



---

# SQL INTERVIEW QUESTIONS

---

Pdf By Aniket Singh



2024

{PDF BABA}

# INDEX

## ➤ BASIC

- WHAT IS SQL ?
- WHAT IS DATABASE ?
- WHAT IS A TABLE IN SQL ?
- WHAT IS A PRIMARY KEY ?
- What IS A FOREIGN KEY ?
- WHAT IS A SQL QUERY ?
- WHAT IS IS A “ SELECT” STATEMENT ?
- WHAT IS A “WHERE” CLAUSE ?
- WHAT IS AN “INSERT” STATEMENT ?
- WHAT IS AN “UPDATE” STATEMENT ?
- WHAT IS A DELETE STATEMENT ?
- WHAT IS A “JOIN” IN SQL ?
- WHAT IS AN INNER JOIN ?
- WHAT IS LEFT JOIN ?
- WHAT IS A RIGHT JOIN ?
- WHAT IS FULL JOIN ?
- WHAT IS A SELF-JOIN ?
- WHAT IS A UNION ?WHAT IS A SUBQUERY ?
- WHAAT IS NORMALIZATION ?

## ➤ INTERMEDIATE

- WHAT IS A DENORMALIZATION ?
- WHAT IS AN INDEX IN SQL ?
- WHAT IS A VIEW IN SQL ?
- WHAT IS A STORE PROCEDURE ?
- WHAT IS A TRIGGER IN SQL ?

- WHAT IS A TRANSACTION IN SQL ?
- WHAT ARE ACID PROPERTIES ?
- WHAT IS A CONSTRAINT IN SQL ?
- WHAT IS A UNIQUE CONSTRAINT ?
- WHAT IS A CHECK CONSTRAINT ?
- WHAT IS THE DEFAULT CONSTRAINT ?
- WHAT IS A DATA TYPE IN SQL ?
- WHAT IS THE DIFFERENCE BETWEEN CHAR AND VARCHAR ?
- WHAT IS A SCALAR FUNCTION IN SQL ?
- WHAT IS AN AGGREGATE FUNCTION IN SQL ?
- WHAT IS GROUP BY ?
- WHAT IS HAVING CLAUSE ?
- WHAT IS A CASE STATEMENT IN SQL ?
- WHAT IS A CURSOR IN SQL ?

### ➤ ADVANCE

- WHAT IS THE DIFFERENCE BETWEEN DELETE AND TRUNCATE ?
- WHAT IS A SCHEMA IN SQL ?
- WHAT IS A SYNONYM IN SQL ?
- WHAT IS A CORRELATED SUBQUERY ?
- WHAT IS SQL INJECTION ?
- WHAT IS A DIFFERENCE BETWEEN UNION AND UNION ALL ?
- WHAT IS A COMPOSITE KEY ?
- WHAT IS AN ALIAS IN SQL ?
- WHAT IS SEQUENCE IN SQL ?
- . WHAT IS TEMPORARY TABLE ?

## ❖BASIC

### Q 1- WHAT IS SQL ?

- SQL, or Structured Query Language, is a standard programming language used for managing and manipulating relational databases. It allows users to perform operations such as querying data, updating records, inserting new data, and deleting existing data. SQL is essential for database administration and data analysis

### Q 2- WHAT IS A DATABASE ?

- A database is an organized collection of structured data stored electronically, typically managed by a Database Management System (DBMS). It allows for efficient data retrieval, management, and manipulation, enabling users to store and analyze information systematically.

### Q 3- WHAT IS A TABLE IN SQL ?

- A table in SQL is a structured format for organizing data within a database, consisting of rows and columns. Each row represents a unique record, while each column corresponds to a specific attribute or field of that record. Tables are fundamental components of relational databases, enabling data storage and retrieval.

### Q 4- WHAT IS A PRIMARY KEY ?

- A primary key is a unique identifier for each record in a database table. It ensures that no two rows have the same value in this column, maintaining data integrity. A primary key can consist of a single column or a combination of multiple columns.

## Q 5- WHAT IS A FOREIGN KEY ?

- A foreign key is a field in a database table that establishes a link between two tables. It refers to the primary key of another table, ensuring referential integrity by allowing data in one table to relate to data in another. This relationship enables the enforcement of constraints and the maintenance of consistent data across tables.

## Q 6- WHAT IS A SQL QUERY ?

- A SQL query is a request made to a database to retrieve, manipulate, or interact with data. It typically uses SQL syntax to perform operations such as selecting, inserting, updating, or deleting records. Queries enable users to extract specific information based on defined criteria.

## Q 7- WHAT IS A SELECT STATEMENT ?

- A SELECT statement is a SQL command used to retrieve data from one or more tables in a database. It allows users to specify which columns to display and can include filters, sorting, and aggregation to refine the results. For example, `SELECT * FROM table_name;` retrieves all records from the specified table

## Q 8- WHAT IS WHERE CLAUSE ?

- The WHERE clause is a SQL statement component used to filter records based on specific conditions. It restricts the results returned by a query to only those rows that meet the specified criteria. For example, `SELECT * FROM table_name WHERE condition;` retrieves only the records that satisfy the condition.

## Q 9- WHAT IS AN INSERT STATEMENT ?

- An INSERT statement is a SQL command used to add new records to a database table. It specifies the table and the values to be inserted into each column. For example, `INSERT INTO table_name (column1, column2) VALUES (value1, value2);` adds a new row with the specified values.

## Q 10- WHAT IS AN UPDATE STATEMENT ?

- An UPDATE statement is a SQL command used to modify existing records in a database table. It specifies the table, the columns to be updated, the new values, and a condition to identify which records to change. For example, `UPDATE table_name SET column1 = value1 WHERE condition;` updates the specified records based on the condition.

## Q 12- WHAT IS AN DELETE STATEMENT ?

- A DELETE statement is a SQL command used to remove existing records from a database table. It specifies the table and can include a condition to determine which records to delete. For example, `DELETE FROM table_name WHERE condition;` deletes the records that meet the specified condition.

## Q 13- WHAT IS A JOIN IN SQL ?

- A JOIN in SQL is a operation used to combine rows from two or more tables based on a related column between them. It allows users to retrieve data from multiple tables in a single query. Common types of joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN, each serving different purposes in how records are matched.

## Q 14- WHAT IS AN INNER JOIN ?

- An INNER JOIN is a type of SQL join that returns only the rows with matching values in both tables being joined. It excludes records that do not have a corresponding match in the other table. For example, `SELECT * FROM table1 INNER JOIN table2 ON table1.id = table2.id;` retrieves only the rows where the IDs match in both tables.

## Q 15- WHAT IS AN LEFT JOIN ?

- A LEFT JOIN is a type of SQL join that returns all records from the left table and the matched records from the right table. If there is no match, NULL values are returned for columns from the right table. For example, `SELECT * FROM table1 LEFT JOIN table2 ON table1.id = table2.id;` retrieves all rows from the left table, including those without corresponding matches in the right table.

## Q 16- WHAT IS AN RIGHT ?

- A RIGHT JOIN is a type of SQL join that returns all records from the right table and the matched records from the left table. If there is no match, NULL values are returned for columns from the left table. For example, `SELECT * FROM table1 RIGHT JOIN table2 ON table1.id = table2.id;` retrieves all rows from the right table, including those without corresponding matches in the left table.

## Q 17- WHAT IS A FULL JOINT ?

- A FULL JOIN, or FULL OUTER JOIN, is a type of SQL join that returns all records from both tables, with matching rows where available. If there is no match, NULL values are returned for columns of the table that lacks the corresponding record. For example, `SELECT * FROM table1 FULL JOIN table2 ON table1.id = table2.id;` retrieves all rows from both tables, including those without matches.

## Q 16- WHAT IS A SELF-JOIN ?

- A self-join is a SQL join that is used to combine and compare rows within the same table. It involves treating the table as two separate instances to compare data. This is typically done using table aliases. For example, `SELECT a.*, b.* FROM table_name a JOIN table_name b ON a.id = b.parent_id;` retrieves related records from the same table based on a specified relationship.

## Q 17- WHAT IS A UNION ?

- A UNION is a SQL operation that combines the result sets of two or more SELECT queries into a single result set, removing duplicate rows. The SELECT statements must have the same number of columns and compatible data types. For example, `SELECT column1 FROM table1 UNION SELECT column1 FROM table2;` retrieves unique values from both tables.

## Q 18- WHAT IS A SUBQUERY ?

- A subquery is a SQL query nested inside another query, used to retrieve data that will be utilized in the main query. It can appear in SELECT, INSERT, UPDATE, or DELETE statements. For example, `SELECT * FROM table1 WHERE id IN (SELECT id FROM table2);` retrieves records from table1 based on results from the subquery in table2.

## Q 19- WHAT IS NORIZATION ?

- Normalization is the process of organizing a database to reduce redundancy and improve data integrity. It involves dividing large tables into smaller, related ones and defining relationships between them. The goal is to eliminate duplicate data and ensure that data dependencies make sense, typically following a series of normalization forms (1NF, 2NF, 3NF, etc.).



## ❖ INTERMEDIATE

### Q 1- WHAT IS DENORMALIZATIONIN ?

- Denormalization is the process of intentionally introducing redundancy into a database by combining tables or adding redundant data. This is done to improve read performance and simplify complex queries, often at the cost of increased storage and potential data inconsistency. It's typically used in scenarios where fast data retrieval is more critical than maintaining strict normalization.

### Q 2- WHAT IS AN INDEX IN SQL ?

- An index in SQL is a database object that improves the speed of data retrieval operations on a table. It creates a data structure that allows the database to find rows more quickly, similar to an index in a book. While indexes can significantly enhance query performance, they can also slow down data modification operations (INSERT, UPDATE, DELETE) due to the overhead of maintaining the index.

### Q 3- WHAT IS A VIEW IN SQL

- A view in SQL is a virtual table that is based on the result of a SELECT query. It does not store data itself but provides a way to simplify complex queries, encapsulate data, and present it in a specific format. Views can be used for security, allowing users to access a subset of data without exposing the underlying tables. For example, `CREATE VIEW view_name AS SELECT * FROM table_name WHERE condition;` creates a view with the specified data.

## Q 4- WHAT IS A STORED PROCEDURE ?

- A stored procedure is a precompiled collection of SQL statements stored in the database that can be executed as a single unit. It allows for the encapsulation of complex business logic, improves performance, and enhances security by controlling access to data. Stored procedures can accept parameters and return results, making them useful for repetitive tasks or batch processing. For example, `CREATE PROCEDURE procedure_name AS BEGIN SQL_statements END;` defines a stored procedure.

## Q 5- WHAT IS A TRIGGER IN SQL ?

- A trigger in SQL is a set of predefined actions that automatically execute in response to specific events on a table, such as INSERT, UPDATE, or DELETE operations. Triggers are used to enforce business rules, maintain data integrity, and automate processes. For example, a trigger can log changes to a table or update related records whenever a row is modified. They are defined using the syntax `CREATE TRIGGER trigger_name ON table_name FOR INSERT AS BEGIN SQL_statements END;`.

## Q 6- WHAT IS A TRANSACTION IN SQL ?

- A transaction in SQL is a sequence of one or more SQL operations executed as a single unit of work. Transactions ensure data integrity and consistency by following the ACID properties: Atomicity (all or nothing), Consistency, Isolation, and Durability. This means that either all operations within the transaction are completed successfully, or none are applied. Transactions are commonly managed using commands like BEGIN, COMMIT, and ROLLBACK.

## Q 7- WHAT ARE ACID PROPERTIES ?

➤ ACID properties are a set of principles that ensure reliable processing of database transactions:

1. **Atomicity:** Ensures that a transaction is all-or-nothing; if any part fails, the entire transaction is rolled back.
2. **Consistency:** Guarantees that a transaction takes the database from one valid state to another, maintaining all predefined rules and constraints.
3. **Isolation:** Ensures that transactions are executed independently, so the intermediate state of a transaction is not visible to others until it is completed.
4. **Durability:** Ensures that once a transaction is committed, its changes are permanent, even in the event of a system failure.

## Q 8- WHAT IS CONSTRAINT IN SQL ?

➤ A constraint in SQL is a rule applied to a table column to enforce data integrity and ensure valid data entry. Common types of constraints include:

1. **PRIMARY KEY:** Uniquely identifies each row in a table.
2. **FOREIGN KEY:** Ensures referential integrity between tables by linking a column to a primary key in another table.
3. **UNIQUE:** Ensures all values in a column are distinct.
4. **NOT NULL:** Ensures a column cannot have NULL values.
5. **CHECK:** Validates that values in a column meet specific conditions.

Constraints help maintain the accuracy and reliability of the data in the database.

## Q 9- WHAT IS A UNIQUES CONSTRAINT ?

- A UNIQUE constraint in SQL ensures that all values in a specified column or combination of columns are distinct across the table. This means that no two rows can have the same value for that column(s), allowing for data integrity while still permitting NULL values (if applicable). It is used to prevent duplicate entries. For example, `ALTER TABLE table_name ADD CONSTRAINT constraint_name UNIQUE (column_name);` adds a UNIQUE constraint to the specified column.

## Q 10- WHAT IS A CHECK CONSTRAINT ?

- A CHECK constraint in SQL is a rule applied to a column to limit the values that can be entered, ensuring that they meet specific criteria. It enforces data integrity by allowing only values that satisfy a defined condition. For example, `ALTER TABLE table_name ADD CONSTRAINT constraint_name CHECK (column_name > 0);` ensures that only positive values can be entered in that column.

## Q 10- WHAT IS THE DEFAULT CONSTRAINT ?

- A DEFAULT constraint in SQL provides a default value for a column when no value is specified during an insert operation. This ensures that a column has a predetermined value, improving data integrity and simplifying data entry. For example, `ALTER TABLE table_name ADD CONSTRAINT constraint_name DEFAULT value FOR column_name;` sets a default value for the specified column.

## Q 11- WHAT IS THE DATA TYPE IN SQL ?

- A data type in SQL defines the kind of data that can be stored in a column, determining the operations that can be performed on it. Common data types include:

1. **INTEGER**: For whole numbers.
2. **FLOAT/DOUBLE**: For floating-point numbers.
3. **VARCHAR**: For variable-length strings.
4. **CHAR**: For fixed-length strings.
5. **DATE**: For date values.
6. **BOOLEAN**: For true/false values.

Choosing the appropriate data type is essential for optimizing storage and ensuring data integrity.

## Q 12- WHAT IS THE DIFFERENCE BETWEEN CHAR AND VARCHAR ?

- The main differences between CHAR and VARCHAR in SQL are:

1. **Length:**
  - **CHAR**: Fixed-length; always reserves the specified number of bytes. For example, CHAR (10) will always use 10 bytes, padding with spaces if necessary.
  - **VARCHAR**: Variable-length; uses only as much space as needed for the actual data. For example, VARCHAR (10) will use 5 bytes for the string "Hello".
2. **Performance:**
  - **CHAR**: Can be slightly faster for fixed-length data due to its consistent size.
  - **VARCHAR**: More efficient for variable-length data, as it saves space.

In general, use CHAR for fixed-length strings and VARCHAR for variable-length strings.

## Q 13- WHAT IS A SCALAR FUNCTION IN SQL ?

- A scalar function in SQL is a function that takes one or more input values and returns a single value. It operates on individual values rather than sets of values. Examples include mathematical functions (like ABS), string functions (like UPPER), and date functions (like GETDATE). Scalar functions are often used in SQL expressions, SELECT statements, and WHERE clauses to manipulate or evaluate data.

## Q 14- WHAT IS AN AGREEGATE FUNCTION IN SQL ?

- An aggregate function in SQL performs a calculation on a set of values and returns a single summary value. Common aggregate functions include:
  1. **SUM**: Calculates the total of a numeric column.
  2. **AVG**: Computes the average of a numeric column.
  3. **COUNT**: Counts the number of rows or non-NULL values.
  4. **MAX**: Returns the maximum value from a column.
  5. **MIN**: Returns the minimum value from a column.

Aggregate functions are often used with the GROUP BY clause to summarize data based on specific criteria.

## Q 15- WHAT IS GROUP BY ?

- The GROUP BY clause in SQL is used to group rows that have the same values in specified columns into summary rows. It is commonly used with aggregate functions (like COUNT, SUM, AVG) to perform calculations on each group. For example, `SELECT column1, COUNT(*) FROM table_name GROUP BY column1;` groups the results by column1 and counts the number of rows in each group. This helps in generating reports and summarizing data effectively.

## Q 16- WHAT IS HAVING CLAUSE ?

- The HAVING clause in SQL is used to filter the results of a GROUP BY operation based on a specified condition. Unlike the WHERE clause, which filters rows before aggregation, HAVING filters groups after the aggregation has been performed. For example, `SELECT column1, COUNT(*) FROM table_name GROUP BY column1 HAVING COUNT(*) > 5;` retrieves only those groups where the count of rows exceeds 5. This is useful for applying conditions to aggregated data.

## Q 17- WHAT IS AN ORDER BY CLAUSE ?

- The ORDER BY clause in SQL is used to sort the result set of a query based on one or more columns. It allows you to specify the sort order as ascending (ASC) or descending (DESC). For example, `SELECT * FROM table_name ORDER BY column_name ASC;` sorts the results by `column_name` in ascending order. The ORDER BY clause is typically used to enhance the readability of query results by presenting them in a specific sequence.

## Q 17- WHAT IS A CASE STATEMENT IN SQL ?

- A CASE statement in SQL is a conditional expression that allows you to execute different actions based on specific conditions. It can be used in SELECT, UPDATE, and other SQL statements to return values conditionally. For example:

sql

```
SELECT
    column_name,
    CASE
        WHEN condition1 THEN result1
        WHEN condition2 THEN result2
        ELSE result3
    END AS alias_name
FROM table_name;
```

This structure evaluates conditions in order and returns the corresponding result for the first true condition, making it useful for data transformation and categorization.

## Q 18- WHAT IS A CURSOR IN SQL ?

- A cursor in SQL is a database object that allows for the retrieval and manipulation of rows from a result set one at a time. It provides a way to process individual records returned by a query, enabling operations like updates or calculations on a row-by-row basis. Cursors are typically used in procedural SQL to handle complex row-level operations where set-based processing is not sufficient. For example:

```
sql
```

```
DECLARE cursor_name CURSOR FOR SELECT column_name FROM  
table_name;
```

While useful, cursors can be less efficient than set-based operations, so they should be used judiciously.

## ❖ADVANCE

## Q 1- WHAT IS THE DIFFERENCE BETWEEN DELETE AND TRUNCATE ?

- The main differences between DELETE and TRUNCATE in SQL are:

### 1. **Functionality:**

- **DELETE:** Removes specified rows from a table based on a condition (can use a WHERE clause) and logs each row deletion. It can be rolled back if within a transaction.
- **TRUNCATE:** Removes all rows from a table without logging individual row deletions. It cannot use a WHERE clause and cannot be rolled back in most databases.

### 2. **Performance:**

- **DELETE:** Slower, especially for large tables, as it logs each deletion.
- **TRUNCATE:** Faster, as it deallocates data pages rather than logging individual row deletions.



### 3. Usage:

- **DELETE:** Can be used when you need to remove specific records or maintain referential integrity.
- **TRUNCATE:** Best for quickly removing all records from a table when no integrity constraints are involved.

## Q 2- WHAT IS SCHEMA IN SQL ?

- schema in SQL is a logical structure that defines how a database is organized. It includes the tables, views, indexes, and relationships within the database, as well as their properties and constraints. Schemas help to group database objects and provide a way to manage permissions and access. In essence, a schema serves as a blueprint for the database, outlining its design and organization.

## Q 3- WHAT IS SYNONYM ?

- A synonym in SQL is an alias or alternative name for a database object, such as a table, view, or procedure. Synonyms simplify SQL statements by allowing users to reference objects with a shorter or more convenient name, without changing the actual object name. For example, you can create a synonym like this:

```
sql
```

```
CREATE SYNONYM synonym_name FOR  
schema_name.object_name;
```

This allows users to refer to `object_name` using `synonym_name`, enhancing code readability and simplifying access to complex or lengthy object names.

## Q 4- WHAT IS A CORRELATED SUBQUERY ?

- A correlated subquery is a subquery that references columns from the outer query, meaning it is evaluated once for each row processed by the outer query. This creates a dependency between the inner and outer queries. For example:

sql

```
SELECT column1  
FROM table1 t1  
WHERE column2 > (SELECT AVG(column2) FROM table2 t2  
WHERE t2.id = t1.id);
```

In this case, the inner query depends on the current row of the outer query, making it a correlated subquery. They are often used to compare values between related rows in different tables.

## Q 5- WHAT IS SQL INJECTION ?

- SQL injection is a security vulnerability that occurs when an attacker is able to manipulate SQL queries by injecting malicious SQL code through user input fields. This can allow unauthorized access to or manipulation of the database, potentially leading to data theft, modification, or deletion. To prevent SQL injection, developers should use parameterized queries, prepared statements, and proper input validation.

## Q 6- WHAT IS THE DIFFERENCE BETWEEN UNION AND UNION ALL ?

- The main differences between UNION and UNION ALL in SQL are:

### 1. Duplicates:

- **UNION:** Removes duplicate rows from the result set, returning only distinct records.
- **UNION ALL:** Includes all rows from both queries, retaining duplicates.

## 2. Performance:

- **UNION:** Slower due to the overhead of checking for and removing duplicates.
- **UNION ALL:** Faster, as it directly combines the result sets without any duplicate checks.

Use UNION when you need unique records, and UNION ALL when you want to include all records, including duplicates.

## Q 7- WHAT IS A COMPOSITE KEY ?

- A composite key in SQL is a combination of two or more columns in a table that together uniquely identify a record. Each individual column may not be unique on its own, but when combined, they ensure that each row can be uniquely identified. Composite keys are often used in junction tables to create many-to-many relationships. For example, in a table that tracks student course enrollments, a composite key might consist of `student_id` and `course_id`.

## Q 8- WHAT IS AN ALIAS IN SQL ?

- An alias in SQL is a temporary name given to a table or a column for the duration of a query. It is used to make the query more readable and to simplify complex queries. Aliases are created using the AS keyword, but it can be omitted. For example:

```
sql
```

```
SELECT column_name AS alias_name FROM table_name;
```

or simply:

```
sql
```

```
SELECT column_name alias_name FROM table_name;
```

Aliases are especially useful in cases with long or complex names or when performing calculations.

## Q 9- WHAT IS A SEQUENCE IN SQL ?

- A sequence in SQL is a database object that generates a series of unique numeric values, typically used for auto-incrementing primary key values. It provides a way to obtain sequential numbers without the need for locking tables or managing counter values manually. A sequence can be defined with options like starting value, increment value, and maximum value. For example:

Sql

```
CREATE SEQUENCE sequence_name START WITH 1 INCREMENT BY 1;
```

Sequences are useful in scenarios where unique identifiers are required for new records.

## Q 10- WHAT IS A TEMPORARY TABLE ?

- A temporary table in SQL is a special type of table that is created to store data temporarily during a session or transaction. It is automatically deleted when the session ends or when it is explicitly dropped. Temporary tables are useful for storing intermediate results, simplifying complex queries, and improving performance. They are typically prefixed with a # symbol in SQL Server (e.g., CREATE TABLE #temp\_table), but the syntax can vary by database system.

.....THANK YOU.....