Experiment No 3

Aim: To apply SQL commands involving constraints such as Primary key, Unique key & NOT NULL in a database management system.

Aim: To apply SQL commands involving constraints such as Primary key, Unique key & NOT NULL in a database management system.

Theory: Constraints in SQL are used to enforce rules on the data in tables. They help maintain the integrity & accuracy of the data. The key constraints involved are:

i) PRIMARY KEY: A column (or a set of columns) that uniquely identifies each record in a table. It does not allow NULL values & ensures uniqueness.

ii) UNIQUE KEY: Ensures that all values in a column are distinct. Unlike the Primary key, a unique key can accept NULL values.

iii) NOT NULL: Ensures that a column cannot have NULL values.

SQL Commands: Execution:

i) Creating a table with constraints:

```
CREATE TABLE STUDENTS (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE,
```

```
    AGE INT
);
```

StudentID is a primary key, meaning each student must have a unique & not-null identifier.
Name must have a value due to NOT NULL constraint.
Email must be unique for each student, but can accept NULL values.
Age DOES NOT have any ~~const~~ constraints, meaning it can store NULL values or duplicate values.

ii) Inserting data into table:

```
INSERT INTO Students (StudentID, Name, Email, Age)
    VALUES (1, 'Alice', 'alice123@gmail.com', 20);

INSERT INTO Students (StudentID, Name, Email, Age)
    VALUES (2, 'Bob', 'bobo@gmail.com', 20);

INSERT INTO Students (studentID, Name, Email, Age)
    VALUES (3, 'Charlie', NULL, 21);
```

The 3rd record with studentID = 3 was inserted successfully even though the Email field is NULL.

Trying to insert a NULL value in the Name column will cause an error due to the NOT NULL constraint

iii) Attempting Constraint Violations :

```
--Violating PRIMARY KEY constraint (Duplicate
-- StudentID
INSERT INTO Students (StudentID, Name, Email,
                      Age)
    VALUES (1, 'David', 'davidbom@gmail.com', 23);

--Violating UNIQE constraint (Duplicate Email)
INSERT INTO Students (StudentID, Name, Email,
                      Age)
    VALUES (4, 'Eve', 'alicen23@gmail.com', 24);

--Violating NOT NULL constraint
INSERT INTO Students (StudentID, Name, Email,
                      Age)
    VALUES (5, NULL, 'eveR@gmail.com', 25);
```

The first query failed due to duplicate studentID
The second query failed due to a duplicate email.
The Third query failed due to a NULL value in the Name column.

Conclusion: By executing these SQL commands, we successfully applied constraints such as primary key, Unique key & NOT NULL to ensure data integrity. The experiment demonstrated how constraints prevent invalid data entry & enforce rules on DB tables effectively.

```
CREATE DATABASE Rushi;

USE Rushi;

CREATE TABLE Daulatkar (
    id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50),
    age INT,
    email VARCHAR(100) UNIQUE
);

INSERT INTO Daulatkar (first_name, last_name, age, email)
VALUES
    ('sumit', 'helonde', 19, 'sumithelonde@gmail.com'),
    ('shravan', 'sapate', 19, 'shravansapate@gmail.com'),
    ('piyush', 'kalambe', 18, 'piyushkalambe@gmail.com'),
    ('harshit', 'choudhary', 20, 'harshitchoudhary@gmail.com'),
    ('rushi', 'daulatkar', 16, 'rushidaulatkar@gmail.com');

SELECT * FROM Daulatkar;
```

Output:

id	first_name	last_name	age	email
1	sumit	helonde	19	sumithelonde@gmail.com
2	shravan	sapate	19	shravansapate@gmail.com
3	piyush	kalambe	18	piyushkalambe@gmail.com
4	harshit	choudhary	20	harshitchoudhary@gmail.com
5	rushi	daulatkar	16	rushidaulatkar@gmail.com

Rushi

Scanned with OKEN Scanner

Conclusion: By executing these SQL commands, we success fully applied constraints such as PRIMARY KEY, UNIQUE KEY & NOT NULL to ensure data integrity. The experiment demonstrated how the constraints ~~provide~~ prevent invalid data entry & enforce rules on database tables effectively.

(A+) Rushikesh
25/3/2025