DAY-III SQL M.ROHIT MCA, M. TECH

SQL Constraints: Overview

SQL **constraints** are rules enforced on data columns to ensure **accuracy**, **reliability**, and **integrity** of data in the database.

✓ Common SQL Constraints & Real-time Uses

Constraint	Description	Real-time Usage in MNC Projects
NOT NULL	Prevents null values in a column.	Ensures mandatory fields like email, username in user registration tables are not left blank.
UNIQUE	Ensures all values in a column are unique.	Used for fields like email, employee_id to prevent duplicates in HRMS or user management systems.
PRIMARY KEY	Uniquely identifies each row in a table.	Used in almost every table for unique identity (e.g., order_id, customer_id).
FOREIGN KEY	Maintains referential integrity by linking to a primary key in another table.	Used in ecommerce apps (e.g., order.customer_id references customer.id) to maintain relationships.
CHECK	Ensures the value in a column meets a specific condition.	Used to validate values (e.g., age > 18 in user forms, salary >= 0 in payroll systems).
DEFAULT	Sets a default value if no value is specified.	Automatically assigns values like status = 'active' or created_at = current_timestamp.
INDEX	Improves query performance (not technically a constraint, but used often).	Used on large columns like email, order_id for faster searches and joins.
AUTO_INCREMENT / IDENTITY	Auto-generates unique values for primary key columns.	Used to generate sequential IDs for tables like orders, invoices, support_tickets.

DAY-III SQL M.ROHIT MCA, M. TECH

© SQL Constraints in Real-time MNC Projects

1. Banking System

- CHECK (balance >= 0) in account table.
- FOREIGN KEY (customer_id) in transactions table → ensures no transaction without a valid customer.

2. Ecommerce Platform

- UNIQUE (email) on customer table to avoid multiple registrations.
- FOREIGN KEY (product_id) in orders table to ensure only valid products are ordered.

3. HR Management System

- PRIMARY KEY (employee_id) to uniquely identify employees.
- CHECK (age >= 18) for legal employment compliance.
- DEFAULT 'Active' for new employee status.

laterview Questions (Point-wise) on SQL Constraints

Basic Questions

- 1. What are SQL constraints?
- 2. Name the different types of SQL constraints.
- 3. What is the difference between PRIMARY KEY and UNIQUE?
- 4. Can a column have both NOT NULL and UNIQUE? Why would you use this?

□ Intermediate Questions

- 5. What is the purpose of a FOREIGN KEY? Can you give a real-life example?
- 6. Can a table have multiple FOREIGN KEYs?
- 7. Can a PRIMARY KEY be NULL? Can a FOREIGN KEY be NULL?
- 8. What's the difference between CHECK and DEFAULT constraints?
- 9. When would you use a CHECK constraint instead of handling validation in the application layer?

Advanced / Real-Time Scenarios

- 10. Suppose you're building a payment system. What constraints would you apply on the payment_amount, status, and transaction_id columns?
- 11. How would you enforce that an order cannot be placed for a product that doesn't exist?
- 12. Can you modify an existing table to add a constraint? How?

DAY-III SQL M.ROHIT MCA, M. TECH

- 13. What happens when you try to insert a row that violates a FOREIGN KEY constraint?
- 14. How do constraints help in maintaining data integrity in a multi-user environment?
- 15. How would you handle a situation where a business rule changes and an existing CHECK constraint becomes invalid?

□ Pro Tip for Interviews:

Always relate SQL constraint questions to **real business use-cases** you've read about or worked on. This shows you're not just memorizing theory, but understand **why** it's used.

☑ Basic SQL Constraints Questions and Answers

1. What are SQL constraints?

Answer:

SQL constraints are rules applied to table columns to enforce data integrity, consistency, and validity. They ensure that only valid data is stored in the database.

2. Name the different types of SQL constraints.

Answer:

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT

3. What is the difference between PRIMARY KEY and UNIQUE?

Answer:

 PRIMARY KEY: Ensures uniqueness and does not allow NULLs. Only one primary key per table.

DAY-III

SQL

M.ROHIT MCA, M. TECH

 UNIQUE: Ensures uniqueness but allows one NULL value (depending on the DBMS). You can have multiple unique constraints per table.

4. Can a column have both NOT NULL and UNIQUE? Why would you use this?

Answer:

Yes. This combination ensures that a column must always have a value (NOT NULL) and that the value must be unique (UNIQUE).

Example: Email addresses in a user registration system.

ு Intermediate SQL Constraint Questions

5. What is the purpose of a FOREIGN KEY? Can you give a real-life example?

Answer:

A FOREIGN KEY ensures referential integrity between two tables by creating a link.

Example:

In an order management system: orders.customer_id → references customers.id.

6. Can a table have multiple FOREIGN KEYs?

Answer:

Yes, a table can have multiple foreign keys referencing different parent tables or even the same table.

7. Can a PRIMARY KEY be NULL? Can a FOREIGN KEY be NULL?

Answer:

- PRIMARY KEY: Cannot be NULL.
- FOREIGN KEY: **Can** be NULL (if the relationship is optional).

8. What's the difference between CHECK and DEFAULT constraints?

Answer:

- CHECK: Validates that the data meets a specific condition (e.g., age ≥ 18).
- DEFAULT: Assigns a default value if none is provided during insertion.

4 | Page R O # I T

DAY-III

SQL

M.ROHIT MCA, M. TECH

9. When would you use a CHECK constraint instead of handling validation in the application layer?

Answer:

Use CHECK to enforce critical business rules **at the database level** for consistency and security, especially in multi-client environments where app-level validation may be bypassed.

Advanced / Real-Time Scenarios

10. Suppose you're building a payment system. What constraints would you apply on the payment_amount, status, and transaction_id columns?

Answer:

- payment_amount: CHECK (payment_amount > 0)
- status: DEFAULT 'Pending' + optional CHECK (status IN ('Pending', 'Success', 'Failed'))
- transaction_id: PRIMARY KEY or UNIQUE NOT NULL

11. How would you enforce that an order cannot be placed for a product that doesn't exist?

Answer:

By using a FOREIGN KEY constraint: orders.product_id → products.product_id
This ensures that only existing products can be used in an order.

12. Can you modify an existing table to add a constraint? How?

Answer:

Yes, using ALTER TABLE. Example:

ALTER TABLE employees ADD CONSTRAINT chk_age CHECK (age >= 18);

13. What happens when you try to insert a row that violates a FOREIGN KEY constraint?

Answer:

The database throws an error and **rejects** the insert or update operation, preventing invalid data from being added.

14. How do constraints help in maintaining data integrity in a multi-user environment?

SQL

M.ROHIT MCA, M. TECH

Answer:

Constraints prevent invalid, duplicate, or inconsistent data regardless of how many users are interacting with the system, maintaining **consistent and reliable** data.

15. How would you handle a situation where a business rule changes and an existing CHECK constraint becomes invalid?

Answer:

You can **drop** and **recreate** the constraint:

ALTER TABLE employees DROP CONSTRAINT chk_age;

ALTER TABLE employees ADD CONSTRAINT chk_age CHECK (age >= 21);

□ Pro Tips for Interview:

- Always support answers with real-life use cases.
- Mention any specific DBMS behavior (e.g., MySQL allows multiple NULLs in a UNIQUE column, SQL Server does not).
- Be prepared to write SQL queries related to adding/dropping constraints.

✓ SQL Constraints –

☐ Part 1: Multiple Choice Questions (MCQs)

- Q1. Which SQL constraint ensures that a column cannot have a NULL value?
- A. PRIMARY KEY
- B. UNIQUE
- C. NOT NULL
- D. DEFAULT
- ✓ Answer: C
- **Q2.** Which constraint automatically provides a value if none is supplied during insertion?
- A. UNIQUE
- B. CHECK
- C. FOREIGN KEY
- D. DEFAULT
- ✓ **Answer:** D

D&Y-III

SQL

M.ROHIT MCA, M. TECH

Q3. Which constraint is used to maintain referential integrity between two tables?

- A. FOREIGN KEY
- **B. PRIMARY KEY**
- C. UNIQUE
- D. DEFAULT
- ✓ Answer: A
- **Q4.** Which of the following statements is TRUE about UNIQUE and PRIMARY KEY?
- A. Both allow multiple NULLs.
- B. PRIMARY KEY allows NULL but UNIQUE doesn't.
- C. UNIQUE allows NULLs, but PRIMARY KEY does not.
- D. Neither allows NULLs.
- ✓ Answer: C
- **Q5.** You want to restrict salary values in the employees table to only values >= 3000. Which constraint should you use?
- A. DEFAULT
- **B. FOREIGN KEY**
- C. CHECK
- D. NOT NULL
- **✓ Answer:** C

A Part 2: Short Answer Questions

Q6. What is the difference between CHECK and FOREIGN KEY constraints?

✓ Answer:

- CHECK: Validates data based on a condition within the same table.
- FOREIGN KEY: Ensures values in one table exist in another, maintaining referential integrity.
- Q7. Can a table have more than one UNIQUE constraint?

✓ Answer:

Yes, a table can have multiple UNIQUE constraints on different columns or combinations of columns.

Q8. What happens if you try to insert a duplicate value in a column with a UNIQUE constraint?

✓ Answer:

The database will reject the insert and throw a constraint violation error.

Q9. What are the differences between NOT NULL and DEFAULT constraints? **Answer:**

- NOT NULL: Prevents NULL values in a column.
- DEFAULT: Automatically assigns a value if none is provided.

Part 3: Query-Based Questions

Q10. Write a SQL query to create a students table with the following constraints:

- student_id: primary key
- email: must be unique and not null
- age: must be >= 18
- status: default value 'active'

```
CREATE TABLE students (

student_id INT PRIMARY KEY,

email VARCHAR(255) UNIQUE NOT NULL,

age INT CHECK (age >= 18),

status VARCHAR(50) DEFAULT 'active'
);
```

Q11. Alter the employees table to add a CHECK constraint on salary to allow only salaries greater than 0.

ALTER TABLE employees

ADD CONSTRAINT chk_salary CHECK (salary > 0);

Q12. Create two tables departments and employees where employees.department_id is a foreign key referencing departments.department_id.

```
CREATE TABLE departments (

department_id INT PRIMARY KEY,

department_name VARCHAR(100)
);
```

8 | Page R O # I T

```
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    name VARCHAR(100),
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

Part 4: Scenario-Based Questions

Q13. You are building a bank_accounts table. Which constraints would you apply to ensure:

- Every account has a unique account number.
- Balance cannot be negative.
- Account must be linked to an existing customer.

✓ Answer:

- account_number: PRIMARY KEY or UNIQUE NOT NULL
- balance: CHECK (balance >= 0)
- customer_id: FOREIGN KEY referencing customers(customer_id)

Q14. You find that users are able to insert rows without specifying a status in the orders table. What constraint would you add to ensure a default status is applied?

✓ Answer:

Add a DEFAULT constraint to the status column:

ALTER TABLE orders

MODIFY status VARCHAR(50) DEFAULT 'Pending';

(Note: syntax may vary by DBMS)

Q15. A UNIQUE constraint is applied on the email column, but the system is still allowing duplicate NULLs. Why?

✓ Answer:

Because in most DBMSs (e.g., MySQL), UNIQUE allows **multiple NULLs**, as NULL is treated as "unknown" and not equal to another NULL.