

# SQL Interview Questions and Answers for Freshers

## 1. What is SQL?

**SQL** stands for Structured Query Language. It is a language used to interact with the database, i.e to create a database, to create a table in the database, to retrieve data or update a table in the database, etc. SQL is an ANSI(American National Standards Institute) standard. Using SQL, we can do many things. For example – we can execute queries, we can insert records into a table, can update records, can create a database, can create a table, can delete a table, etc.

## 2. What is a database?

A **Database** is defined as a structured form of data storage in a computer or a collection of data in an organized manner and can be accessed in various ways. It is also the collection of schemas, tables, queries, views, etc. Databases help us with easily storing, accessing, and manipulating data held on a computer. The Database Management System allows a user to interact with the database.

## 3. Does SQL support programming language features?

It is true that SQL is a language, but it does not support programming as it is not a programming language, it is a command language. We do not have conditional statements in SQL like for loops or if..else, we only have commands which we can use to query, update, delete, etc. data in the database. SQL allows us to manipulate data in a database.

## 4. What is the difference between CHAR and VARCHAR2 datatype in SQL?

Both of these data types are used for characters, but varchar2 is used for character strings of variable length, whereas char is used for character strings of fixed length. **For example**, if we specify the type as char(5) then we will not be allowed to store a string of any other length in this variable, but if we specify the type of this variable as varchar2(5) then we will be allowed to store strings of variable length. We can store a string of length 3 or 4 or 2 in this variable.

## 5. What do you mean by data definition language?

**Data definition language** or DDL allows to execution of queries like CREATE, DROP, and ALTER. That is those queries that define the data.

## 6. What do you mean by data manipulation language?

Data manipulation Language or DML is used to access or manipulate data in the database. It allows us to perform the below-listed functions:

- Insert data or rows in a database
- Delete data from the database
- Retrieve or fetch data
- Update data in a database.

## 7. What is the view in SQL?

**Views in SQL** are a kind of virtual table. A view also has rows and columns as they are on a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain conditions.

The CREATE VIEW statement of SQL is used for creating views.

**Basic Syntax:**

```
CREATE VIEW view_name AS
SELECT column1, column2.....
FROM table_name
WHERE condition;
```

**view\_name:** Name for the View

**table\_name:** Name of the table

**condition:** Condition to select rows

### 8. What do you mean by foreign key?

A [Foreign key](#) is a field that can uniquely identify each row in another table. And this constraint is used to specify a field as a Foreign key. That is this field points to the primary key of another table. This usually creates a kind of link between the two tables.

Consider the two tables as shown below:

#### Orders

O_ID	ORDER_NO	C_ID
1	2253	3
2	3325	3
3	4521	2
4	8532	1

#### Customers

C_ID	NAME	ADDRESS
1	RAMESH	DELHI
2	SURESH	NOIDA
3	DHARMESH	GURGAON

As we can see clearly, that the field C\_ID in the Orders table is the primary key in the Customers' table, i.e. it uniquely identifies each row in the Customers table. Therefore, it is a Foreign Key in the Orders table.

#### Syntax:

```
CREATE TABLE Orders
(
O_ID int NOT NULL,
```

```
ORDER_NO int NOT NULL,  
C_ID int,  
PRIMARY KEY (O_ID),  
FOREIGN KEY (C_ID) REFERENCES Customers(C_ID)  
)
```

### 9. What are table and Field?

**Table:** A table has a combination of rows and columns. Rows are called records and columns are called fields. In MS SQL Server, the tables are being designated within the database and schema names.

**Field:** In DBMS, a database field can be defined as – a single piece of information from a record.

### 10. What is the primary key?

A [Primary Key](#) is one of the candidate keys. One of the candidate keys is selected as the most important and becomes the primary key. There cannot be more than one primary key in a table.

### 11. What is a Default constraint?

The [DEFAULT](#) constraint is used to fill a column with default and fixed values. The value will be added to all new records when no other value is provided.

### 12. What is normalization?

It is a process of analyzing the given relation schemas based on their functional dependencies and primary keys to achieve the following desirable properties:

1. Minimizing Redundancy
2. Minimizing the Insertion, Deletion, And Update Anomalies

Relation schemas that do not meet the properties are decomposed into smaller relation schemas that could meet desirable properties.

### 13. What is Denormalization?

[Denormalization](#) is a database optimization technique in which we add redundant data to one or more tables. This can help us avoid costly joins in a relational database. Note that denormalization does not mean not doing normalization. It is an optimization technique that is applied after normalization.

In a traditional normalized database, we store data in separate logical tables and attempt to minimize redundant data. We may strive to have only one copy of each piece of data in the database.

### 14. What is a query?

An **SQL** query is used to retrieve the required data from the database. However, there may be multiple SQL queries that yield the same results but with different levels of efficiency. An inefficient query can drain the database resources, reduce the database speed or result in a loss of service for other users. So it is very important to optimize the query to obtain the best database performance.

### 15. What is a subquery?

In SQL, a [Subquery](#) can be simply defined as a query within another query. In other words, we can say that a Subquery is a query that is embedded in the WHERE clause of another SQL query.

### 16. What are the different operators available in SQL?

There are three operators available in SQL namely:

1. Arithmetic Operators
2. Logical Operators
3. Comparison Operators

### **17. What is a Constraint?**

Constraints are the rules that we can apply to the type of data in a table. That is, we can specify the limit on the type of data that can be stored in a particular column in a table using constraints. For more details please refer to [SQL Constraints](#) article.

### **18. What is Data Integrity?**

Data integrity is defined as the data contained in the database being both correct and consistent. For this purpose, the data stored in the database must satisfy certain types of procedures (rules). The data in a database must be correct and consistent. So, data stored in the database must satisfy certain types of procedures (rules). DBMS provides different ways to implement such types of constraints (rules). This improves data integrity in a database. For more details please refer [difference between data security and data integrity](#) article.

### **19. What is Auto Increment?**

Sometimes, while creating a table, we do not have a unique identifier within the table, hence we face difficulty in choosing Primary Key. So as to resolve such an issue, we've to manually provide unique keys to every record, but this is often also a tedious task. So we can use the Auto-Increment feature that automatically generates a numerical Primary key value for every new record inserted. The Auto Increment feature is supported by all the Databases. For more details please refer [SQL Auto Increment](#) article.

### **20. What is MySQL collation?**

A MySQL collation is a well-defined set of rules which are used to compare characters of a particular character set by using their corresponding encoding. Each character set in MySQL might have more than one collation, and has, at least, one default collation. Two character sets cannot have the same collation. For more details please refer [What are collation and character set in MySQL?](#) article.

### **21. What are user-defined functions?**

We can use User-defined functions in PL/SQL or Java to provide functionality that is not available in SQL or SQL built-in functions. SQL functions and User-defined functions can appear anywhere, that is, wherever an expression occurs.

For example, it can be used in:

- Select a list of SELECT statements.
- Condition of the WHERE clause.
- CONNECT BY, ORDER BY, START WITH, and GROUP BY
- The VALUES clause of the INSERT statement.
- The SET clause of the UPDATE statement.

### **22. What are all types of user-defined functions?**

User-Defined Functions allow people to define their own T-SQL functions that can accept 0 or more parameters and return a single scalar data value or a table data type.

Different Kinds of User-Defined Functions created are:

**1. Scalar User-Defined Function** A Scalar user-defined function returns one of the scalar data types. Text, image, and timestamp data types are not supported. These are the type of user-defined functions that most developers are used to in other programming languages. You pass in 0 to many parameters and you get a return value.

**2. Inline Table-Value User-Defined Function** An Inline Table-Value user-defined function returns a table data type and is an exceptional alternative to a view as the user-defined function can pass parameters into a T-SQL select command and, in essence, provide us with a parameterized, non-updateable view of the underlying tables.

**3. Multi-statement Table-Value User-Defined Function** A Multi-Statement Table-Value user-defined function returns a table and is also an exceptional alternative to a view, as the function can support multiple T-SQL statements to build the final result where the view is limited to a single SELECT statement. Also, the ability to pass parameters into a TSQL select command or a group of them gives us the capability to, in essence, create a parameterized, non-updateable view of the data in the underlying tables. Within the create function command you must define the table structure that is being returned. After creating this type of user-defined function, it can be used in the FROM clause of a T-SQL command, unlike the behavior found when using a stored procedure which can also return record sets.

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

**Stored Procedures** are created to perform one or more DML operations on databases. It is nothing but a group of SQL statements that accepts some input in the form of parameters and performs some task and may or may not return a value. For more details please refer to our [Stored procedures in the SQL](#) article.

### **24. What are aggregate and scalar functions?**

For doing operations on data SQL has many built-in functions, they are categorized into two categories and further sub-categorized into seven different functions under each category. The categories are:

- **Aggregate functions:** These functions are used to do operations from the values of the column and a single value is returned.
- **Scalar functions:** These functions are based on user input, these too return a single value.

For more details, please read the [SQL | Functions \(Aggregate and Scalar Functions\)](#) article.

### **25. What is an ALIAS command?**

Aliases are the temporary names given to a table or column for the purpose of a particular SQL query. It is used when the name of a column or table is used other than its original name, but the modified name is only temporary.

- Aliases are created to make table or column names more readable.
- The renaming is just a temporary change and the table name does not change in the original database.
- Aliases are useful when table or column names are big or not very readable.
- These are preferred when there is more than one table involved in a query.

For more details, please read the [SQL | Aliases](#) article.

### **26. What are Union, minus, and Intersect commands?**

Set Operations in SQL eliminate duplicate tuples and can be applied only to the relations which are union compatible. Set Operations available in SQL are :

- Set Union
- Set Intersection
- Set Difference

**UNION Operation:** This operation includes all the tuples which are present in either of the relations. For example: To find all the customers who have a loan or an account or both in a bank.

```
SELECT CustomerName FROM Depositor  
UNION
```

```
SELECT CustomerName FROM Borrower ;
```

The union operation automatically eliminates duplicates. If all the duplicates are supposed to be retained, UNION ALL is used in place of UNION.

**INTERSECT Operation:** This operation includes the tuples which are present in both of the relations. For example: To find the customers who have a loan as well as an account in the bank:

```
SELECT CustomerName FROM Depositor  
INTERSECT
```

```
SELECT CustomerName FROM Borrower ;
```

The Intersect operation automatically eliminates duplicates. If all the duplicates are supposed to be retained, INTERSECT ALL is used in place of INTERSECT.

**EXCEPT for Operation:** This operation includes tuples that are present in one relationship but should not be present in another relationship. For example: To find customers who have an account but no loan at the bank:

```
SELECT CustomerName FROM Depositor  
EXCEPT
```

```
SELECT CustomerName FROM Borrower ;
```

The Except operation automatically eliminates the duplicates. If all the duplicates are supposed to be retained, EXCEPT ALL is used in place of EXCEPT.

## 27. What is a T-SQL?

T-SQL is an abbreviation for Transact Structure Query Language. It is a product by Microsoft and is an extension of SQL Language which is used to interact with relational databases. It is considered to perform best with Microsoft SQL servers. T-SQL statements are used to perform the transactions to the databases. T-SQL has huge importance since all the communications with an instance of an SQL server are done by sending Transact-SQL statements to the server. Users can also define functions using T-SQL.

Types of T-SQL functions are :

- **Aggregate** functions.
- **Ranking** functions. There are different types of ranking functions.
- **Rowset** function.
- **Scalar** functions.

## 28. What is ETL in SQL?

ETL is a process in Data Warehousing and it stands for **Extract, Transform, and Load**. It is a process in which an ETL tool extracts the data from various data source systems,

transforms it in the staging area, and then finally, loads it into the Data Warehouse system. These are three database functions that are incorporated into one tool to pull data out from one database and put data into another database.

### 29. How to copy tables in SQL?

Sometimes, in SQL, we need to create an exact copy of an already defined (or created) table. [MySQL](#) enables you to perform this operation. Because we may need such duplicate tables for testing the data without having any impact on the original table and the data stored in it.

```
CREATE TABLE Contact List(Clone_1) LIKE Original_table;
```

For more details, Please read [Cloning Table in the MySQL](#) article.

### 30. What is SQL injection?

SQL injection is a technique used to exploit user data through web page inputs by injecting SQL commands as statements. Basically, these statements can be used to manipulate the application's web server by malicious users.

- SQL injection is a code injection technique that might destroy your database.
- SQL injection is one of the most common web hacking techniques.
- SQL injection is the placement of malicious code in SQL statements, via web page input.

For more details, please read the [SQL | Injection](#) article.

### 31. Can we disable a trigger? If yes, how?

Yes, we can disable a trigger in PL/SQL. If consider temporarily disabling a trigger and one of the following conditions is true:

- An object that the trigger references is not available.
- We must perform a large data load and want it to proceed quickly without firing triggers.
- We are loading data into the table to which the trigger applies.
- We disable a trigger using the ALTER TRIGGER statement with the DISABLE option.
- We can disable all triggers associated with a table at the same time using the ALTER TABLE statement with the DISABLE ALL TRIGGERS option.

## Intermediate SQL Interview Questions and Answers

### 32. What are the differences between SQL and PL/SQL?

Some common differences between SQL and PL/SQL are as shown below:

SQL	PL/SQL
SQL is a query execution or commanding language	PL/SQL is a complete programming language
SQL is a data-oriented language.	PL/SQL is a procedural language
SQL is very declarative in nature.	PL/SQL has a procedural nature.

SQL	PL/SQL
It is used for manipulating data.	It is used for creating applications.
We can execute one statement at a time in SQL	We can execute blocks of statements in PL/SQL
SQL tells databases, what to do?	PL/SQL tells databases how to do.
We can embed SQL in PL/SQL	We can not embed PL/SQL in SQL

### 33. What is the difference between BETWEEN and IN operators in SQL?

**BETWEEN:** The **BETWEEN** operator is used to fetch rows based on a range of values.

For example,

```
SELECT * FROM Students
WHERE ROLL_NO BETWEEN 20 AND 30;
```

This query will select all those rows from the table. Students where the value of the field ROLL\_NO lies between 20 and 30.

**IN:** The **IN** operator is used to check for values contained in specific sets.

For example,

```
SELECT * FROM Students
WHERE ROLL_NO IN (20,21,23);
```

This query will select all those rows from the table Students where the value of the field ROLL\_NO is either 20 or 21 or 23.

### 34. Write an SQL query to find the names of employees starting with 'A'.

The LIKE operator of SQL is used for this purpose. It is used to fetch filtered data by searching for a particular pattern in the where clause.

The Syntax for using LIKE is,

```
SELECT column1,column2 FROM table_name WHERE column_name LIKE pattern;
```

**LIKE:** operator name

**pattern:** exact value extracted from the pattern to get related data in result set.

The required query is:

```
SELECT * FROM Employees WHERE EmpName like 'A%' ;
```

You may refer to this article [WHERE clause](#) for more details on the LIKE operator.

### 35. What is the difference between primary key and unique constraints?

The primary key cannot have NULL values, the unique constraints can have NULL values. There is only one primary key in a table, but there can be multiple unique constraints. The primary key creates the clustered index automatically but the unique key does not.

### 36. What is a join in SQL? What are the types of joins?

An 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:** The INNER JOIN keyword selects all rows from both tables as long as the condition is satisfied. This keyword will create the result set by combining all rows from both the tables where the condition satisfies i.e. the value of the common field will be the same.
- **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 the join. For the rows for which there is no matching row on the right side, the result set will be null. LEFT JOIN is also known as LEFT OUTER JOIN
- **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 the join. For the rows for which there is no matching row on the left side, the result set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.
- **FULL JOIN:** FULL JOIN creates the result set by combining the results of both LEFT JOIN and RIGHT JOIN. The result set will contain all the rows from both tables. For the rows for which there is no matching, the result set will contain NULL values.

### 37. What is an index?

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data. Data can be stored only in one order on a disk. To support faster access according to different values, a faster search like a binary search for different values is desired. For this purpose, indexes are created on tables. These indexes need extra space on the disk, but they allow faster search according to different frequently searched values.

### 38. What is the On Delete cascade constraint?

An 'ON DELETE CASCADE' constraint is used in MySQL to delete the rows from the child table automatically when the rows from the parent table are deleted. For more details, please read [MySQL – On Delete Cascade constraint](#) article.

### 39. Explain WITH clause in SQL?

The WITH clause provides a way relationship of defining a temporary relationship whose definition is available only to the query in which the with clause occurs. SQL applies predicates in the WITH clause after groups have been formed, so aggregate functions may be used.

### 40. What are all the different attributes of indexes?

The indexing has various attributes:

- **Access Types:** This refers to the type of access such as value-based search, range access, etc.
- **Access Time:** It refers to the time needed to find a particular data element or set of elements.
- **Insertion Time:** It refers to the time taken to find the appropriate space and insert new data.
- **Deletion Time:** Time is taken to find an item and delete it as well as update the index structure.
- **Space Overhead:** It refers to the additional space required by the index.

### 41. What is a Cursor?

The cursor is a Temporary Memory or Temporary Work Station. It is Allocated by Database Server at the Time of Performing DML operations on the Table by the User. Cursors are used to store Database Tables.

#### 42. Write down various types of relationships in SQL?

There are various relationships, namely:

- One-to-One Relationship.
- One to Many Relationships.
- Many to One Relationship.
- Self-Referencing Relationship.

#### 43. What is a trigger?

**The trigger** is a statement that a system executes automatically when there is any modification to the database. In a trigger, we first specify when the trigger is to be executed and then the action to be performed when the trigger executes. Triggers are used to specify certain integrity constraints and referential constraints that cannot be specified using the constraint mechanism of SQL.

#### 44. What is the difference between SQL DELETE and SQL TRUNCATE commands?

SQL DELETE	SQL TRUNCATE
The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row.	TRUNCATE TABLE removes the data by deallocating the data pages used to store the table data and records only the page deallocations in the transaction log.
DELETE command is slower than the identityTRUNCATE command.	While the TRUNCATE command is faster than the DELETE command.
To use Delete you need DELETE permission on the table.	To use Truncate on a table we need at least ALTER permission on the table.
The identity of the column retains the identity after using DELETE Statement on the table.	The identity of the column is reset to its seed value if the table contains an identity column.
The delete can be used with indexed views.	Truncate cannot be used with indexed views.

#### 45. What is the difference between Cluster and Non-Cluster Index?

CLUSTERED INDEX	NON-CLUSTERED INDEX
The clustered index is faster.	The non-clustered index is slower.

<b>CLUSTERED INDEX</b>	<b>NON-CLUSTERED INDEX</b>
The clustered index requires less memory for operations.	The non-Clustered index requires more memory for operations.
In a clustered index, the index is the main data.	In the Non-Clustered index, the index is a copy of data.
A table can have only one clustered index.	A table can have multiple non-clustered indexes.
The clustered index has an inherent ability to store data on the disk.	The non-Clustered index does not have the inherent ability to store data on the disk.
Clustered indexes store pointers to block not data.	The non-Clustered index store both value and a pointer to the the the actual row that holds data.
In Clustered index leaf nodes are actual data itself.	In a Non-Clustered index, leaf nodes are not the actual data itself rather they only contain included columns.
In the Clustered index, the Clustered key defines the order of data within the table.	In the Non-Clustered index, the index key defines the order of data within the index.
A Clustered index is a type of index in which table records are physically reordered to match the index.	A Non-Clustered index is a special type of index in which the logical order of index does not match the physical stored order of the rows on the disk.

For more details please refer [Difference between Clustered index and the No-Clustered index](#) article.

#### **46. What is a Live Lock?**

**Livelock** occurs when two or more processes continually repeat the same interaction in response to changes in the other processes without doing any useful work. These processes are not in the waiting state, and they are running concurrently. This is different from a deadlock because in a deadlock all processes are in the waiting state.

#### **47. What is Case WHEN in SQL?**

Control statements form an important part of most languages since they control the execution of other sets of statements. These are found in SQL too and should be exploited

for uses such as query filtering and query optimization through careful selection of tuples that match our requirements. In this post, we explore the Case-Switch statement in SQL. The CASE statement is SQL's way of handling if/then logic.

**Syntax 1:**

*CASE case\_value WHEN when\_value THEN statement\_list [WHEN when\_value THEN statement\_list] ... [ELSE statement\_list]END CASE*

**Syntax 2:**

*CASE WHEN search\_condition THEN statement\_list [WHEN search\_condition THEN statement\_list] ... [ELSE statement\_list]END CASE*