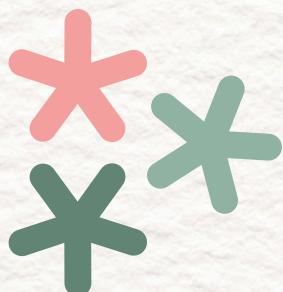




tops technologies

SQL

constraints



## Que. I

Constraints in SQL are rules or restrictions enforced on a database table's columns to ensure the integrity, accuracy, and reliability of the data. They help maintain the quality of data by preventing invalid data entry and enforcing relationships between tables.

Constraints can be applied at two levels:

1. Column Level: Applies to a single column.
2. Table Level: Applies to multiple columns or the entire table.

### *Types of Constraints*

#### *NOT NULL Constraint*

- Ensures that a column cannot have a NULL value.
- Useful for fields where data is mandatory.

#### *UNIQUE Constraint*

- Ensures all values in a column (or combination of columns) are unique.
- Allows one NULL value per column.

#### *PRIMARY KEY Constraint*

- Combines the NOT NULL and UNIQUE constraints.
- Uniquely identifies each row in a table.
- A table can have only one primary key.

## Que. 2

*The PRIMARY KEY and FOREIGN KEY constraints serve different purposes in SQL and are used to maintain data integrity in relational databases.*

### ***PRIMARY KEY***

I. ***Purpose:***

- *Uniquely identifies each record in a table.*

***Characteristics:***

- *Combines the properties of NOT NULL and UNIQUE.*
- *Ensures that no two rows in the table can have the same value for the primary key column(s).*
- *A table can have only one primary key, but it can consist of multiple columns (composite key).*
- *The value in a primary key column cannot be NULL.*

***Scope:***

- *Exists within the table it is defined.*

***Use Case:***

- *Typically used to identify each record in a table uniquely (e.g., employee\_id, order\_id).*

# **FOREIGN KEY**

## **Purpose:**

- Establishes a relationship between two tables by linking a column in one table to the primary key in another table.

## **Characteristics:**

- Enforces referential integrity by ensuring that the value in the foreign key column exists in the referenced table's primary key column.

○ A table can have multiple foreign keys.

## **Scope:**

- References a column (or columns) in another table.

## **Use Case:**

- Used to create relationships between tables (e.g., linking orders to customers by customer\_id).

## Que. 3

he NOT NULL and UNIQUE constraints play crucial roles in maintaining data integrity in a database.

## **NOTNULL Constraint**

### **Role:**

- Ensures that a column cannot have a NULL value.
- Used to enforce that data is always provided for the specified column.

### **Purpose:**

- Guarantees that a column will always have a valid (non-null) value.
- Useful for mandatory fields such as name, email, or ID.

## UNIQUE Constraint

### *Role:*

1.
  - Ensures that all values in a column (or a combination of columns) are distinct.
  - Prevents duplicate entries for the specified column.

### *Purpose:*

2.
  - Guarantees that no two rows have the same value in the specified column(s).
  - Useful for enforcing rules like unique usernames, email addresses, or IDs.

### *Characteristics:*

3.
  - Can be applied to one or more columns (unique constraints on multiple columns create a composite uniqueness requirement).
  - Allows a single NULL value in the column (in most database systems).

### *Use Case:*

4.
  - Fields where each entry must be unique, such as email, phone\_number, or username.
  - Example: Ensuring no two users share the same email address.