

1. What database models do you know?
Hierarchical, Network, Relational, Object-Oriented, NoSQL
2. Which are the main functions performed by a Relational Database Management System (RDBMS)?
 - Data Storage, Retrieval, and Update (CRUD);
 - Data Integrity Management;
 - Data Security and Access Control
 - Transaction Management
 - Backup and Recovery
3. Define what is "table" in database terms.
Basic structure used to store and organize data in a relational database.
4. Explain the difference between a primary and a foreign key.
Primary key is a column of the table that uniquely identifies its rows
The foreign key is an identifier of a record located in another table (usually its primary key)
5. Explain the different kinds of relationships between tables in relational databases.
 - **One-to-many**: A single record in Table A can be related to multiple records in Table B, but a record in Table B can only be related to one record in Table A.
 - **Many-to-many**: Multiple records in Table A can be related to multiple records in Table B.
 - **One-to-one**: A single record in Table A is related to only one record in Table B, and vice versa.
6. When is a certain database schema normalized?
When its tables and columns have been organized to minimize data redundancy (duplication) and to protect data dependencies (ensuring data is logically stored).

7. What are the advantages of normalized databases?
 - Reduced Redundancy -> smaller database size and less chance of wasted space;
 - Improved Data Integrity -> fewer opportunities for data inconsistencies;
 - Easier Maintenance;
 - Faster and More Efficient Sorting/Indexing.
8. What are database integrity constraints and when are they used?

They are rules that prevent invalid data from being inserted, updated, or deleted from the database. They are used to ensure that the data stored meets certain business rules and logical standards.
9. Point out the pros and cons of using indexes in a database.

Pros: Faster data retrieval and sorting; Ensures uniqueness
Cons: Slower data modification; Increased storage
10. What's the main purpose of the SQL language?

Manage and manipulate data stored in a RDBMS.
11. What are transactions used for?

Transactions are used to group a set of database operations into a single, logical, and indivisible unit of work (Either all of them execute successfully or none of them are executed at all)
12. Give an example.

A bank transfer from one account into another (withdrawal + deposit)
If either the withdrawal or the deposit fail the entire operation should be cancelled.
13. What is a NoSQL database?

NoSQL databases are non-relational databases designed to handle diverse data types and large-scale applications (suitable for unstructured or semi-structured data).

14. Explain the classical non-relational data models.

- **Key-Value:** Data is stored as a massive dictionary or hash table where each item is stored with a key (unique identifier) and a value (the actual data, which can be anything).
- **Document:** Stores data as semi-structured "documents," often in formats like JSON, which can have dynamic schemas.
- **Column-Family (Wide-Column Store):** Organizes data into rows and columns, but with a flexible schema where columns can vary between rows within the same table
- **Graph Database:** Stores information as a collection of nodes and edges, where the edges represent the relationships between the nodes.

15. Give few examples of NoSQL databases and their pros and cons.

- **MongoDB** (Document DB):

Pros: Highly flexible schema; excellent horizontal scalability; intuitive for developers using JSON.

Cons: Transactions across multiple documents are complex; data redundancy is common.

- **Redis** (Key-Value):

Pros: Extremely fast (data is stored in-memory); great for caching, session management, and message queues.

Cons: Data size is limited by available memory; less suitable for complex, persistent data storage.

- **Apache Cassandra** (Column-Family):

Pros: Excellent linear scalability and fault tolerance; high availability for mission-critical applications.

Cons: Limited query flexibility (often requires pre-modeling data for specific queries); complex to set up and manage.

- **Neo4j** (Graph):

Pros: Optimized for querying relationships and connections; ideal for social networks, fraud detection, and recommendation engines.

Cons: Specialized use case; performance degrades for simple data retrieval; requires learning a new query language (Cypher).

