**SQL vs NOSQL DBMS**

**1. Definition**

* **SQL DBMS (Relational Database Management System):**  
  A database system that stores data in structured tables with rows and columns and uses SQL (Structured Query Language) to manage and query the data.
* **NoSQL DBMS (Non-Relational Database Management System):**  
  A database system that stores data in formats other than relational tables, such as documents, key-value pairs, graphs, or wide-columns. It is designed for flexibility, scalability, and performance.

**2. Key Differences**

| **Feature** | **SQL DBMS** | **NoSQL DBMS** |
| --- | --- | --- |
| Data Model | Table-based (rows and columns) | Document, key-value, column-family, or graph |
| Schema | Fixed, predefined schema | Dynamic or schema-less |
| Query Language | SQL (Structured Query Language) | Various (e.g., JSON-based queries in MongoDB) |
| Transactions | ACID compliant | BASE (Eventually Consistent) |
| Scalability | Vertical (scale-up) | Horizontal (scale-out) |
| Best Use Case | Structured data, complex relationships | Large-scale unstructured or changing data |
| Examples | MySQL, PostgreSQL, Oracle, SQL Server | MongoDB, Cassandra, DynamoDB, Couchbase |

**3. When to Use**

* Use **SQL DBMS** when:
  + Your data is highly structured and relational.
  + You need strong data integrity and consistency.
  + You require complex queries, joins, and transactions.
* Use **NoSQL DBMS** when:
  + Your data is semi-structured or unstructured.
  + The schema may change frequently.
  + You need to store and process large volumes of data quickly across distributed systems.

**4. Simple Example**

**SQL (Relational):**

Table:

A group of names and numbers

AI-generated content may be incorrect.

Query:

SELECT \* FROM students WHERE age > 20;

**NoSQL (Document - MongoDB):**

{

"\_id": 1,

"name": "Alice",

"age": 21

}

Query:

db.students.find({ age: { $gt: 20 } })