

# TOP INTERVIEW QUESTIONS

## DBMS AND SQL SERVER

 @himanshu\_shekhar16

### What is a database?

A database is described as an organized way of collection of DATA. It is the collection of schemes, tables, queries, reports, views and other objects.

Syntax: `CREATEDATABASE DatabaseName`

Example: `CREATEDATABASE Student`

or you can Create Database through Design/ Wizard form by right clicking on DATABASE option- New Database.

### What is DBMS?

A Database Management System (DBMS) is a program that controls creation, maintenance and use of a database. DBMS can be termed as File Manager that manages data in a database rather than saving it in file systems.

### What is SQL?

Structured Query Language, also known as SQL, is a programming language designed for managing Relational Database Management Systems (RDBMSs). SQL is an International Organization for Standardization (ISO) standard. In RDBMS all the data is stored in tables with each table consisting of rows and columns. Example of Sql Server 2014 SQL format:

## What is RDBMS?

RDBMS: It is referred as Relation Database Management Systems (RDBMS). RDBMS possesses a set of the below given characteristics:

- Write-intensive operations: The RDBMS is frequently written to and is often used in transaction-oriented applications.
- Data in flux or historical data: The RDBMS is designed to handle frequently changing data. Alternatively, RDBMS can also store vast amounts of historical data, which can later be analyzed or "mined".
- Application-specific schema: The RDBMS is configured on a per-application basis and a unique schema exists to support each application.
- Complex data models. The relational nature of the RDBMS makes it suitable for handling sophisticated, complex data models that require many tables, foreign key values, complex join operations, and so on.
- Data integrity: The RDBMS features many components designed to ensure data integrity. This includes rollback operations, referential integrity, and transaction-oriented operations.

## What is a database table?

Database table: Table contains records in the form of rows and columns. A permanent table is created in the database you specify and remains in the database permanently, until you delete it.

Syntax:

1. Create table TableName (ID INT, NAME VARCHAR(30) )
2. Drop syntax: drop table TableName
3. Select Syntax: Select \* from TableName

## What is a query?

A DB query is a code written in order to get the information back from the database. Query can be designed in such a way that it matched with our expectation of the result set. Simply, a question to the Database.

## What is subquery?

A subquery is a query within another query. The outer query is called as main query, and inner query is called subquery. SubQuery is always executed first, and the result of subquery is passed on to the main query.

## What are the types of subquery?

There are two types of subquery – Correlated and Non-Correlated.

A correlated subquery cannot be considered as independent query, but it can refer the column in a table listed in the FROM the list of the main query.

A Non-Correlated sub query can be considered as independent query and the output of subquery are substituted in the main query.

## **How to create a table in SQL?**

SQL provides an organized way for table creation.

Syntax: Create table TableName (columnName1 datatype, columnName2 datatype )

The following is an example of creating a simple table-

```
create table Info (Name varchar(20), BirthDate date, Phone nvarchar(12), City varchar(20))
```

## **What are tables and Fields?**

A table is a set of data that are organized in a model with Columns and Rows. Columns can be categorized as vertical, and Rows are horizontal. A table has specified number of column called fields but can have any number of rows which is called record.

## **How to delete a table in SQL Server?**

Delete Data Record from Database Table and deleting an existing table by the following method:

Syntax: To delete all table records of a table:

```
Delete TableName DELETE info
```

## **How to update a database table using SQL?**

To update an existing Table we use SQL Command UPDATE: It will update the records as per user defined query/need.

Syntax:

```
Update TableName SET ColumnName = NewData where Condition
```

```
Update info Set City = 'Baroda' where id = 2
```

## **What is a database relationship?**

Relationships are created by linking the column in one table with the column in another table. There are four different types of relationship that can be created.

The relationships are listed below:

1. One to One Relationship
2. Many to One relationship
3. Many to Many relationship
4. One to One relationship

One to Many & Many to One Relationship:

For a One to many relationships, a single column value in one table has one or more dependent column values in another table. Look at the following diagram:

Many to Many Relationship:

The third table acts as a bridge between the tables that want to establish a Many to Many relationship. The bridge table stores the common information between Many to Many relationship tables. Have a look at the following diagram:

## **What is a primary key of a database?**

Primary key:-

A table column with this constraint is called the key column for the table. This constraint helps the table to make sure that the value is not repeated and also that there are no null entries.

Now this column does not allow null values and duplicate values. You can try inserting values to violate these conditions and see what happens. A table can have only one Primary key. Multiple columns can participate on the primary key.

Example:

## **What is a unique key?**

A Unique key constraint uniquely identified each record in the database. This provides uniqueness for the column or set of columns. A Primary key constraint has automatic unique constraint defined on it. But not, in the case of Unique Key. There can be many unique constraint defined per table, but only one Primary key constraint defined per table.

## **What is a foreign key of a database?**

To define the relationship between two tables (one is called parent and the other one is the child table) connected by columns, a foreign key constraint is used. In this constraint the values of the child table must appear in the parent table, which means that for a foreign key, one table should point to a Primary Key in another table. A table can have multiple foreign keys and each foreign key can have a different referenced table.

Example: To understand the foreign key clearly let's assume the following two tables:

CUSTOMER {Cust\_ID, Cust\_Name, Age, ContactNo, Gender, Address} VENDOR {Vend\_ID, Vend\_Name, Cust\_ID}

Example: Foreign Key Constraint while using CREATE TABLE statement.

Syntax: CREATE TABLE table\_name(Col1 datatype NOT NULL, Col2 datatype NOT NULL, Col3 datatype NOT NULL, CONSTRAINT FK\_Column FOREIGN KEY(Col1, Col2, Col3) REFERENCES parent\_table(Col1, Col2, Col3) );

AT SINGLE COLUMN LEVEL

## What is database normalization?

Database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships among them. Normalization is a bottom-up technique for database design.

The evolution of Normalization theories is illustrated below:

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Codd Normal Form (BCNF)
- 4th Normal Form
- 5th Normal Form
- 6th Normal Form

## What are database normalization forms?

Normalization is the process of organizing data into a related table. it also eliminates redundancy and increases the integrity which improves performance of the query. To normalize a database, we divide the database into tables and establish relationships between the tables.

- First Normal Form (1st NF)
- Second Normal Form (2nd NF)
- Third Normal Form (3rd NF)
- Boyce-Codd Normal Form (BCNF)
- Fourth Normal Form (4th NF)
- Fifth Normal Form (5th NF)

First Normal Form (1NF):

This should remove all the duplicate columns from the table. Creation of tables for the related data and identification of unique columns.

Second Normal Form (2NF):

Meeting all requirements of the first normal form. Placing the subsets of data in separate tables and Creation of relationships between the tables using primary keys. Third Normal Form (3NF):

This should meet all requirements of 2NF. Removing the columns which are not dependent on primary key constraints.

Fourth Normal Form (3NF):

Meeting all the requirements of third normal form and it should not have multi-valued dependencies.

## **What is Denormalization.**

DeNormalization is a technique used to access the data from higher to lower normal forms of database. It is also process of introducing redundancy into a table by incorporating data from the related tables.

## **What is a stored procedure?**

A Stored Procedure is a collection or a group of T-SQL statements. Stored Procedures are a precompiled set of one or more statements that are stored together in the database. They reduce the network load because of the precompilation. We can create a Stored Procedure using the "Create proc" statement.

## **Why we use Stored Procedure**

There are several reasons to use a Stored Procedure. They are a network load reducer and decrease execution time because they are precompiled. The most important use of a Stored Procedure is for security purposes. They can restrict SQL Injection. We can avoid SQL injection by use of a Stored Procedure.

## **How to create a Stored Procedure**

```
CREATE PROCEDURE spEmployee AS
```

```
BEGIN
```

```
SELECT EmployeeId, Name, Gender, DepartmentName FROM tblEmployees INNER JOIN  
tblDepartments ON tblEmployees.EmployeeDepartmentId = tblDepartments.DepartmentId
```

```
END
```

Advantages of using a Stored Procedure in SQL Server

- It is very easy to maintain a Stored Procedure and they are re-usable.
- The Stored Procedure retains the state of the execution plans.
- Stored Procedures can be encrypted and that also prevents SQL Injection Attacks

## What is a function in SQL Server?

A function is a sequence of statements that accepts input, processes them to perform a specific task and provides the output. Functions must have a name but the function name can never start with a special character such as @, \$, #, and so on.

Types of function

- Pre-Defined Function
- User-Defined Function

User-defined Function:

In a user-defined function we write our logic according to our needs. The main advantage of a user-defined function is that we are not just limited to pre-defined functions. We can write our own functions for our specific needs or to simplify complex SQL code. The return type of a SQL function is either a scalar value or a table.

Creation of a function

Create function ss(@id int)

returns table as return select \* from item where itemId = @id

Execution of a Function

select \* from ss(1)

Output:

## What are the different types of functions in SQL Server?

A function must return a result. So that is also called a function that returns a result or a value. When we create it a function must specify a value type that will return a value.

- Functions only work with select statements.
- Functions can be used anywhere in SQL, such as AVG, COUNT, SUM, MIN, DATE and so on with select statements.
- Functions compile every time.
- Functions must return a value or result.
- Functions only work with input parameters.

- Try and catch statements are not used in functions.

Function Types:

The following is the function list in SQL Server databases.

SQL Server contains the following aggregates functions:

## **What is a view in the database?**

A View is nothing but a select query with a name given to it or we can simply say a view is a Named Query. Ok! Why do we need a view? There can be many answers for this. Some of the important stuff I feel is:

- A view can combine data from multiple tables using adequate joins and while bringing it may require complex filters and calculated data to form the required result set. From a user's point of view, all these complexities are hidden data queried from a single table.
- Sometimes for security purposes, access to the table, table structures and table relationships are not given to the database user. All they have is access to a view not knowing what tables actually exist in the database.
- Using the view, you can restrict the user update only portions of the records.

The following are the key points to be noted about views:

1. Multiple views can be created on one table.
2. Views can be defined as read-only or updatable.
3. Views can be indexed for better performance.
4. Insert, update, delete can be done on an updatable view.

## **What is the difference between Primary key and unique key?**

Primary key does not allow the null values but unique key allows one null value.

Primary key will create clustered index on column but unique key will create non-clustered index by default.

## **How can you increase SQL performance?**

Following are tips which will increase your SQL performance:-

Every index increases the time takes to perform INSERTS, UPDATES, and DELETES, so the number of indexes should not be too much. Try to use maximum 4-5 indexes on one table, not more. If you have read-only table, then the number of indexes may be increased.

Keep your indexes as narrow as possible. This reduces the size of the index and reduces the number of reads required to read the index.



Try to create indexes on columns that have integer values rather than character values.

If you create a composite (multi-column) index, the orders of the columns in the key are very important. Try to order the columns in the key as to enhance selectivity, with the most selective columns to the leftmost of the key.

If you want to join several tables, try to create surrogate integer keys for this purpose and create indexes on their columns. Create surrogate integer primary key (identity for example) if your table will not have many insert operations.

Clustered indexes are more preferable than nonclustered, if you need to select by a range of values or you need to sort results set with GROUP BY or ORDER BY. If your application will be performing the same query over and over on the same table, consider creating a covering index on the table.

You can use the SQL Server Profiler Create Trace Wizard with "Identify Scans of Large Tables" trace to determine which tables in your database may need indexes. This trace will show which tables are being scanned by queries instead of using an index.

## **What is RAID and how does it work?**

Redundant Array of Independent Disks (RAID) is a term used to describe the technique of improving data availability through the use of arrays of disks and various data-striping methodologies. Disk arrays are groups of disk drives that work together to achieve higher data-transfer and I/O rates than those provided by single large drives. An array is a set of multiple disk drives plus a specialized controller (an array controller) that keeps track of how data is distributed across the drives. Data for a particular file is written in segments to the different drives in the array rather than being written to a single drive.

For speed and reliability, it is better to have more disks. When these disks are arranged in certain patterns and are use a specific controller, they are called a Redundant Array of Inexpensive Disks (RAID) set. There are several numbers associated with RAID, but the most common are 1, 5 and 10.

## **What is the difference between DELETE TABLE and TRUNCATE TABLE commands?**

DELETE TABLE syntax logs the deletes thus make the delete operation slow. TRUNCATE table does not log any information but it logs information about deallocation of data page of the table so TRUNCATE table is faster as compared to delete table.

DELETE table can have criteria while TRUNCATE cannot.

TRUNCATE table does not invoke trigger.

I had mentioned that TRUNCATE table can not be rolled back while delete can be.

## **What is SQL injection?**

It is a Form of attack on a database-driven Web site in which the attacker executes unauthorized SQL commands by taking advantage of insecure code on a system connected to the Internet,

bypassing the firewall. SQL injection attacks are used to steal information from a database from which the data would normally not be available and/or to gain access to an organization's host computers through the computer that is hosting the database.

SQL injection attacks typically are easy to avoid by ensuring that a system has strong input validation.

As name suggest we inject SQL which can be relatively dangerous for the database. Example this is a simple SQL

```
SELECT email, passwd, login_id, full_name FROM members WHERE email = 'x'
```

Now somebody does not put "x" as the input but puts "x ; DROP TABLE members;". So the actual SQL which will execute is:-

```
SELECT email, passwd, login_id, full_name FROM members WHERE email = 'x'; DROP TABLE members;
```

Think what will happen to your database.

Feel Free to connect with me –

<https://www.linkedin.com/in/himanshushekhar16/>