1. Q: What database models do you know?

A: There are different database models, but above all two main types mark most of the differences between databases – relational databases and non-relational databases. Some other types include hierarchical databases, network databases and object-oriented databases.

1. Q: Which are the main functions performed by a Relational Database Management System?

A: RDBMS are used to manage data, which is stored in tables with relations between them.

1. Q: Define what is “table” in database terms.

A: Tables consist of data, arranged in rows and columns, and in broad terms, they could be thought of as templates. Each row has the same structure and represents a single data entry, and each column has name and data type.

1. Q: Explain the difference between a primary and a foreign key.

A: A primary key is a column of the table that uniquely identifies its rows. Thus two records in the table are different if and only if their primary keys are different. The primary key type is usually a number and it is recommended that the primary key column has the only and single responsibility of holding that unique number.

On the contrary, a foreign key is just a pointer (an identifier) in a table of a record that is located in another table.

1. Q: Explain the different kinds of relationships between tables in relational databases.

A: There are three kinds of relationships between tables: one-to-many, many-to-many, and one-to-one.

One-to-many – when a single record in the first table corresponds with many records in the second table.

Many-to-many – when a single record in the first table corresponds with many records in the second table and vice versa.

One-to-one – when a single record in a table corresponds to a single record in the second table.

1. Q: When is a certain database schema normalized? What are the advantages of normalized databases?

A: When there are no duplicated data. Non-normalized database schemas contain data repetitions. The advantages are that data is isolated in smaller and less redundant tables, so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database using the defined relationships between tables.

1. Q: What are database integrity constraints and when are they used?

A: Integrity constraints ensure data integrity () in the database tables. These are constraints that enforce data rules which cannot be violated by data records. There are several types of database integrity constraints – entity integrity (concerning the concept of a primary key), referential integrity (concerning the concept of a foreign key), user-defined integrity (a set of rules defined by a user).

1. Q: Point out the pros and cons of using indexes in a database.

A: Indexes speed up searching of values in a certain column or group of columns, but on the other hand they slow down adding and deleting records in indexed tables. That is why indexes should only be used in large-scale tables of tens of thousands of records.

1. Q: What’s the main purpose of the SQL language?

A: SQL is designed for managing data held in a RDBMS. It comprises DDL and DML and its scope includes data insert, query, update and delete, schema creation and modification, and data access control.

1. Q: What are transactions used for? Give an example.

A: Transactions are a sequence of database operations that are executed as a single unit and are either executed successfully or none of the operations is executed at all. A transaction comprises a unit of work performed within a database management system against a database, and treated in a coherent and reliable way independent of other transactions.

Example: Double-entry accounting (account debit requires corresponding account credit): Debit $50 Expenses against Credit $50 Salaries. A database transaction would make either debit and credit operations pass or both entries would fail.

1. Q: What is a NoSQL database?

A: A NoSQL database is a non-relational, schema free, document-based database. It provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

1. Q: Explain the classical non-relational data models.

A: There are different types of non-relational data models:

* Document model – represents a set of documents, e.g. JSON strings, and each document stands for a single record.
* Key-value model: represents a set of key-value pairs;
* Hierarchical key-value – represents a hierarchy of key-value pairs;
* Wide-column – key-value model with corresponding schema;
* Object model – represents a set of object-oriented-style objects.

1. Q: Give few examples of NoSQL databases and their pros and cons.

A:

* Redis – ultra-fast in-memory data structures server. When the durability of data is not needed, the in-memory nature of Redis allows it to perform extremely well compared to database systems that write every change to disk before considering a transaction committed. There is no notable speed difference between write and read operations. Redis operates as a single process and single-threaded. Therefore a single Redis instance cannot utilize parallel execution of tasks e.g. stored procedures;
* MongoDB – mature and powerful JSON-document database. MongoDB can be used as a file system, taking advantage of load balancing and data replication features over multiple machines for storing files. MongoDB provides high availability with replica sets. A replica set consists of two or more copies of the data. Each replica set member may act in the role of primary or secondary replica at any time. The primary replica performs all writes and reads by default. Secondary replicas maintain a copy of the data on the primary using built-in replication.
* CouchDB – JSON-based document database with REST API. All items have a unique URI that gets exposed via HTTP. REST uses the HTTP methods POST, GET, PUT and DELETE for the four basic CRUD (Create, Read, Update, Delete) operations on all resources.
* Cassanda – wide-column model database. Cassandra's data model is a partitioned row store with tunable consistency. Rows are organized into tables; the first component of a table's primary key is the partition key; within a partition, rows are clustered by the remaining columns of the key. Other columns may be indexed separately from the primary key.

Have a nice day! ☺