1. Explain the difference between data and information.
Data refers to raw, unprocessed facts or figures, often in the form of numbers, text, or symbols.
Information, on the other hand, is processed or interpreted data that provides meaning and context, making it valuable for decision-making.

2. What is metadata?
Metadata is data that provides information about other data. It describes the structure, format, and characteristics of the data, helping to organize and interpret it.

3. What is a DBMS, and what are its advantages?
A DBMS (Database Management System) is collection of programs or software that provides an interface to store, retrieve, control access and manage data in a structured format.

4. Compare and contrast operational databases with analytical databases, and provide an example of each.
Operational databases (online transaction processing) are designed for everyday transactional processes (e.g., banking systems, online stores), where real-time data access and updates are crucial.
Analytical databases, or often referred to (Online analytical processing) are optimized for querying and analyzing large datasets (e.g., data warehouses, business intelligence systems).

5. Explain the types of data and use cases where NoSQL databases are most effective.
NoSQL databases are effective for unstructured or semi-structured data, such as documents, key-value pairs, or graphs.
They are well-suited for use cases with large-scale, high-velocity, or rapidly evolving data (e.g., social media, real-time analytics, IoT).

6. Which DBMS does not require server configuration, and what are the advantages of using it?
SQLite is a DBMS that does not require server configuration. It is lightweight, serverless, and stores data in a single file, making it easy to deploy and use in embedded systems and applications with minimal resources.

7. Explain ACID properties in the context of a DBMS.
ACID stands for Atomicity, Consistency, Isolation, and Durability. These properties ensure reliable transaction processing in a DBMS:
 Ensures a transaction is all-or-nothing, keeps the database correct, avoids conflicts between users, and makes sure changes are saved even if there's a crash.