Database Systems

1. What database models do you know?
   1. Hierarchical (tree)
   2. Network / graph
   3. Relational (table)
   4. Object-oriented
2. Which are the main functions performed by a Relational Database Management System (RDBMS)?
   1. Manage data stored in tables
3. Define what is "table" in database terms.
   1. Database tables consist of data, arranged in rows and columns
   2. All rows have the same structure
   3. Columns have name and type (number, string, date, image, or other)
4. Explain the difference between a primary and a foreign key.
   1. Primary key is a column of the table that uniquely identifies its rows (usually its is a number)
   2. Two records (rows) are different if and only if their primary keys are different
   3. The primary key can be composed by several columns (composite primary key)
5. Explain the different kinds of relationships between tables in relational databases.
   1. Relationships between tables are based on interconnections: primary key / foreign key
   2. The foreign key is an identifier of a record located in another table (usually its primary key)
   3. By using relationships we avoid repeating data in the database
   4. In the last example the name of the country is not repeated for each town (its number is used instead)
   5. Relationships have multiplicity:
   6. One-to-many – e.g. country / towns
   7. Many-to-many – e.g. student / course
   8. One-to-one – e.g. example human / student
6. When is a certain database schema normalized? What are the advantages of normalized databases?
   1. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them. The objective is to isolate data 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.
7. What are database integrity constraints and when are they used?
   1. Integrity constraints ensure data integrity in the database tables
   2. Enforce data rules which cannot be violated
8. Point out the pros and cons of using indexes in a database.
   1. Advantages: use an index for quick access to a database table specific information. The index is a structure of the database table the value of one or more columns to sort
      1. As a general rule, only when the data in the index column Frequent queries, only need to create an index on the table. The index take up disk space and reduce to add, delete, and update the line speed. In most cases, the speed advantages of indexes for data retrieval greatly exceeds it.
   2. Disadvantages: too index will affect the speed of update and insert, because it requires the same update each index file. For a frequently updated and inserted into the table, there is no need for a rarely used where the words indexed separately, small table, the cost of sorting will not be great, there is no need to create additional indexes. In some cases, the indexing words may not be fast, for example, the index is placed in a contiguous memory space, which will increase the burden of disk read, which is optimal, it should be through the actual use of the environment to be tested.
9. What's the main purpose of the SQL language?
   1. SQL ; Structured Query Language) is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS).
10. What are transactions used for? Give an example.
    1. Transactions are a sequence of database operations which are executed as a single unit:
    2. Either all of them execute successfully
    3. Or none of them is executed at all
    4. Example:
       1. A bank transfer from one account into another (withdrawal + deposit)
       2. If either the withdrawal or the deposit fails the entire operation should be cancelled
11. What is a NoSQL database?
    1. (non-relational) databases
    2. Use document-based model (non-relational)
    3. Schema-free document storage
    4. Still support CRUD operations  
       (create, read, update, delete)
    5. Still support indexing and querying
    6. Still supports concurrency and transactions
    7. Highly optimized for append / retrieve
    8. Great performance and scalability
    9. NoSQL == “No SQL” or “Not Only SQL”?
12. Explain the classical non-relational data models.
    1. Document model
       1. Set of documents, e.g. JSON strings
13. Give few examples of NoSQL databases and their pros and cons.
    1. Relational databases
       1. Data stored as table rows
       2. Relationships between related rows
       3. Single entity spans multiple tables
       4. RDBMS systems are very mature, rock solid
    2. NoSQL databases
       1. Data stored as documents
       2. Single entity (document) is a single record
       3. Documents do not have a fixed structure