Constraints Demo

MUKESH KUMAR

Table: Users

- We’ll apply these constraints:
- NOT NULL – Name and Email must be provided
- UNIQUE – Email must be unique
- PRIMARY KEY – user_id must be unique and not null
- CHECK – Age must be between 18 and 100
- DEFAULT – Status defaults to 'active'

Step 1: Create Table with Constraints

```
CREATE TABLE Users (
  user_id INT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  age INT CHECK (age BETWEEN 18 AND 100),
  status VARCHAR(20) DEFAULT 'active'
);
```


```
CREATE TABLE Users (
  user_id INT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  age INT CHECK (age BETWEEN 18 AND 100),
  status VARCHAR(20) DEFAULT 'active'
);
```

Step 2: Insert Valid Data

INSERT INTO Users (user_id, name, email, age)
VALUES (1, 'Alice', 'alice@example.com', 25);

- **This works — all constraints are satisfied.**

```
INSERT INTO Users (user_id, name, email, age)
VALUES (1, 'Alice', 'alice@example.com', 25);
```

Step 3: Violate a Constraint — NOT NULL

INSERT INTO Users (user_id, name, email, age)
VALUES (2, NULL, 'bob@example.com', 30);

Error: name cannot be NULL (violates NOT NULL constraint)

```
INSERT INTO Users (user_id, name, email, age)
VALUES (2, NULL, 'bob@example.com', 30);
```

Step 4: Violate UNIQUE

INSERT INTO Users (user_id, name, email, age)
VALUES (3, 'Charlie', 'alice@example.com', 35);

Error: Duplicate entry for email (violates UNIQUE constraint)

```sql
INSERT INTO Users (user_id, name, email, age)
VALUES (3, 'Charlie', 'alice@example.com', 35);
```

Step 5: Violate CHECK

INSERT INTO Users (user_id, name, email, age)
VALUES (4, 'David', 'david@example.com', 120);

Error: Age is out of range (violates CHECK constraint)

```
INSERT INTO Users (user_id, name, email, age)
VALUES (4, 'David', 'david@example.com', 120);
```

Step 6: Use DEFAULT Value

INSERT INTO Users (user_id, name, email, age)
VALUES (5, 'Eve', 'eve@example.com', 29);

- This works — status will be 'active' automatically.

```
INSERT INTO Users (user_id, name, email, age)
VALUES (5, 'Eve', 'eve@example.com', 29);
```

Step 7: See Table Contents

```
SELECT * FROM Users;
```

Summary of Used Constraints

Constraint	Purpose
NOT NULL	Ensures name and email are mandatory
UNIQUE	Prevents duplicate emails
PRIMARY KEY	Makes user_id unique and not null
CHECK	Validates age range
DEFAULT	Auto-assigns 'active' status