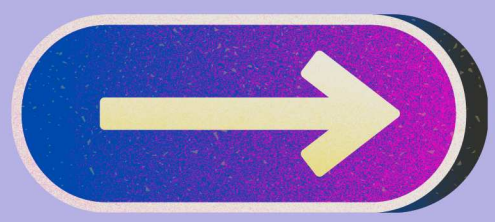
Part - 1

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What is the difference between SQL and MySQL?



Key difference:

➡ **SQL is a Query language and you can write various queries and command using SQL, in order to access the data from the database,**

whereas MYSQL is a relational database management system which is use to manage different databases.

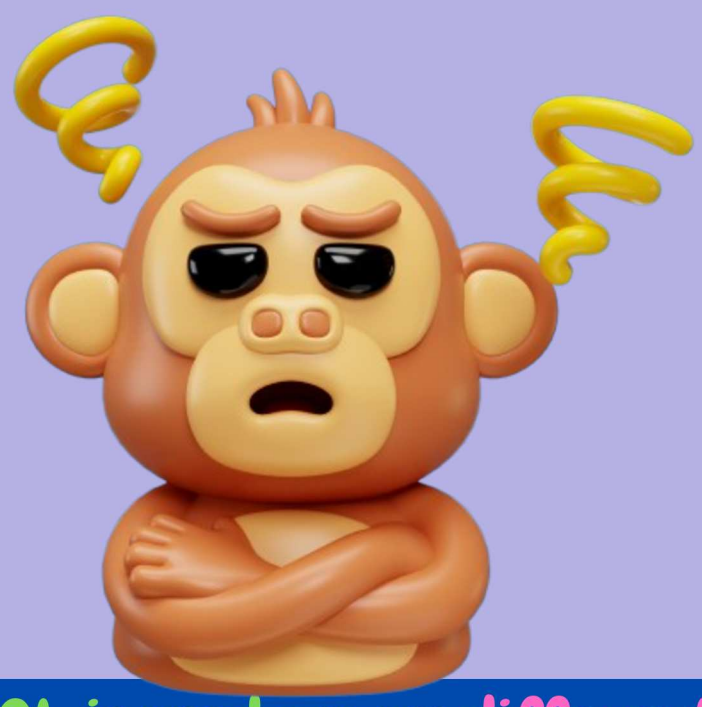
How is SQL different from MySQL?

Key difference:

➡ **SQL** is a language used for managing data in a database, while

MySQL is a database management system(Software) that uses SQL as its query language.

SQL provides the commands, and MySQL executes them within its environment.



Can you explain how SQL is used across different database systems?

Key difference:

➡ **SQL is a standardized language that is used by various database systems such as**

- **MySQL,**
- **PostgreSQL,**
- **Oracle, and others.**

Each system may have its extensions,

but they all rely on SQL for core data manipulation tasks.

What are some of the key operations you can perform with SQL?

Key difference:

➡ **With SQL, you can perform operations like**

- **querying data (SELECT),**
- **updating records (UPDATE),**
- **inserting new data (INSERT),**
- **and deleting existing data (DELETE).**

Who maintains MySQL, and how does it relate to SQL standards?

Key difference:

➡ **MySQL is maintained by Oracle Corporation. It follows the SQL standards defined by ANSI (American National Standards Institute) and ISO (International Organization for Standardization),**

However, it also includes additional features specific to MySQL.

wanna see some
Counter Questions

1. What are some alternatives to MySQL that also use SQL?

➡ **Alternatives to MySQL include**

- **PostgreSQL,**
- **SQLite,**
- **Oracle Database,**
- **and Microsoft SQL Server.**

These systems also use SQL as their query language but may offer different features and performance characteristics.

Next Question

2. Why would you choose MySQL over other SQL-based databases?

➡ **MySQL is often chosen for its**

- **ease of use,**
- **strong community support,**
- **scalability, and reliability,**
- **particularly for web applications.**

It’s also open-source, which makes it accessible to a wide range of users.

3. Can MySQL handle large-scale enterprise applications?

➡ **Yes, MySQL can handle large-scale enterprise applications, especially with its support for**

- **clustering,**
- **replication,**
- **and high-availability solutions.**

However, some enterprises may opt for other databases like Oracle or SQL Server for specific features.

Next Question

4. How does MySQL ensure data security?

- **MySQL provides several security features such as user authentication, encryption, and access control through privileges and roles.**

It also supports secure connections using SSL.

5. What are some common MySQL commands for database management?

➡ **Common MySQL commands include**

- **CREATE DATABASE for creating a new database,**
- **SHOW DATABASES to list all databases,**
- **CREATE TABLE to create a table, and**
- **ALTER TABLE to modify an existing table.**

Additionally, BACKUP DATABASE and RESTORE DATABASE are used for backup and recovery operations.

you completed one interview question with me,

can you do me a favour

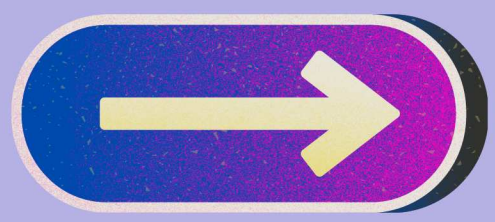
Part - 2

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

In what order does SQL execute different clauses such as SELECT, WHERE , GROUP BY etc.



Answer:

➡ **In SQL, the execution order of different clauses in a query is not the same as their written order.**

The SQL query execution order follows a specific sequence to ensure that data is processed correctly.

Here is the order in which SQL executes the various clauses:

Also give some explaination as well

Order Of Execution

1. FROM

The first step is to identify the tables involved in the query and join them if necessary.

2. WHERE

After identifying the tables, SQL applies the filters specified in the WHERE clause to restrict the rows returned.

3. GROUP BY

The first step is to identify the tables involved in the query and join them if necessary.

4. HAVING

After grouping, SQL applies the filters specified in the HAVING clause to the grouped rows.

Order Of Execution

5. SELECT

SQL then selects the columns specified in the SELECT clause from the remaining rows.

6. ORDER BY

Finally, SQL orders the result set based on the columns specified in the ORDER BY clause.

7. LIMIT/OFFSET

If present, SQL applies the LIMIT and OFFSET clauses to restrict the number of rows returned.

Can you explain these all using an example query

Example;

```sql
SELECT department,
    COUNT(employee_id) AS employee_count
FROM employees
WHERE salary > 50000

GROUP BY department
HAVING COUNT(employee_id) > 10

ORDER BY employee_count DESC
LIMIT 5;
```

can you have Explanation of the Execution Order

Explanation of Execution Order:

- **FROM employees:** Identify the table to query, employees.
- **WHERE salary > 50000:** Filter rows where the salary is greater than 50,000.
- **GROUP BY department:** Group the filtered rows by the department column
- **HAVING COUNT(employee_id) > 10:** Filter groups where the number of employees in each department is greater than 10.
- **SELECT department, COUNT(employee_id) AS employee_count:** Select the department and count of employees in each group.
- **ORDER BY employee_count DESC:** Order the result set by the employee count in descending order.
- **LIMIT 5:** Limit the result to the top 5 rows.

wanna see some

Counter Questions

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

➡ **The WHERE clause filters rows before any grouping occurs,**

while the HAVING clause filters groups after the GROUP BY clause has been applied.

2. Can you use HAVING without GROUP BY in a SQL query?

➥ **Yes, HAVING can be used without GROUP BY, but it would act like a WHERE clause**

and apply conditions to the entire result set, rather than to groups.

3. What happens if you omit the ORDER BY clause in a SQL query?

➥ **If we omit the ORDER BY clause, the result set is returned in an arbitrary order,**

which is typically the order in which rows are retrieved from the database.

Next Question

4. Is it possible to use aggregate functions in the WHERE clause?

➡ **No, aggregate functions cannot be used in the WHERE clause.**

They should be used in the SELECT or HAVING clauses, as the WHERE clause is applied before grouping.

5. How does SQL handle complex queries with subqueries or joins?

- **SQL executes subqueries first and then processes the main query.**

In the case of joins, SQL first joins the tables specified in the FROM clause before applying other clauses like WHERE, GROUP BY, etc.

you completed one interview question with me,

can you do me a favour

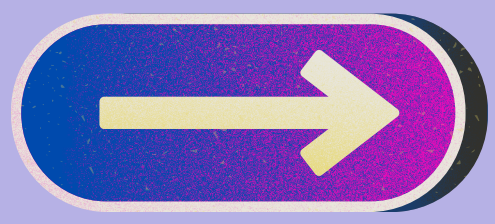
Part - 3

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What is the difference between DROP and TRUNCATE statements?



How drop Works;

➡ **The DROP statement completely removes a table from the database, including its data, structure, and all associated elements like indexes, constraints, and permissions.**

Example:

```
DROP TABLE employees;
```

- **This command deletes the employees table and everything related to it from the database.**

How TRUNCATE works

TRUNCATE;

➡ **The TRUNCATE statement deletes all rows in a table but keeps the table structure intact.**

It is typically faster than DELETE because it uses fewer system resources and minimal logging.

Example:

```
TRUNCATE TABLE employees;
```

- **This command removes all data from the employees table but leaves the table structure, indexes, and constraints in place.**

Difference between DROP and TRUNCATE

Difference;

- **DROP** Removes the entire table, including its structure and data.
- **TRUNCATE** Removes only the data, keeping the table structure.

- **DROP** Can be slower due to the additional tasks of deleting related objects and the table structure.
- **TRUNCATE** is Faster than DELETE because it uses minimal logging and doesn't delete the structure.

- **DROP** Cannot be rolled back; once executed, it permanently deletes the table.
- **TRUNCATE** Typically cannot be rolled back; it permanently deletes the data.

- **DROP** Affects the database schema by removing the table and its relationships.
- **TRUNCATE** Does not affect the schema; only removes data.

- **DROP** is Used when you need to permanently delete a table and all its contents.
- **TRUNCATE** is used when you need to quickly clear all data from a table but plan to reuse the table structure.

Example;

➡ **Scenario:** You no longer need the employees table and want to remove it completely from the database.

Use DROP

```
DROP TABLE employees;
```

➡ **Scenario:** You want to quickly clear out all employee records but keep the table for future use.

Use TRUNCATE

```
TRUNCATE TABLE employees;
```

wanna see some

Counter Questions

1. Can TRUNCATE trigger a DELETE trigger in SQL?

➡️ **No, TRUNCATE does not trigger a DELETE trigger because it is not considered a row-level operation.**

It works on the table as a whole and bypasses any DELETE triggers that might be set on the table.

2. What happens to the table's constraints and indexes when you use TRUNCATE?

➥ **When we use TRUNCATE, the table's constraints and indexes remain intact.**

The operation only removes the data, leaving the table structure, including constraints, indexes, and relationships, unchanged.

3. Is it possible to TRUNCATE a table that is referenced by a foreign key?

➡ **No, you cannot TRUNCATE a table that is referenced by a foreign key constraint.**

The TRUNCATE operation requires that there be no foreign key dependencies on the table, as it cannot be rolled back and would violate referential integrity.

Next Question

4. How does the performance of TRUNCATE compare to DELETE?

➡ **TRUNCATE is generally faster than DELETE because it deallocates the data pages used to store the table data, rather than logging each row deletion individually.**

This makes TRUNCATE more efficient for clearing all data from a table, especially for large tables.

5. When should you use DROP versus TRUNCATE?

➡ **Use DROP when you want to permanently delete a table and all its contents from the database.**

Use TRUNCATE when you want to quickly clear out all data from a table but keep the table structure intact for future use.

you completed one interview question with me,

can you do me a favour

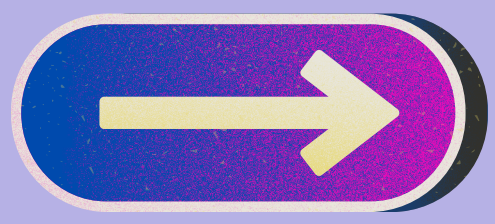
Part - 4

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What is Alias in SQL?



What is Alias

➡ **An alias in SQL is a temporary name given to a table or column within a SQL query to make it easier to refer to, especially when names are long or complex. It's similar to giving a nickname to make things simpler.**

Why are aliases used in SQL?

Why it aliases used

- **Aliases are used to make SQL queries more readable and concise.**

They help to shorten long table or column names and are particularly useful when dealing with complex queries involving multiple tables or columns with similar names.

Do aliases change the actual names of tables or columns in the database?

Alias in database

➡ **No, aliases are temporary and only exist during the execution of a query.**

They do not change the actual names of the columns or tables in the database.

Once the query execution is complete, the aliases are no longer in effect.

How do you create an alias for a column in SQL?

Create an alias for a column in SQL?

➡ **To create an alias for a column, you use the AS keyword followed by the alias name.**

For example,

SELECT customer_id AS cid FROM customer;

Here, cid is the alias for the customer_id column.

wanna see some

Counter Questions

1. Can you use aliases without the AS keyword in SQL?

➡ **Yes, in many SQL databases, you can use aliases without the AS keyword. For example,**

SELECT customer_id cid FROM customer;

will work the same as using AS. However, using AS makes the query more readable and is often considered a best practice.

2. Can you use aliases in the WHERE clause of a SQL query?

➡ **No, you cannot use column aliases directly in the WHERE clause because the WHERE clause is evaluated before the SELECT clause.**

However, you can use aliases in the ORDER BY and GROUP BY clauses since these are evaluated after the SELECT clause.

3. Are aliases case-sensitive in SQL?

➡ **It depends on the SQL database being used. In some databases, aliases are case-insensitive by default, meaning CID and cid would be considered the same.**

In others, aliases might be case-sensitive, requiring the use of consistent case or quotation marks.

Next Question

4. Can you use an alias in the HAVING clause of a SQL query?

➡ **Yes, you can use column aliases in the HAVING clause because the HAVING clause is evaluated after the GROUP BY clause and after the SELECT clause, where aliases are defined.**

This makes them accessible in HAVING conditions.

5. Can you alias multiple columns in a SQL query? How?

➡ **Yes, you can alias multiple columns in a SQL query by specifying each alias individually.**

For example:

```
SELECT customer_id AS cid, customer_name AS cname FROM
customer;
```

you completed one interview question with me,

can you do me a favour

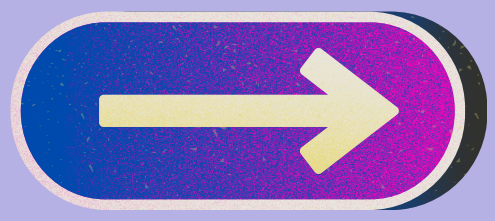
Part - 5

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What is the similarities between DROP and TRUNCATE ?



Why they are used in SQL

➥ **Both DROP and TRUNCATE are used to remove data from a database.**

- **TRUNCATE removes all rows from a table but keeps the table structure intact, while**

- **DROP removes the entire table along with its structure and data.**

Are DROP and TRUNCATE operations reversible?

DROP and TRUNCATE

➡ **No, both DROP and TRUNCATE are irreversible operations.**

Once executed, the changes cannot be undone, and

the data cannot be recovered without a backup. They are considered destructive commands.

How do DROP and TRUNCATE differ in terms of what they remove?

DROP and TRUNCATE

➡**TRUNCATE removes all rows from a table but retains the table structure for future use.**

In contrast, DROP removes both the table's data and its structure, effectively deleting the table from the database.

Why are DROP and TRUNCATE considered faster than DELETE?

BIG YES;

➡ **Both DROP and TRUNCATE are faster than DELETE because they do not log individual row deletions.**

TRUNCATE deallocates data pages used by the table, and

DROP removes the entire table definition, which requires less processing.

When should you use TRUNCATE instead of DROP?

DROP and TRUNCATE

➡ **Use TRUNCATE when you want to quickly remove all rows from a table but plan to reuse the table structure.**

Use DROP when you want to permanently delete a table and its structure from the database.

1. Can you TRUNCATE a table that has a foreign key constraint?

➡ **No, you cannot TRUNCATE a table if it is referenced by a foreign key constraint.**

You would need to either drop the foreign key constraint first or use a DELETE statement to remove rows while maintaining referential integrity.

2. Does TRUNCATE reset identity columns in a table?

➡ **Yes, TRUNCATE typically resets the identity seed of an identity column to its initial value,**

unlike DELETE, which does not affect the identity seed.

3. Can DROP be used to remove other database objects besides tables?

➥ **Yes, DROP can be used to remove other database objects such as views, stored procedures, indexes, and even entire databases.**

The syntax changes slightly depending on the object being dropped

(e.g., DROP VIEW view_name;).

Next Question

4. How does TRUNCATE handle triggers on a table?

➡ **TRUNCATE does not activate triggers defined on a table.**

If there are any triggers that you want to execute during data deletion, you should use the DELETE statement instead.

5. Is it possible to TRUNCATE or DROP a table within a transaction?

➥ **In most database systems, you can TRUNCATE a table within a transaction, but the effect might not be rolled back, depending on the system.**

DROP operations are usually allowed within a transaction and can be rolled back if the transaction is not committed.

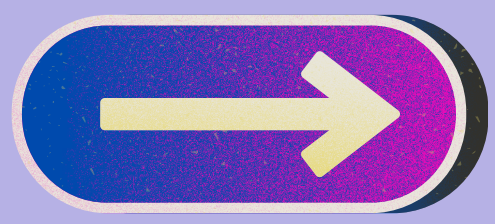
Part - 6

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What are the constraints in SQL?



What are constraints in SQL, and why are they important?

➡ **Constraints in SQL are rules that enforce restrictions on the data in a table. They are essential for**

- **ensuring data integrity,**
- **consistency, and**
- **accuracy within a database.**

Constraints prevent invalid data entry and help maintain reliable relationships between tables.

What is a Primary Key Constraint?

Primary Key Constraint

➥ **A Primary Key Constraint ensures that each record in a table is uniquely identifiable.**

It means that values in the primary key column(s) must be unique and cannot be NULL.

For example: a student_id column in a students table could be defined as a primary key.

How does a Foreign Key Constraint work?

Foreign Key Constraint work

➡ **A Foreign Key Constraint establishes a relationship between two tables.**

It ensures referential integrity by requiring that values in a column match values in another table's primary key or unique key.

This constraint prevents invalid data from being entered into a table, maintaining consistent relationships.

What is the purpose of a Unique Constraint?

Purpose of a Unique Constraint

- **A Unique Constraint ensures that all values in a column (or a group of columns) are unique across the table.**

Unlike the primary key, a unique constraint allows NULL values, but only one NULL value is permitted in a column.

It's used to enforce uniqueness without setting a column as a primary key.

How does a Check Constraint help maintain data integrity?

Check Constraint in maintain data integrity

➡ **A Check Constraint defines a condition that must be met for values being inserted or updated in a column.**

It restricts the range or type of acceptable values for that column, ensuring that only valid data is entered.

For instance: a check constraint on a salary column might enforce that salaries are non-negative.

What is the Not Null Constraint in SQL?

Not Null Constraint in SQL

➡ **The Not Null Constraint ensures that a column cannot contain NULL values.**

It requires that every row in the table has a value for that column.

This constraint is commonly used for fields that must always have data, like product_name in a products table

1. Can a table have more than one Primary Key Constraint?

➡ **No, a table can only have one primary key constraint.**

However, the primary key can consist of multiple columns (a composite key) to ensure unique identification of records.

2. What is the difference between a Unique Constraint and a Primary Key Constraint?

➡ **Both ensure uniqueness of values in a column, but a primary key constraint does not allow NULL values and uniquely identifies each record in a table.**

A unique constraint, on the other hand, can allow one NULL value per column and does not necessarily uniquely identify each record.

3. What happens if you try to delete a row in a table that is referenced by a Foreign Key Constraint in another table?

➥ **If a foreign key constraint exists, attempting to delete a referenced row will result in an error unless cascading options are defined.**

(ON DELETE CASCADE or ON DELETE SET NULL)

These options either delete the related records or set the foreign key to NULL in the referencing table.

4. How do you modify or remove a constraint from a table in SQL?

➡ **To modify or remove a constraint, you typically use the ALTER TABLE statement.**

For instance, to remove a check constraint named chk_salary, you would use:

```
ALTER TABLE employees DROP CONSTRAINT chk_salary;
```

To add or modify a constraint, the ALTER TABLE statement can also include ADD CONSTRAINT clauses.

5. Can you have a Check Constraint on multiple columns in SQL?

➡ **Yes, a check constraint can be defined on multiple columns.**

For example: you can ensure that the start_date is always before the end_date in a table by using a check constraint that compares these two columns.

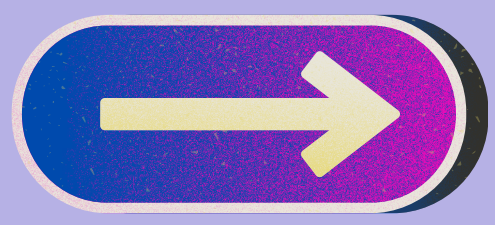
Part - 7

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What are DDL and DML languages?



What are DDL and DML in SQL?

➡ **DDL (Data Definition Language) and DML (Data Manipulation Language) are two subsets of SQL used for different purposes.**

- **DDL defines and modifies the structure of a database, while**
- **DML manipulates and manages the data within the database.**

What does DDL stand for, and what is its purpose?

DDL and its purpose?

➡ **DDL stands for Data Definition Language. It is used to define the structure or schema of a database. Commands like**

- **CREATE,**
- **ALTER, and**
- **DROP**

Are part of DDL and are used by database designers or administrators to create, modify, or delete database structures.

Can you explain what DML is and its primary function?

DML is and its primary function;

- **DML stands for Data Manipulation Language. It is used to manipulate and manage data within an existing database.**

Common DML commands include

- **SELECT,**
- **INSERT,**
- **UPDATE, and**
- **DELETE**

which allow users to retrieve, add, modify, or remove data from a database.

What is the difference between Procedural DML and Non-Procedural DML?

Difference between Procedural DML and Non-Procedural DML

- **Procedural DML requires the user to specify both the data they want and the procedure to obtain it.**

In contrast, Non-Procedural DML only requires users to specify what data they need, and the system determines how to retrieve it.

Non-Procedural DML is generally simpler and easier to use.

Who typically uses DDL commands, and why?

Who typically uses DDL commands, and why?

➡️ **DDL commands are typically used by database designers or database administrators (DBAs).**

These commands are not meant for end-users

because they deal with defining and altering the database structure,

which requires a deep understanding of the database system.

1. What are some examples of DDL commands, and what do they do?

➡ **Common DDL commands include:**

- **CREATE:** **Creates a new table or database.**
- **ALTER:** **Modifies an existing table structure, such as adding or dropping a column.**
- **DROP:** **Deletes a table or database entirely.**
- **TRUNCATE:** **Removes all rows from a table without deleting the table structure itself.**

2. How does DML differ from DDL in terms of database operations?

➥ **DML deals with data manipulation within the existing structure, such as**

- **inserting,**
- **updating, or**
- **deleting data.**

DDL, on the other hand, is concerned with defining, altering, and managing the structure of the database itself, not the data inside it.

3. What are the advantages of Non-Procedural DML over Procedural DML?

➡ **Non-Procedural DML is more user-friendly because it only requires users to specify what data they want without worrying about how to get it.**

This makes it easier to learn and use, especially for beginners, and allows the system to optimize query execution.

Next Question

4. Why is it important to distinguish between DDL and DML in database management?

➡ **Understanding the difference is crucial because DDL commands affect the database schema and are generally irreversible without backups,**

while DML commands manipulate the data within that structure. Misusing these commands can lead to data loss or structural issues in the database.

5. What is the significance of the COMMIT and ROLLBACK commands in DML?

➡ **COMMIT and ROLLBACK are transaction control commands used in conjunction with DML.**

COMMIT saves all changes made by DML commands to the database permanently, while

ROLLBACK undoes those changes if necessary. They are essential for ensuring data integrity during transactions.

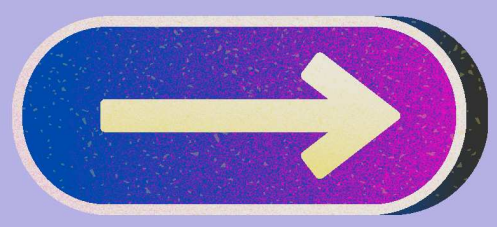
Part - 8

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What is the difference between DELETE and TRUNCATE statements?



What is the primary purpose of the DELETE statement in SQL?

➡ **The DELETE statement is used to remove specific rows from a table based on a given condition.**

It allows for selective deletion and maintains logging and triggers,

Ensuring that each deletion is recorded and any associated actions are executed.

When would you use the TRUNCATE statement instead of DELETE?

Use of TRUNCATE statement instead of DELETE

➡ **You would use the TRUNCATE statement when you need to quickly remove all rows from a table without triggering logging and triggers.**

TRUNCATE is faster than DELETE for removing all records

because it doesn't process individual row deletions.

How does DELETE differ from TRUNCATE in terms of functionality?

DELETE differ from TRUNCATE in terms of functionality

➡ **DELETE removes specific rows based on a condition, and it logs each deletion and triggers any associated actions.**

TRUNCATE, on the other hand, removes all rows from a table without logging individual deletions or triggering actions,

and it resets any auto-increment counters.

What happens to the table structure after using DELETE and TRUNCATE?

Table structure after using DELETE and TRUNCATE

➡ **After using DELETE, the table structure remains intact, and the deleted rows are simply removed based on the specified condition.**

Similarly, TRUNCATE also leaves the table structure intact but removes all rows and resets auto-increment values, if any.

Can you roll back a DELETE operation? What about TRUNCATE?

Impact on Roll back a DELETE operation and TRUNCATE?

➡ **Yes, a DELETE operation can be rolled back if it is part of a transaction that has not been committed yet.**

However, TRUNCATE is generally not reversible and cannot be rolled back,

as it is a non-logged operation.

1. How does TRUNCATE affect performance compared to DELETE??

➡ **TRUNCATE is generally faster than DELETE, especially for large tables, because it doesn't log individual row deletions or trigger associated actions.**

It deallocates the data pages used by the table directly, leading to better performance for bulk deletions.

2. Can you use a WHERE clause with TRUNCATE? Why or why not?

➡ **No, you cannot use a WHERE clause with TRUNCATE because it is designed to remove all rows from a table without any condition.**

If you need conditional deletion,

DELETE should be used instead.

3. What are the implications of using TRUNCATE on a table with foreign key constraints?

➡ **TRUNCATE cannot be used on a table with foreign key constraints unless the foreign key constraints are temporarily disabled or removed.**

This is because TRUNCATE does not allow partial deletions that could lead to referential integrity issues.

4. How does TRUNCATE handle auto-increment columns differently than DELETE?

➡ **TRUNCATE resets the auto-increment counter to the starting value, typically 1, for the table, whereas DELETE does not reset the counter.**

This means that after a TRUNCATE operation, any new row inserted will have the auto-increment column start from the beginning.

5. In what scenarios is it more appropriate to use DELETE instead of TRUNCATE?

➥ **DELETE is more appropriate when you need to selectively remove rows based on a condition, when you need to trigger actions associated with row deletions, or when you require transaction logging for auditing purposes.**

TRUNCATE is not suitable for these scenarios as it removes all rows unconditionally and without logging.

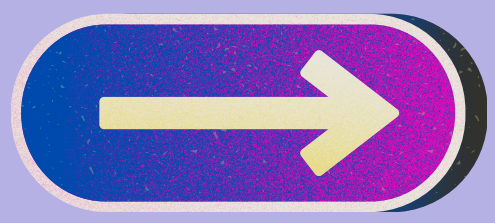
Part - 9

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What is the difference between DISTINCT and GROUP BY?



Distinct;

➥ **Distinct purpose is to remove duplicate rows from the result set. Ensures each row in the result set is unique based on the selected columns.**

Means:

```
SELECT DISTINCT column1 FROM table_name;
```

Let's understand it with a example

Example;

Suppose you have a list of fruits with duplicates:

[apple, banana, apple, orange, banana.]

```
SELECT DISTINCT fruit FROM fruits_list;
```

Using DISTINCT will give you a list with no duplicates: apple, banana, orange

What about GROUP BY

GROUP BY:

- **The purpose is to rows that have the same values in specified columns into summary rows. Typically used with aggregate functions like COUNT, SUM, AVG, etc.**

Means:

```sql
SELECT column1, COUNT(*)
FROM table_name GROUP BY column1;
```

Let's understand it with a example

Example;

Suppose you have the same list of fruits:

[apple, banana, apple, orange, banana.]

```sql
SELECT fruit, COUNT(*)
FROM fruits_list
GROUP BY fruit
```

Using GROUP BY with COUNT will give you the number of each fruit:

- **apple - 2,**
- **banana - 2,**
- **orange - 1.**

1. When should you use DISTINCT instead of GROUP BY?

➡ **Use DISTINCT when you want to remove duplicate rows from the result set based on one or more columns.**

DISTINCT is typically used when you want to return only unique values for the specified columns without any aggregation or grouping.

2. Can you use GROUP BY without an aggregate function? What happens in that case?

➥ **Yes, you can use GROUP BY without an aggregate function. In that case, GROUP BY simply returns a single row for each unique combination of values in the grouped columns.**

However, using it without an aggregate function is uncommon, as GROUP BY is primarily used for aggregation.

3. How does the performance of DISTINCT compare to GROUP BY?

➡ **Performance depends on the context, but generally, DISTINCT can be more efficient than GROUP BY when you're only removing duplicates without needing to perform any aggregation.**

GROUP BY might require additional computation, especially if aggregate functions are involved.

4. What happens if you apply DISTINCT to a query that already uses GROUP BY?

➡ **Applying DISTINCT to a query that uses GROUP BY is redundant**

because GROUP BY already ensures that each group of rows is unique based on the grouped columns.

DISTINCT would have no additional effect and could potentially slow down the query.

5. Can you combine DISTINCT with aggregate functions? How would that work?

➡ **Yes, you can combine DISTINCT with aggregate functions.**

For example, COUNT(DISTINCT column_name) counts the number of unique, non-null values in the specified column.

This is useful when you want to apply aggregation only to distinct values in a column.

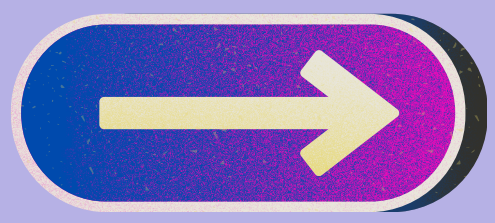
Part -10

SQL Basics Interview Questions and Answers..!

krishna kumar

@Krishan kumar

What is the difference between WHERE and HAVING clauses in SQL?



Where:

➡ **The WHERE clause in SQL is used to filter rows before any grouping takes place.**

It applies conditions to individual rows in the table,

allowing you to specify which rows you want to include in your results.

Let's understand it with a example

Example;

➡ **if you have a list of employees and you want to find only those in the Sales department, you would use the WHERE clause like this:**

```
SELECT * FROM Employees
WHERE Department = 'Sales
```

This command filters the data at the row level, retrieving only the rows that meet the specified condition.

What about HAVING

Having:

➡ **The HAVING clause is used to filter groups after the GROUP BY clause has been applied.**

This means it works on the aggregated data rather than on individual rows.

It is typically used in conjunction with aggregate functions like COUNT, SUM, or AVG.

Let's understand it with a example

Example;

if you want to find departments that have more than 10 employees, you would use the HAVING clause

```
SELECT Department,
        COUNT(*) FROM Employees
GROUP BY Department
HAVING COUNT(*) > 10;
```

- **This filters the results after grouping by department**

Ensuring that only groups meeting the condition appear in the final output. Thus, WHERE is for row-level filtering and HAVING is for group-level filtering.

wanna see some

Counter Questions

1. Can you use the HAVING clause without a GROUP BY clause? What happens?

➡ **Yes, you can use the HAVING clause without a GROUP BY clause, but it behaves like a WHERE clause in that context.**

However, it is uncommon to do so since HAVING is generally used with aggregate functions.

2. What would happen if you try to use an aggregate function in the WHERE clause?

➥ **Using an aggregate function in the WHERE clause will result in an error because WHERE is evaluated before aggregation occurs.**

Aggregate functions should be used in the HAVING clause.

3. Can WHERE and HAVING clauses be used together in the same query? How?

➡ **Yes, WHERE and HAVING clauses can be used together in the same query.**

The WHERE clause filters rows before aggregation, and the HAVING clause filters the aggregated results.

For example:

```
SELECT department, SUM(salary)
FROM employees
WHERE salary > 30000
GROUP BY department
HAVING SUM(salary) > 100000
```

4. If you need to filter results based on an aggregate condition, why can't you use WHERE instead of HAVING?

➡ **WHERE cannot be used to filter results based on an aggregate condition because WHERE is applied before aggregation.**

HAVING is necessary because it filters the results after aggregation

5.Is it possible to use WHERE and HAVING clauses to filter the same condition? How would that work?

➡ **It's possible but uncommon to filter the same condition with both WHERE and HAVING. Typically, WHERE is used to filter raw data before aggregation, and HAVING refines the result after aggregation. However, if you need to apply the same condition before and after aggregation**

it can be done like this:

```
SELECT department, SUM(salary)
FROM employees
WHERE salary > 30000
GROUP BY department
HAVING SUM(salary) > 100000
```

Find This Useful

Time to hit that like button
and give it some love! 🤩

Visit my Linkedin for such amazing Content😉

in krishan kumar