.

* **Consistency** – consistency refers to the consistency of the values in different copies of the same data item in a replicated distributed system. A guarantee that every node in a distributed cluster returns the same, most recent and a successful write.
* **Availability** – Availability means that each read or write request for a data item will either be processed successfully or will receive a message that the operation cannot be completed. every node (on either side of a network partition) must be able to respond in a reasonable time.
* **Partition Tolerance** – Partition tolerance means that the system can continue operating even if the network connecting the nodes has a fault that results in two or more partitions, where the nodes in each partition can only communicate among each other. the system continues to function and upholds its consistency guarantees in spite of network partitions.

**BASE Model**

* **Basic Availability.** The NoSQL database approach focuses on the availability of data even in the presence of multiple failures. It achieves this by using a highly distributed approach to database management. Instead of maintaining a single large data store and focusing on the fault tolerance of that store, NoSQL databases spread data across many storage systems with a high degree of replication. In the unlikely event that a failure disrupts access to a segment of data, this does not necessarily result in a complete database outage.
* **Soft State.** BASE databases abandon the consistency requirements of the ACID model pretty much completely. One of the basic concepts behind BASE is that data consistency is the developer's problem and should not be handled by the database.
* **Eventual Consistency**. The only requirement that NoSQL databases have regarding consistency is to require that at some point in the future, data will converge to a consistent state. No guarantees are made, however, about when this will occur.

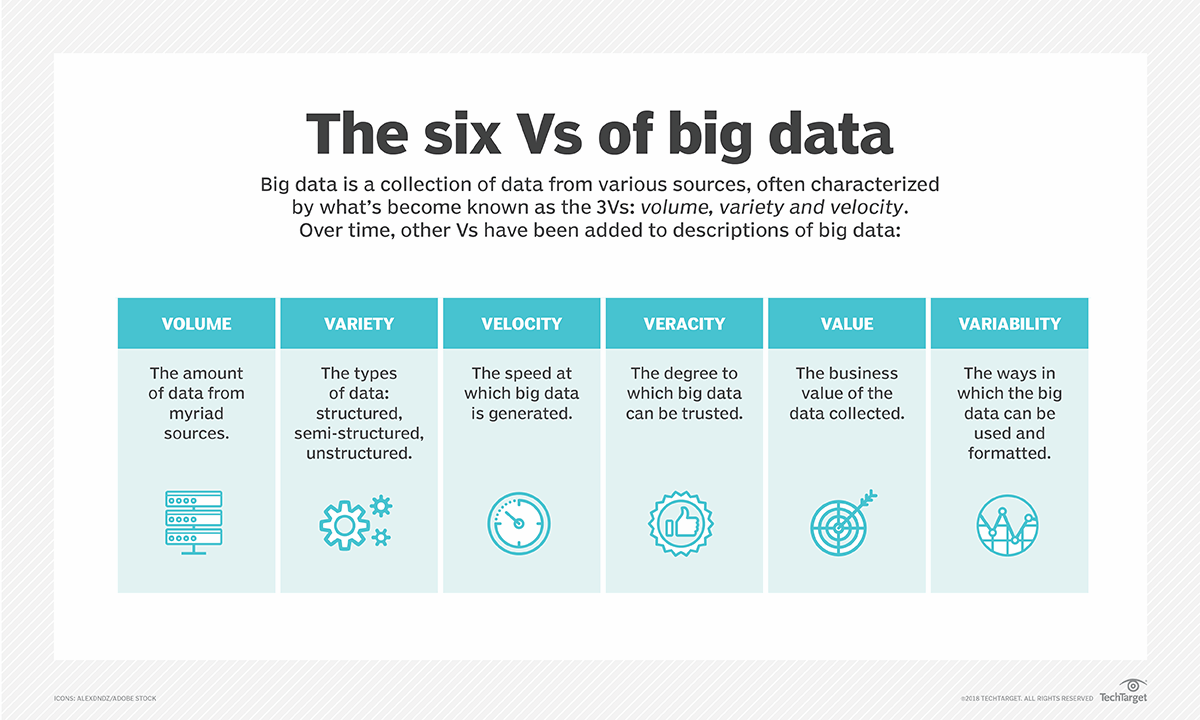
**Categories of NoSQL Databases:**

1. **Document-Based Database:** A document database stores data in JSON, BSON, or XML documents. The elements can be accessed by using the assigned index value for faster querying. **Collections** are the group of documents that store documents that have similar contents. Not all the documents are in any collection as they require a similar schema. Key features of documents database:
   1. **Flexible schema:** Documents in the database has a flexible schema.
   2. **Faster creation and maintenance**: the creation of documents is easy and minimal maintenance is required once we create the document.
   3. **No foreign keys:** There is no dynamic relationship between two documents so documents can be independent of one another.
   4. **Open formats:** To build a document we use XML, JSON, and others.
2. **Key-Value Stores:** it is the simplest form of a NoSQL database. Every data element in the database is stored in key-value pairs. The data can be retrieved by using a unique key of each element in the database. The values can be simple data types like strings and numbers or complex objects.
   1. **Simplicity. b. Scalability. C. Speed.**
3. **Column Oriented Databases:** it stores the data in columns instead of rows. Columnar databases are designed to read data more efficiently and retrieve the data with greater speed. A columnar database is used to store a large amount of data. Key features of columnar oriented database:
   1. **Scalability. b. Compression. c. Very responsive.**
4. **Graph-Based databases:** Graph-based databases focus on the relationship between the elements. It stores the data in the form of nodes in the database. The connections between the nodes are called links or relationships. Key features of graph database:
   1. It is easy to identify the relationship between the data by using the links.
   2. The Query’s output is real-time results.
   3. The speed depends upon the number of relationships among the database elements.

**JSON and BSON documents:**

JSON, or JavaScript Object Notation, is the wildly popular standard for data interchange on the web, on which BSON (Binary JSON) is based. But JSON is a text-based format, and text parsing is very slow, is not space-efficient, and only supports a limited number of basic data types. BSON was invented to bridge the gap: a binary representation to store data in JSON format, optimized for speed, space, and flexibility. BSON’s binary structure encodes type and length information, which allows it to be parsed much more quickly. It can store data types, like dates and binary data. It can save distinction between data types (int, long, etc.) and also allows for comparisons and calculations to happen directly on data.

**6 V’s of Big Data / Characteristics of Big Data:**



Properties of Big Data:

1. Debuggability
2. Scalability
3. Generalization
4. Ad hoc queries
5. Extensibility
6. Low latency reads and updates
7. Minimal Maintenance
8. Robustness and error tolerance