What is difference between relational database, non-relational database.

Relational Databases (RDBMS):

Data Structure:

- Structured Data: Relational databases organize data into tables with rows and columns. Each table represents an entity, and each row represents a record related to that entity.
- Schema: RDBMS follows a predefined schema, meaning the structure of the database, including tables, relationships, and constraints, must be defined before data can be inserted.

Data Integrity:

- ACID Properties: Relational databases adhere to ACID properties (Atomicity, Consistency, Isolation, Durability), which ensure transactions are reliable and maintain data integrity.

Scalability:

- Vertical Scaling: Traditionally, relational databases have scaled vertically by increasing the power of a single server. This can be limiting and costly.

Use Cases:

- Structured Data: RDBMS are suitable for scenarios where the data is highly structured and relationships between entities are well-defined.
- Transactions: They are often used in applications where data consistency and integrity are critical, such as financial systems.

Examples:

- MySQL, PostgreSQL, Oracle, SQL Server.

Non-Relational Databases (NoSQL):

Data Structure:

- Various Models: NoSQL databases can use various data models, including document-oriented (like MongoDB), key-value pairs, column-family, or graph databases.
- Dynamic Schema: NoSQL databases often have a dynamic schema, allowing for flexibility in adding fields to documents without the need to define a rigid structure beforehand.

Data Integrity:

- Eventual Consistency: NoSQL databases may sacrifice strict consistency in favor of performance and availability, providing eventual consistency.

Scalability:

- Horizontal Scaling: NoSQL databases are designed for horizontal scaling, which means adding more servers to a distributed database system to handle increased load.

Use Cases:

- Big Data: NoSQL databases are often used in scenarios with large amounts of unstructured or semi-structured data, like social media, log files, or real-time data.
- Scalability: They are well-suited for applications that need to scale horizontally to handle a high volume of read and write operations.

Examples:

- MongoDB, Cassandra, Redis, Couchbase.