A non-relational database, often referred to as a NoSQL (Not Only SQL) database, is a type of database that does not follow the traditional relational database model used by SQL databases like MySQL, PostgreSQL, or Oracle. Instead, NoSQL databases are designed to handle large volumes of unstructured or semi-structured data and provide flexible schema design.

Here are some key characteristics and features of NoSQL databases:

Flexible Schema: Unlike relational databases, which require a predefined schema, NoSQL databases allow for dynamic and flexible schema design. This flexibility is particularly useful for handling data with varying structures and formats.

Scalability: NoSQL databases are often designed to scale horizontally, meaning they can handle increasing amounts of data by adding more servers or nodes to the database cluster. This makes them well-suited for handling Big Data applications and large-scale distributed systems.

High Performance: NoSQL databases are optimized for specific use cases, such as high throughput or low latency, which can result in better performance compared to traditional relational databases for certain workloads.

Data Models: NoSQL databases support various data models, including key-value stores, document stores, column-family stores, and graph databases. Each data model is optimized for specific types of data and access patterns.

CAP Theorem: NoSQL databases are often designed with trade-offs based on the CAP theorem, which states that it's impossible for a distributed system to simultaneously provide Consistency, Availability, and Partition tolerance. NoSQL databases typically prioritize either Consistency and Availability (CA) or Availability and Partition tolerance (AP).

Use Cases: NoSQL databases are commonly used in web applications, real-time analytics, content management systems, mobile apps, IoT (Internet of Things) applications, and other scenarios where scalability, flexibility, and high performance are essential.

Popular examples of NoSQL databases include MongoDB (document-oriented), Cassandra (column-family), Redis (key-value), Neo4j (graph), and Amazon DynamoDB (key-value and document). Each of these databases offers unique features and capabilities tailored to different use cases and requirements.