

## 1. What database models do you know?

- Hierarchical model
- Hierarchical key-value model
- Network model
- Relational (table) model
- Object model
- Document model
- Associative (Key-value) model
- Entity–relationship model
- etc ...

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

- Database management system is designed to manage a database.
- RDBMS Systems are very mature, rock solid.

### ➔ Popular RDBMS Servers:

- Microsoft SQL Server
  - Microsoft Access
  - Oracle MySQL
  - Database
  - IBM DB2
  - PostgreSQL (Open-source cloning of Oracle)
  - SQLite (.dll - Database Server / .db - DB data)
  - Sybase ASE
- Using for management of relational data stored in tables.
  - Definition of relational schema (database schema).
  - Creating, modifying, deleting tables and relationships between them.
  - Adding, modifying, deleting, searching and retrieving data stored in tables.
  - SQL Language Support.
  - Management (maintenance) of transactions.

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

- Table represents the structure the data will be stored. Table stores information organized in rows and columns.
- Row represents information about one record.
- Column represents a piece information / characteristic about certain record. Column has name and type.

### ➔ Schema of table (example):

```
Persons (  
    Id: number,  
    FirstName: string,  
    LastName: string,  
    Employer: string  
)
```

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

- **Primary key** (Composite primary key) is a column(s) of the table that uniquely identifies the rows (records). Primary key is unique - meets only one time in certain column.
- The purpose of **Foreign key** is to avoid data duplicates in table columns. The repeating data is separate in new table as each record has primary key used in the master table. Foreign key is not unique - can be used many times in certain column (usually number instead strings e.g.).

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

- **One-to-many** – e.g. country -> towns (Many towns in one country)
- **Many-to-many** – e.g. students <-> courses (Many students in many courses and conversely) - Implemented through additional table
- **One-to-one** – e.g. example human <- student (Using Inheritance)
- **Self-relationship** - e.g. 1. Root <- 2. Documents (ParentId: 1) <- 3. Pictures (1) <- 4. Birthday Party (3)

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

- Normalization removes repeating data from certain column or group of columns.
- Separate repeating values from certain column to a new table (Master table) and replace old repeating values to Details table with their Primary keys from the Master table.
- Avoiding data duplication using unique data identifier (Primary key - usually number) from the new table (Master table).

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

- Ensure data integrity in the database tables
- Enforce data rules which cannot be violated

➔ Used for Primary key:

- Ensures unique value for each table row

➔ Used for Unique key:

- Ensures that all values in a certain column are unique

➔ Used for Foreign key:

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

➔ Used for Check constraint (data restriction):

- 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:

- Faster lookup for results in certain column or group of columns (using index structures such as B-Trees or Hash Indexes to speed up searching of values).
- Instead of scanning the entire table for the results.

➔ Cons:

- Slower writes - adding / deleting records in indexed tables is slower.
- May cause the system to restructure the index of structure (Hash Index, B-Tree, etc), which can be very computationally expensive.
- Takes up more disk space - stores more data.

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

- ➔ DDL - Data Definition Language:
  - Creating, altering, deleting tables and other objects in the database.
- ➔ DML - Data Manipulation Language:
  - Searching, retrieving, inserting, modifying and deleting table data (rows).

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

- Used of competitive data access.
- Sequence of operations executing as a single unit.
- Can be rolled back if they are not completed properly.

## 11. What is a NoSQL database?

- Use document-based model (non-relational)
- Data stored as documents
- Single entity (document) is a single record
- Documents do not have a fixed structure

## 12. Explain the classical non-relational data models.

- A non-relational database is a database that does not incorporate the table/key model that relational database management systems (RDBMS) promote.
- These kinds of databases require data manipulation techniques and processes designed to provide solutions to big data problems that big companies face.
- The most popular emerging non-relational database is called NoSQL (Not Only SQL).

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

- ➔ Databases:
  - Cassandra (Distributed wide-column database)
  - MongoDB (Mature and powerful JSON-Documents database)
  - CouchDB (JSON-based document database with REST API)
  - Redis (Ultra-fast in-memory data structures server)
  - H-Base
- ➔ Models:
  - Document model
  - Associative (Key-value) model
  - Hierarchical key-value model
  - Wide-column model
  - Object model
- ➔ Pros:
  - Support CRUD operations
  - Support Indexing and querying
  - Support concurrency and transactions
  - Highly optimized for append / retrieve
  - Great performance and scalability
- ➔ Cons:
  - Difficult administration and support
  - etc ...