Database systems – Overview – Homework

1. What database models do you know?

Answer:

- Hierarchical (tree)
- Network / graph
- Relational (table)
- Object-oriented

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

- Represent the data to the user as relations (a presentation in tabular form, i.e. as a collection of tables with each table consisting of a set of rows and columns);
- Provide relational operators to manipulate the data in the tables

3. Define what is "table" in databaseterms.

Physical representation of an entity or object that is in a tabular format consisting of columns and rows. Columns are the fields of a record or the attributes of an entity. The rows contain the values or data instances; these are also called records. Each row is identified by primary key which is unique for the row.

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

Primary key is a column of the table that uniquely identifies its rows (usually it is a number). 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.

5.1. Relationship one-to-one

✓ A single record in a table corresponds to a single record in the other table.

✓ Used to model inheritance between tables.

5.2. Relationship one-to-many (or many-to-one)

✓ Asingle record in the first table has many corresponding records in the second table

- ✓ Used very often

5.3. Relationship many-to-many

- ✓ Records in the first table have many corresponding records in the second one and vice versa.
- ✓ Implemented through additional table.

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

6.1. When is a certain database schema normalized?

A database should be normalized when there is excessive duplicate data (redundancy) in it. Such redundancy potentially leads to slow operations, data anomalies and corruption.

6.2. What are the advantages of normalized databases?

- ✓ Minimize redesign when extending the database structure.
- ✓ Index searching is often faster, since indexes tend to be narrower and shorter.
- ✓ Searching, sorting, and creating indexes is faster since tables are narrower and more rows fit on a data page
- ✓ Data modification anomalies are reduced.
- ✓ Make the data model more informative to users.
- ✓ Usually there are fewer indexes per table, so data modification commands are faster.

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

Integrity constraints ensure data integrity in the database tables and enforce data rules which cannot be violated.

- ✓ Primary key constraint

Each table can have one primary key, which uniquely identifies each row in a table and ensures that no duplicate rows exist.

- ✓ Unique key constraint

Ensures that all values in a certain column (or a group of columns) are unique. Some examples of good unique keys include:

- o An employee's social security number (the primary key is the employee id).
- o A truck's license plate number (the primary key is the truck id).

- ✓ Foreign key constraint

Ensures that the value in given column is a key from another table.

- ✓ Check constraint

Ensures that values in a certain column meet some predefined condition.

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

Pros:

- o Speed up searching of values in a certain column or group of columns.

Cons:

- o Adding and deleting records in indexed tables is slower.

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

SQL Language is designed for managing data in relational databases. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

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

When a group of logically inseparable operations have to be performed together over data in a database. If one of the operations fails than the data gets to the initial state and the rest of operations are not executed.

11. What is a NoSQL database?

NoSQL or Not Only SQL 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 structure (e.g. key-value, graph, or document) differs from the RDBMS, and therefore some operations are faster in NoSQL and some in RDBMS. NoSQL databases are increasingly used in big data and real-time web applications. NoSQL systems are also called "Not only SQL" to emphasize that they may also support SQL-like query languages.

12. Explain the classical non-relational data models.

Document-oriented

A document-oriented database provides a semi-structured representation for nested data. It works in a similar fashion to column-based ones; however, they allow much deeper nesting and complex structures to be achieved (e.g. a document, within a document, within a document). Documents overcome the constraints of one or two level of key / value nesting of columnar databases. Basically, any complex and arbitrary structure can form a document, which can be stored using these management systems. Despite their powerful nature, and the ability to query records by individual keys, document based management systems have their own issues and downfalls compared to others. For example, retrieving a value of a record means getting the whole lot of it and same goes for updates, all of which affect the performance.

Example: MongoDB - open source document-oriented data store with a Binary Object Notation (BSON) storage format that is JSON-style and familiar to web developers.

Key-value Pair

Works by matching keys with values, similar to a dictionary. There is no structure nor relation. After connecting to the database server (e.g. Redis), an application can state a key (e.g. the_answer_to_life) and provide a matching value (e.g. 42) which can later be retrieved the same way by supplying the key. Key / value DBMSs are usually used for quickly storing basic information, and sometimes not-so-basic ones after performing, for example, a CPU and memory intensive computation. They are extremely performant, efficient and usually easily scalable. Example: Redis - open source in-memory key-value pair data store.

Column-family table

These databases work very simply by creating collections of one or more key / value pairs that match a record. Unlike the traditional defines schemas of relational databases, column-based NoSQL solutions do not require a pre-structured table to work with the data. Each record comes with one or more

columns containing the information and each column of each record can be different. Examples: Apache HBase, Apache Cassandra.

Graph

A graph database represents and stores data in three aspects: nodes, edges, and properties. A node is an entity, such as a person or business. An edge is the relationship between two entities. For example, an edge could represent that a node for a person entity is an employee of a business entity. A property represents information about nodes. For example, an entity representing a person could have a property of "female" or "male". Similarly to mathematics, certain operations are much simpler to perform using these type of models thanks to their nature of linking and grouping related pieces of information (e.g. connected people). These databases are commonly used by applications whereby clear boundaries for connections are necessary to establish. For example, when you register to a social network of any sort, your friends' connection to you and their friends' friends' relation to you are much easier to work with using graph-based database management systems.

Graph data stores: Neo4J, OrientDB, Cayley, Titan

Object Databases

A cleaner alternative to ORM are object-oriented databases that directly add database functionality to the object oriented model and language itself. Object persistence becomes almost native and application code is tremendously simplified. Some implementations are db4o, Cache and Jade, just to cite a few. Once you use an object database, you realize how inadequate the relational model is and you will try never go back to it.

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

MongoDB

Mature and powerful JSON-document database.

Redis

Ultra-fast in-memory data structures server.

CouchDB

JSON-based document database with REST API.

Cassandra

Distributed wide-column database.