

Database Systems - Overview

1. What database models do you know?

- A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized, and manipulated. The most popular example of a database model is the relational model, which uses a table-based format.
- Common logical data models for databases include:
 - Record base models
 - Hierarchical database model - In this model each entity has only one parent but can have several children . At the top of hierarchy there is only one entity which is called Root.
 - Network model - In the network model, entities are organised in a graph,in which some entities can be accessed through sveral path
 - Relational model - In this model, data is organised in two-dimesional tables called relations. The tables or relation are related to each other.
 - Entity–relationship model
 - Enhanced entity–relationship model
 - Object-oriented model
 - Document model
 - Entity–attribute–value model

2. Which are the main functions performed by a Relational Database Management System (RDBMS)?

- A relational database management system (RDBMS) is a program that lets you create, update, and administer a relational database. Most commercial RDBMS's use the Structured Query Language (SQL) to access the database, although SQL was invented after the development of the relational model and is not necessary for its use.
- Main functions
 - Provides data to be stored in tables

- Persists data in the form of rows and columns
- Provides facility primary key, to uniquely identify the rows
- Creates indexes for quicker data retrieval
- Provides a virtual table creation in which sensitive data can be stored and simplified query can be applied.(views)
- Sharing a common column in two or more tables(primary key and foreign key)
- Provides multi user accessibility that can be controlled by individual users

3. **Define what is "table" in database terms.**

- A table is a collection of related data held in a structured format within a database. It consists of fields (columns), and rows. A table has a specified number of columns, but can have any number of rows. Each row is identified by one or more values appearing in a particular column subset. The columns subset which uniquely identifies a row is called the primary key. Table is another term for relation; although there is the difference in that a table is usually a multiset (bag) of rows where a relation is a set and does not allow duplicates. Besides the actual data rows, tables generally have associated with them some metadata, such as constraints on the table or on the values within particular columns.

4. **Explain the difference between a primary and a foreign key.**

- Primary Key - Primary key uniquely identify a record in the table.
- Foreign Key - Foreign key is a field in the table that is primary key in another table.
- Primary Key - Primary Key can't accept null values.
- Foreign Key - Foreign key can accept multiple null value.
- Primary Key - By default, Primary key is clustered index and data in the database table is physically organized in the sequence of clustered index.
- Foreign Key - Foreign key do not automatically create an index, clustered or non-clustered. You can manually create an index on foreign key.
- Primary Key - We can have only one Primary key in a table.
- Foreign Key - We can have more than one foreign key in a table.

5. **Explain the different kinds of relationships between tables in relational databases.**

- One-to-one: Both tables can have only one record on either side of the relationship. Each primary key value relates to only one (or no) record in the related table. They're like spouses—you may or may not be married, but if you are, both you and your spouse have only one spouse. Most one-to-one relationships are forced by business rules and don't flow naturally from the data. In the absence of such a rule, you can usually combine both tables into one table without breaking any normalization rules.
- One-to-many: The primary key table contains only one record that relates to none, one, or many records in the related table. This relationship is similar to the one between you and a parent. You have only one mother, but your mother may have several children.
- Many-to-many: Each record in both tables can relate to any number of records (or no records) in the other table. For instance, if you have several siblings, so do your siblings (have many siblings). Many-to-many relationships require a third table, known as an associate or linking table, because relational systems can't directly accommodate the relationship.

6. **When is a certain database schema normalized? What are the advantages of normalized databases?**

Database normalization is the process of organizing the attributes and tables of a relational database to minimize data redundancy. Normalization involves decomposing a table into less redundant (and smaller) tables but without losing information; defining foreign keys in the old table referencing the primary keys of the new ones. The objective is to isolate data so that additions, deletions, and modifications of an attribute can be made in just one table and then propagated through the rest of the database using the defined foreign keys.

Advantages

- Smaller database: By eliminating duplicate data, you will be able to reduce the overall size of the database.
- Better performance: Fewer indexes per table mean faster maintenance tasks such as index rebuilds. Only join tables that you need.

- Narrow tables: Having more fine-tuned tables allows your tables to have less columns and allows you to fit more records per data page.

7. **What are database integrity constraints and when are they used?**

- Integrity constraints are used to ensure accuracy and consistency of data in a relational database. Data integrity is handled in a relational database through the concept of referential integrity. Many types of integrity constraints play a role in referential integrity (RI).
 - Primary Key Constraints
 - Unique Constraints
 - Foreign Key Constraints
 - NOT NULL Constraints
 - Check Constraints
 - Dropping Constraints

8. **Point out the pros and cons of using indexes in a database.**

- Advantages:

Faster lookup for results. This is all about reducing the # of Disk IO's. Instead of scanning the entire table for the results, you can reduce the number of disk IO's(page fetches) by using index structures such as B-Trees or Hash Indexes to get to your data faster.

- Disadvantages:

Slower writes(potentially). Not only do you have to write your data to your tables, but you also have to write to your indexes. This may cause the system to restructure the index structure(Hash Index, B-Tree etc.), which can be very computationally expensive.

Takes up more disk space, naturally. You are storing more data.

9. **What's the main purpose of the SQL language?**

Purpose of SQL Structured Query Language is to provide a Structured way by which one can Query information in database using a standard Language.

SQL provides a global standard of working with databases with little or not differences over different platform. F or e.g. if you are familiar with SQL you can work with major DBs like SQL Server mySql & Oracle few minor differences in

syntax exists but they aren't very prominent at least as far as basic operations are concerned.

10. What are transactions used for? Give an example.

- A transaction is a set of changes that must all be made together. Transaction is executed as a single unit. If the database was in consistent state before a transaction, then after execution of the transaction also, the database must be in a consistent state.
- For example, a transfer of money from one bank account to another requires two changes to the database both must succeed or fail together.

If you send 100\$ from your bank account (A) to another account (B) there will be two operations.

```
Withdraw the money from the account A  
Add the money to account B
```

So if one of the two operations fail, both of them should fail.

11. What is a NoSQL database?

A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. Motivations for this approach include simplicity of design, horizontal scaling, and finer control over availability. The data structures used by NoSQL databases (e.g. key-value, graph, or document) differ from those used in relational databases, making some operations faster in NoSQL and others faster in relational databases.

12. Explain the classical non-relational data models. The non-relational data model would look more like a sheet of paper. In fact, the concept of one entity and all the data that pertains to that one entity is known in Mongo as a "document", so truly this is a decent way to think about it. You just add things whenever you need them. Your software doesn't need to know ahead of time, you don't need to know ahead of time. It all just kind of works, provided you know what you're doing on the software level.

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

- MongoDB. Open-source document database.

- CouchDB. Database that uses JSON for documents, JavaScript for MapReduce queries, and regular HTTP for an API.
- Redis. Data structure server wherein keys can contain strings, hashes, lists, sets, and sorted sets.
- Cassandra. Database that provides scalability and high availability without compromising performance. memcached. Open source high-performance, distributed-memory, and object-caching system.