**Objectives:**

The objectives of these SQL constraint lab questions are to provide practical experience in defining and managing database constraints using MySQL Workbench. Students will begin by creating a new database and constructing two tables, student and teacher, with specified columns. They will apply the NOT NULL constraint to ensure mandatory fields and use the UNIQUE constraint to prevent duplicate entries in the id columns. The tasks include setting primary keys to uniquely identify each record and applying default values to automate data entry for specific columns. Additionally, students will learn to modify existing tables by adding and removing constraints, such as deleting a primary key and a NOT NULL constraint. These exercises aim to deepen students' understanding of how constraints enforce data integrity and consistency within a relational database.

**Lab 2: Constraints in SQL**

**Theory:**

In SQL, constraints are used to specify rules for the data in a table. Constraints ensure the accuracy and reliability of the data within the database. There are several types of constraints, each serving a different purpose. Here are the most commonly used constraints in SQL:

1. **NOT NULL Constraint**: Ensures that a column cannot have a NULL value.
2. **UNIQUE Constraint**: Ensures that all the values in a column are different.
3. **PRIMARY KEY Constraint**: Uniquely identifies each record in a table. It must contain unique values and cannot contain NULL values.
4. **FOREIGN KEY Constraint**: Uniquely identifies a record in another table, ensuring referential integrity.
5. **CHECK Constraint**: Ensures that the values in a column satisfy a specific condition.
6. **DEFAULT Constraint**: Provides a default value for a column when none is specified.

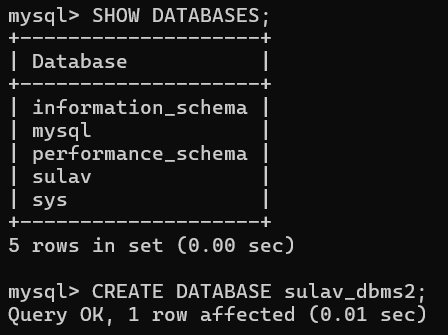
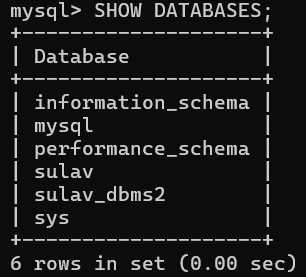
**Questions:**

1. **Create a database <yourname>dbms2:**

**Query:**

CREATE DATABASE sulav\_dbms2;

**Output:**



1. **Create tables**
   1. **student** (id(int), name, gender, program, contact\_number, address)

**Query:**

USE sulav\_dbms2;

CREATE TABLE sulav\_student (

id INT,

name VARCHAR(100),

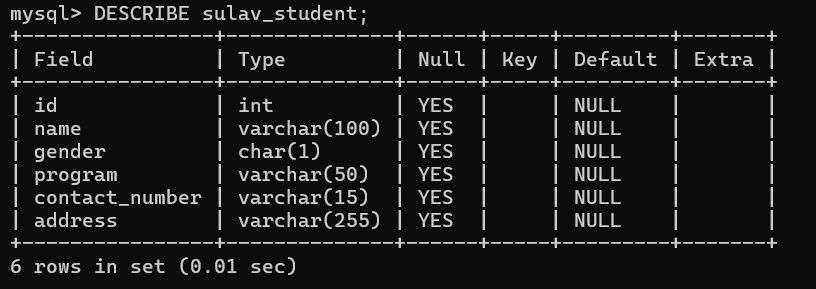
gender CHAR(1),

program VARCHAR(50),

contact\_number VARCHAR(15),

address VARCHAR(255));

**Output:**

****

* 1. **teacher** (id(int), name, subject, contact\_number, address)

**Query:**

USE sulav\_dbms2;

CREATE TABLE sulav\_teacher (

id INT,

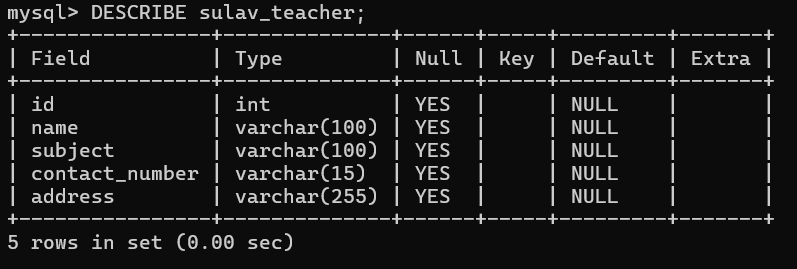
name VARCHAR(100),

subject VARCHAR(100),

contact\_number VARCHAR(15),

address VARCHAR(255));

**Output:**

****

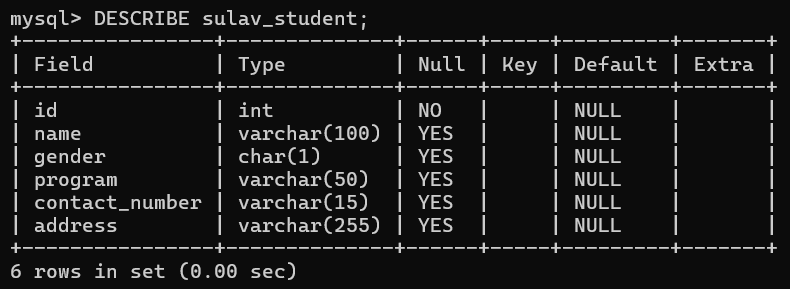
* 1. **Set NOT NULL constraint on the "id" column for both tables**

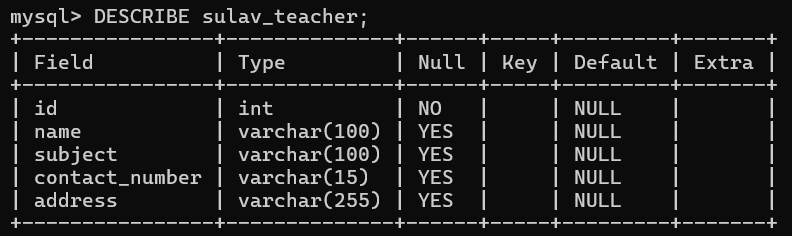
**Query:**

ALTER TABLE sulav\_student MODIFY id INT NOT NULL;

ALTER TABLE sulav\_teacher MODIFY id INT NOT NULL;

**Output:**

****

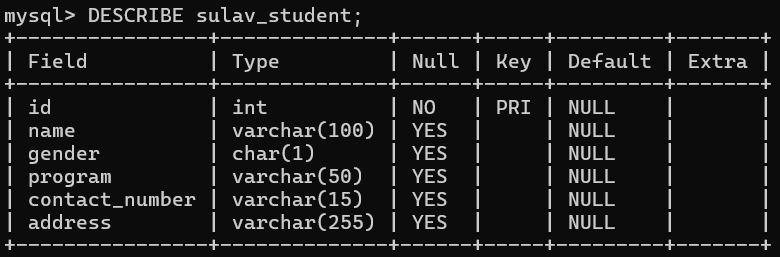
****

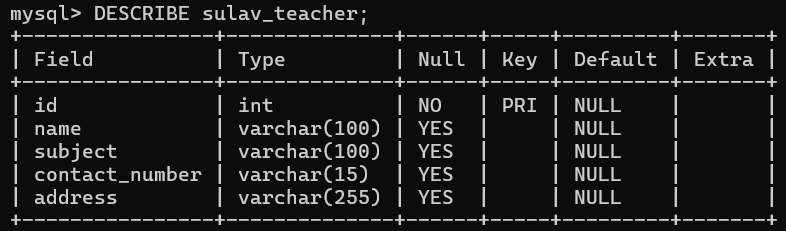
* 1. **Set UNIQUE constraint on the "id" column**

**Query:**

ALTER TABLE sulav\_student ADD CONSTRAINT unique\_student\_id UNIQUE (id);

ALTER TABLE sulav\_teacher ADD CONSTRAINT unique\_teacher\_id UNIQUE (id);

**Output:**



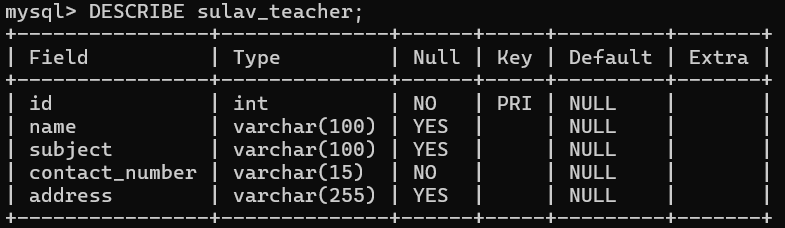
1. **Set NOT NULL constraint on the contactno column in the existing teacher table.**

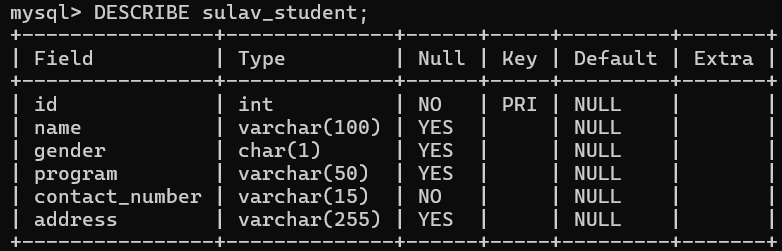
**Query:**

ALTER TABLE sulav\_teacher MODIFY contact\_number VARCHAR(15) NOT NULL;

ALTER TABLE sulav\_student MODIFY contact\_number VARCHAR(15) NOT NULL;

**Output:**

****

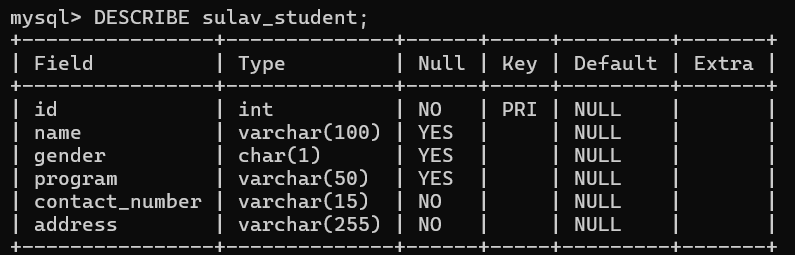
****

1. **Set NOT NULL constraint on the address column in the existing student table.**

**Query:**

ALTER TABLE sulav\_student MODIFY address VARCHAR(255) NOT NULL;

**Output:**

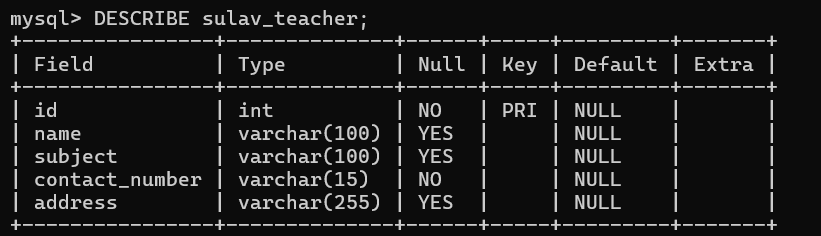
****

1. **Set PRIMARY KEY on the id column on the teacher table.**

**Query:**

ALTER TABLE sulav\_teacher ADD PRIMARY KEY (id);

**Output:**

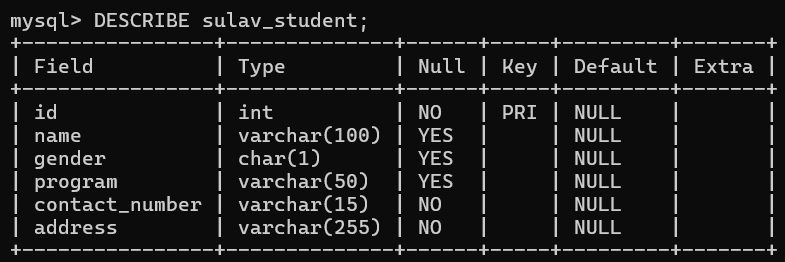
****

1. **Set PRIMARY KEY on the id column on the student table.**

**Query:**

ALTER TABLE sulav\_student ADD PRIMARY KEY (id);

**Output:**

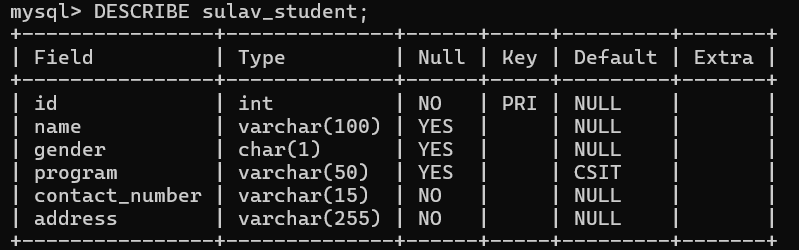
****

1. **Set DEFAULT VALUE of program as “CSIT” in student table.**

**Query:**

ALTER TABLE sulav\_student MODIFY program VARCHAR(50) DEFAULT 'CSIT';

**Output:**

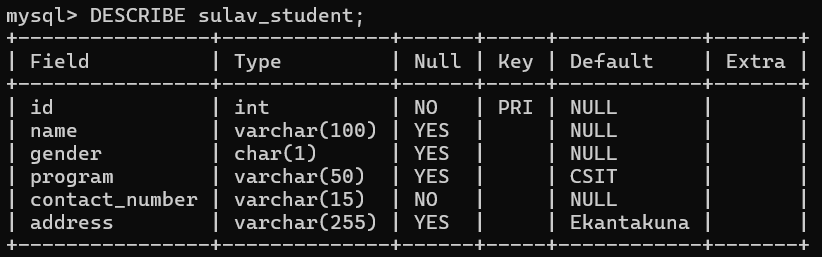
****

1. **Set DEFAULT VALUE of address as “Ekantakuna” in student table.**

**Query:**

ALTER TABLE sulav\_student MODIFY address VARCHAR(255) DEFAULT 'Ekantakuna';

**Output:**

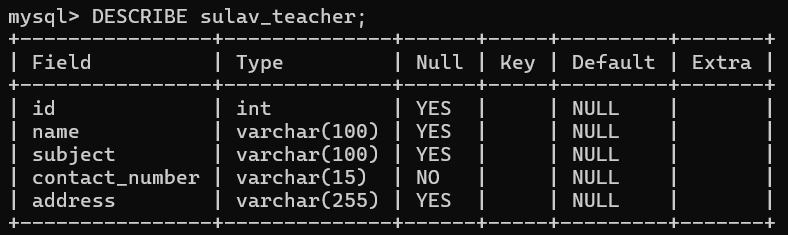
****

1. **Delete the constraint PRIMARY KEY on the id column on  teacher table.**

**Query:**

ALTER TABLE sulav\_teacher DROP INDEX unique\_teacher\_id;

**Output:**

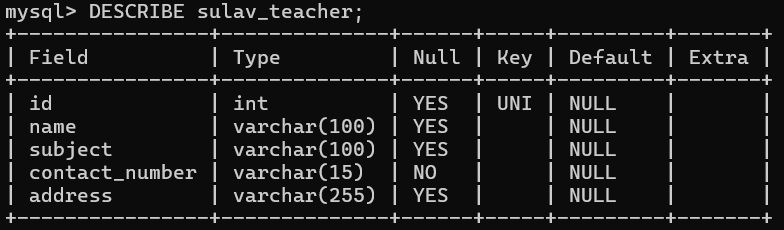
****

1. **Delete NOT NULL constraint on the id column for teacher table.**

**Query:**

ALTER TABLE sulav\_teacher MODIFY id INT;

**Output:**

****

**Conclusion:**

In conclusion, mastering the use of constraints in SQL is crucial for maintaining the integrity and reliability of a database. By working through these practical exercises, students gain valuable skills in setting up and managing constraints, which are essential for ensuring that the data adheres to the defined rules and business requirements. This hands-on experience is fundamental for anyone looking to become proficient in database management and design, as it highlights the importance of constraints in preserving data quality and consistency.