SQL Vs NoSQL databases

SQL – Structured Query Language
NoSQL – Non-Structured Query Language

Picking a right database is a critical decision nowadays.

There are mainly two types of databases available –
Relational, Non-Relational

Differences?

1. Relational databases use Structured Query Language (select name, age from EMPLOYEE) on a predefined schema (EMPLOYEE table: name, age, gender, grade). Non-Relational databases have dynamic schema (like JSON).

2. SQL (Relational) databases are row-column based; No-SQL databases are document/key-value/graph/column-oriented.

3. SQL databases follow ACID properties, No-SQL databases are not suitable for ACID use cases.

When to use SQL?

- Fixed columns for every set of data – consistent structure.
- Transaction is a requirement – write-read consistency.
- Complex queries to be executed – joins, multiple table queries, conditionals.
- Predefined Schema is available – upfront decision.

When to use No-SQL?

- Dynamic schema, unstructured data, every set can contain different key-value pairs.
- Horizontally scalable by distributing the data on multiple servers. Sky is the limit :)
- provides a large number of users with simultaneous access; data may be eventual consistent (remember CAP theorem).
- Best suitable for hierarchical data storage.
- Huge and ever-growing data.

Comparison of features:

Feature	SQL	NoSQL
Performance	Low	High
Reliability	High	Low
Availability	Same	Depends*
Consistency	High	Depends*
Scalability	High but expensive	High

*CAP theorem

Examples of SQL databases:

- MySQL
- MariaDB
- Oracle
- PostgreSQL
- MSSQL

Examples of NoSQL databases:

- DynamoDB
- MongoDB
- Redis
- Cassandra
- Elasticsearch
- Firebase

Examples of systems built using SQL:

- Banking software
- Payment gateways storing your transactions
- **Comment if you can think of any example(s)!**

Examples of systems built using NoSQL:

- Uber storing your trip information
- Facebook/LinkedIn storing your connection information
- **Comment if you can think of any example(s)!**