***Mostly Asked 15 SQL Interview questions and Answers:***

**Mostly asked questions (15):**

1. What are tables and fields in database?

A table has records (rows) and fields (columns). Fields have different types of data, such as text, numbers, dates, and hyperlinks. A record: Contains specific data, like information about a particular employee or a product.

1. What is the difference between a primary key, foreign key and Unique Key?

A primary key is used to ensure data in the specific column is unique. A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It uniquely identifies a record in the relational database table. ... Only one primary key is allowed in a table.

Primary Key

Primary key cannot have a NULL value.

Each table can have only one primary key.

By default, Primary key is clustered index, and the data in database table is physically organized in the sequence of clustered index.

Primary key can be related to another tables as a Foreign Key.

We can generate ID automatically with the help of Auto Increment field. Primary key supports Auto Increment value.

We can define Primary key constraint on temporary table and table variable.

We can't delete primary key value from the parent table which is used as a foreign key in child table. To delete we first need to delete that primary key value from the child table.

Unique Key

Unique Constraint may have a NULL value.

Each table can have more than one Unique Constraint.

By default, Unique key is a unique non-clustered index.

Unique Constraint cannot be related with another table as a Foreign Key.

Foreign Key

Foreign key is a field in the table that is Primary key in another table.

Foreign key can accept multiple null value.

Foreign key does not automatically create an index, clustered or non-clustered. You can manually create an index on foreign key.

We can have more than one foreign key in a table.

Foreign keys do not automatically create an index, clustered or non-clustered. You must manually create an index on foreign keys.

There are actual advantages to having a foreign key be supported with a clustered index, but you get only one per table. What's the advantage? If you are selecting the parent plus all child records, you want the child records next to each other. This is easy to accomplish using a clustered index.

Having a null foreign key is usually a bad idea instead of NULL  referred to as "orphan record".

We can’t define foreign key constraint on temporary table or table variable.

We can delete the foreign key value from the child table even though that refers to the primary key of the parent table.

1. What is the select statement? What are some common clauses used with SELECT Query in SQL?

The SELECT clause specifies the table columns that are retrieved. The FROM clause specifies the tables accessed. The WHERE clause specifies which table rows are used. The WHERE clause is optional; if missing, all table rows are used.

Mainly Five Clauses of the SELECT statement

SELECT – the columns in the result set.

FROM – names the base table(s) from which results will be retrieved.

WHERE – specifies any conditions for the results set (filter)

ORDER BY – sets how the result set will be ordered.

LIMIT – sets the number of rows to be returned.

1. What is a join and what are different types of joins in SQL?

A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are:

INNER JOIN

LEFT JOIN

RIGHT JOIN

FULL JOIN

SELECT StudentCourse.COURSE\_ID, Student.NAME, Student.AGE FROM Student

INNER JOIN StudentCourse ON Student.ROLL\_NO=StudentCourse.ROLL\_NO;

**LEFT JOIN**: This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join. The rows for which there is no matching row on right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN.Syntax:

SELECT Student.NAME, StudentCourse.COURSE\_ID FROM Student

**LEFT JOIN** StudentCourse ON StudentCourse.ROLL\_NO = Student.ROLL\_NO;

**RIGHT JOIN:** RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join. The rows for which there is no matching row on left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.Syntax:

SELECT Student.NAME, StudentCourse.COURSE\_ID FROM Student

**RIGHT JOI**N StudentCourse ON StudentCourse.ROLL\_NO = Student.ROLL\_NO;

**FULL JOIN:** FULL JOIN creates the result-set by combining result of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both the tables. The rows for which there is no matching, the result-set will contain NULL values. Syntax:

SELECT Student.NAME, StudentCourse.COURSE\_ID FROM Student

**FULL JOIN** StudentCourse ON StudentCourse.ROLL\_NO = Student.ROLL\_NO;

1. What is the difference between TRUNCATE, DROP and DELETE Statement?

DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command. DELETE operations can be rolled back (undone), while DROP and TRUNCATE operations cannot be rolled back

1. DELETE :

Basically, it is a [Data Manipulation Language Command (DML)](https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/). It is used to delete one or more tuples of a table. With the help of the “DELETE” command, we can either delete all the rows in one go or can delete rows one by one. i.e., we can use it as per the requirement or the condition using the Where clause. It is comparatively slower than the TRUNCATE command. The TRUNCATE command does not remove the structure of the table.

**SYNTAX –**   
If we want to delete all the rows of the table:

DELETE from;

**SYNTAX –**   
If we want to delete the row of the table as per the condition then we use the WHERE clause,

DELETE from WHERE ;

**Note –** Here we can use the “ROLLBACK” command to restore the tuple because it does not auto-commit.

2. DROP :

It is a Data Definition Language Command (DDL). It is used to drop the whole table. With the help of the “DROP” command we can drop (delete) the whole structure in one go i.e. it removes the named elements of the schema. By using this command the existence of the whole table is finished or say lost.

**SYNTAX –**   
If we want to drop the table:

DROP table ;

**Note –** Here we can’t restore the table by using the “ROLLBACK” command because it auto commits.

3. TRUNCATE :

It is also a Data Definition Language Command (DDL). It is used to delete all the rows of a relation (table) in one go. With the help of the “TRUNCATE” command, we can’t delete the single row as here WHERE clause is not used. By using this command the existence of all the rows of the table is lost. It is comparatively faster than the delete command as it deletes all the rows fastly.

**SYNTAX –**   
If we want to use truncate :

TRUNCATE;

1. What is an Alias in SQL?

SQL aliases are used to give a table, or a column in a table, a temporary name. Aliases are often used to make column names more readable. An alias only exists for the duration of that query. An alias is created with the AS keyword.

Example: SELECT CustomerID AS ID, CustomerName AS Customer  
FROM Customers;

1. What do you mean by constraints and what are different constraints used in sql?

Constraints are the rules enforced on the data columns of a table. These are used to limit the type of data that can go into a table

The following constraints are commonly used in SQL:

[NOT NULL](https://www.w3schools.com/sql/sql_notnull.asp) - Ensures that a column cannot have a NULL value

[UNIQUE](https://www.w3schools.com/sql/sql_unique.asp) - Ensures that all values in a column are different

[PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

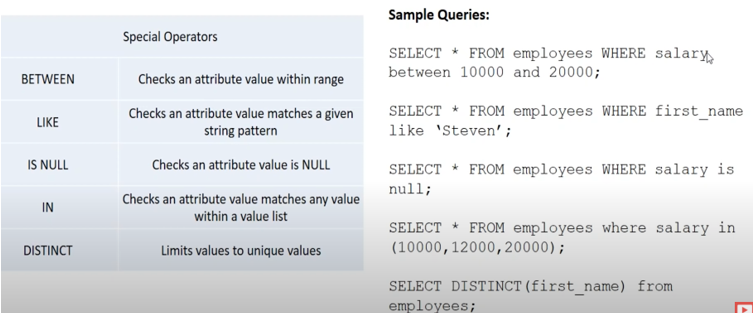
[FOREIGN KEY](https://www.w3schools.com/sql/sql_foreignkey.asp) - Prevents actions that would destroy links between tables

[CHECK](https://www.w3schools.com/sql/sql_check.asp) - Ensures that the values in a column satisfies a specific condition

[DEFAULT](https://www.w3schools.com/sql/sql_default.asp) - Sets a default value for a column if no value is specified

[CREATE INDEX](https://www.w3schools.com/sql/sql_create_index.asp) - Used to create and retrieve data from the database very quickly

1. What are different special operators in SQL.



1. What is the difference between WHERE and HAVING Clause?

The main difference between the WHERE and HAVING clauses comes when used together with the GROUP BY clause. In that case, WHERE is used to filter rows before grouping, and HAVING is used to exclude records after grouping.

SELECT S\_Name, Age FROM Student

WHERE Age >=18;

SELECT Age, COUNT(Roll\_No) AS No\_of\_Students

FROM Student GROUP BY Age

HAVING COUNT(Roll\_No) > 1 ;

1. What is the difference between IN and BETWEEN Operator?

Both of these operators are used to find out the multiple values from the table. Differences between these operators is that the BETWEEN operator is used to select a range of data between two values while The IN operator allows you to specify multiple values.

SELECT Fname, Lname FROM Employee WHERE Salary BETWEEN 30000 AND 45000;

SELECT Fname, Lname FROM Employee WHERE Salary IN (30000, 40000, 25000);

1. What is the default ordering for data using the ORDER BY Clause? How could it be changed?

In SQL ORDER BY clause, we need to define ascending or descending order in which result needs to be sorted. By default, SQL Server sorts out results using ORDER BY clause in ascending order. Specifying ASC in order by clause is optional.

SELECT agent\_code, agent\_name, working\_area, commission FROM agents ORDER BY agent\_code DESC;

1. What is DISTINCT Statement and how to use it?

The SELECT DISTINCT statement is used to return only distinct (different) values. Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

SELECT DISTINCT SALARY FROM CUSTOMERS ORDER BY SALARY;

1. What are aggregate functions and Which aggregate function we used in SQL?

An aggregate function allows you to perform a calculation on a set of values to return a single scalar value. We often use aggregate functions with the [GROUP BY](https://www.zentut.com/sql-tutorial/sql-group-by/)and [HAVING](https://www.zentut.com/sql-tutorial/sql-having/) clauses of the [SELECT](https://www.zentut.com/sql-tutorial/sql-select/)statement.The following are the most commonly used SQL aggregate functions:

[AVG](http://www.sqltutorial.org/sql-avg.aspx)– calculates the average of a set of values.

[COUNT](http://www.sqltutorial.org/sql-count.aspx)– counts rows in a specified table or view.

[MIN](http://www.sqltutorial.org/sql-min-max.aspx)– gets the minimum value in a set of values.

[MAX](https://www.zentut.com/sql-tutorial/sql-max/)– gets the maximum value in a set of values.

[SUM](http://www.sqltutorial.org/sql-sum.aspx)– calculates the sum of values.

What are subsets of SQL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DDL | DML | DCL | TCL | DQL |
| CREATE | INSERT | GRANT | COMMIT | SELECT |
| ALTER | UPDATE | REVOKE | ROLLBACL |  |
| DROP | DELETE |  | SAVEPOINT |  |
| RENAME |  |  |  |  |
| TRUNCATE |  |  |  |  |

(DDL, DML, DCL, TCL, DQL)

DDL (Data Definition Language) – It allows you to perform various operations on the database such as CREATE, ALTER and DROP objects.

DML ( Data Manipulation Language) – It allows you to access and manipulate data. It helps you to insert, update, delete and retrieve data from the database.

DCL ( Data Control Language) – It allows you to control access to the database. Example – Grant, Revoke access permissions.

TCL commands deal with the transaction within the database. They are also used to make the changes that are done by DML statements. Examples of TCL commands

COMMIT– commits a Transaction.

ROLLBACK– rollbacks a transaction in case any error occurs.

SET TRANSACTION–specify characteristics for the transaction.

SAVEPOINT– sets a savepoint within a transaction.

1. What are different types of relationships in Sql?

There are five types of relations in the databases: one-to-one, one-to-many, many-to-one, many-to-many, and self-referencing relationships. So, what's the difference between these database relationship types? In the article, we'll examine each type separately and provide a working example of their usage

[One-to-many relationship](https://blog.devart.com/types-of-relationships-in-sql-server-database.html#one-to-many)  
[One-to-one relationship](https://blog.devart.com/types-of-relationships-in-sql-server-database.html#one-to-one-relationship)  
[Many-to-many relationship](https://blog.devart.com/types-of-relationships-in-sql-server-database.html#many-to-many-relationship)  
[Many-to-one relationship](https://blog.devart.com/types-of-relationships-in-sql-server-database.html#many-to-one-relationship)  
[Self-referencing relationships](https://blog.devart.com/types-of-relationships-in-sql-server-database.html#self-referencing-relationships)

1. What is RDBMS? How it is different from DBMS?

Database Management System (DBMS) is a software that is used to define, create and maintain a database and provides controlled access to the data. Relational Database Management System (RDBMS) is an advanced version of a DBMS.

|  |  |
| --- | --- |
| DBMS | RDBMS |
| DBMS stores data as file. | RDBMS stores data in tabular form. |
| Data elements need to access individually. | Multiple data elements can be accessed at the same time. |
| No relationship between data. | Data is stored in the form of tables which are related to each other. |
| Normalization is not present. | Normalization is present. |
| DBMS does not support distributed database. | RDBMS supports distributed database. |
| It stores data in either a navigational or hierarchical form. | It uses a tabular structure where the headers are the column names, and the rows contain corresponding values. |
| It deals with small quantity of data. | It deals with large amount of data. |
| Data redundancy is common in this model. | Keys and indexes do not allow Data redundancy. |
| It is used for small organization and deal with small data. | It is used to handle large amount of data. |
| It supports single user. | It supports multiple users. |
| Data fetching is slower for the large amount of data. | Data fetching is fast because of relational approach. |
| The data in a DBMS is subject to low security levels with regards to data manipulation. | There exists multiple levels of data security in a RDBMS. |
| Low software and hardware necessities. | Higher software and hardware necessities. |
| Examples: XML, Window Registry, etc. | Examples: MySQL, PostgreSQL, SQL Server, Oracle, Microsoft Access etc. |