Department of
Computer Science and Engineering

Title: Implementation of Integrity Constraints in MySQL

Databse System Lab

CSE 210

Green University of Bangladesh

1 Objective(s)

- To Declare Primary Key
- To Create Composite Key
- To Implement Unique Constraint
- To Implement Foreign Key Constraint

2 Problem analysis

In the previous lab, we have already created database, tables and used them. In this lab, we have to declare primary key, create composite key and implement unique and foreign key constraint. For these purposes, we have to crate a database first. You can also use database that has already been created in the previous lab. Then, we have to create a table with a primary key. In the next, we have to create composite key and implement unique and foreign key constraint for a table. Finally, we have to insert tuples in the tables. Workflow of this lab is as in the figure 1.

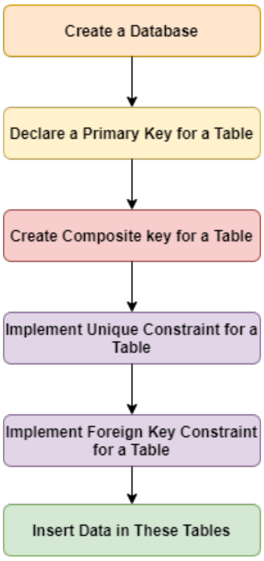

Create a Database
Declare a Primary Key for a Table
Create Composite key for a Table
Implement Unique Constraint for a Table
Implement Foreign Key Constraint for a Table
Insert Data in These Tables

Figure 1: Workflow Diagram of this Lab

3 Procedure

We have practiced with the XAMPP in the previous labs, we can assume that the system is ready to use. First, we have to launch the XAMPP. Then, we have to press the **Start** button of **Apache** and **MySQL** module. After that, we have to press the **Admin** button of the **MySQL** module. As a result, a tab will be opened on your default web browser like figure 2. Or, we can open a tab in the web browser with the link as http://localhost/phpmyadmin/. Then, we have to select the SQL option. An editor space will be opened like figure 3 to write the required commands. Now, it is ready for implementation.

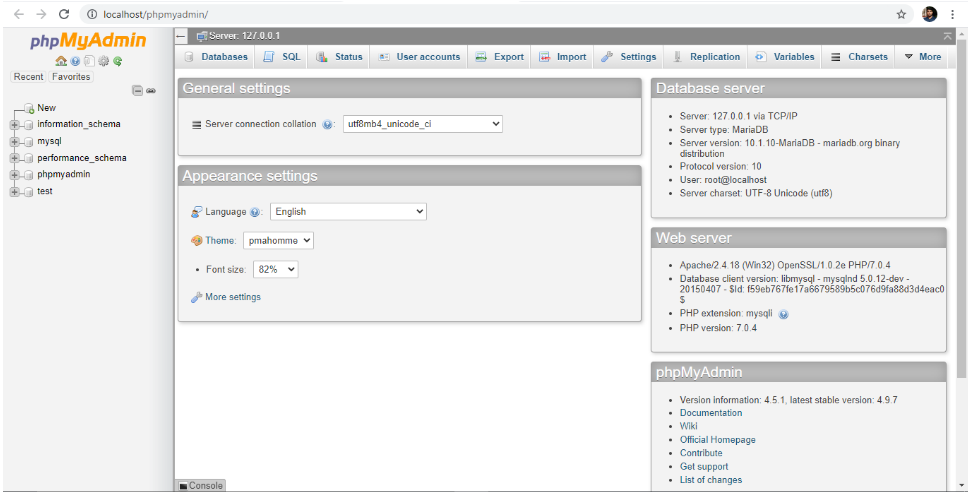
4 Implementations

4.1 Database Creation

To crate a database, we have to write command like " CREATE DATABASE [Database_Name]". Suppose, we have to create a database named "lab3". We have to write the command as below:

```
CREATE DATABASE lab3;
```

A database named "**lab3**" is created in your local-host.



General settings

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language: English

Theme: pmahomme

- Font size: 82%

More settings

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server version: 10.1.10-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

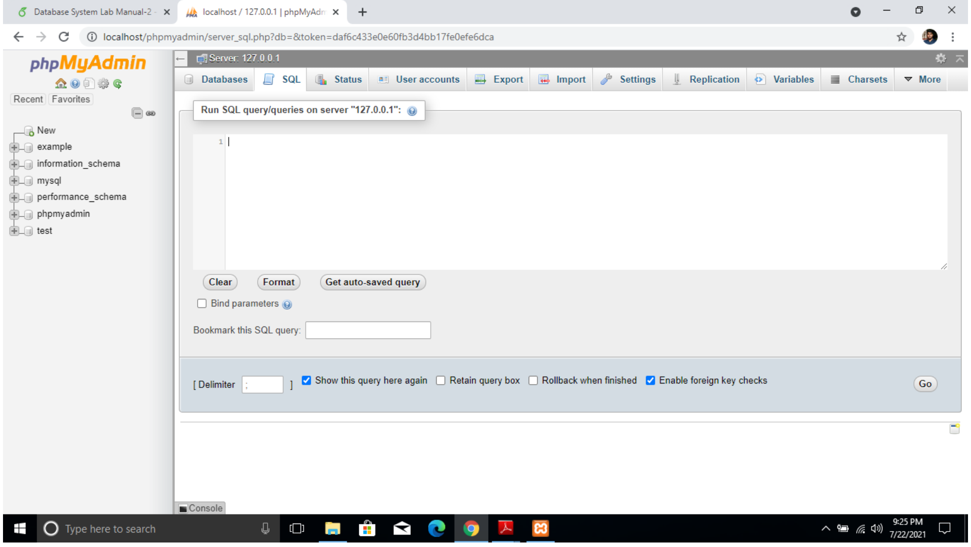
Web server

- Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/7.0.4
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - $Id: f59eb767fe17a6679589b5c076d9fa88d3d4eac0 $
- PHP extension: mysqli
- PHP version: 7.0.4

phpMyAdmin

- Version information: 4.5.1, latest stable version: 4.9.7
- Documentation
- Wiki
- Official Homepage
- Contribute
- Get support
- List of changes

Figure 2: Session in Localhost



Database System Lab Manual-2
localhost / 127.0.0.1 | phpMyAdm
localhost/phpmyadmin/server_sql.php?db=&token=daf6c433e0e60fb3d4bb17fe0efe6dca
phpMyAdmin
Recent Favorites
New
example
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1
Databases SQL Status User accounts Export Import Settings Replication Variables Charsets More
Clear
Format
Get auto-saved query
Go
Console
Type here to search
9:25 PM
7/22/2021

Run SQL query/queries on server "127.0.0.1":

□ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks

Figure 3: Space for Editing Commands

4.2 Database Use

To use the "lab3" database, we have to write the command in SQL editor space as bellow:

```
USE lab3
```

4.3 Declaration of Primary Key

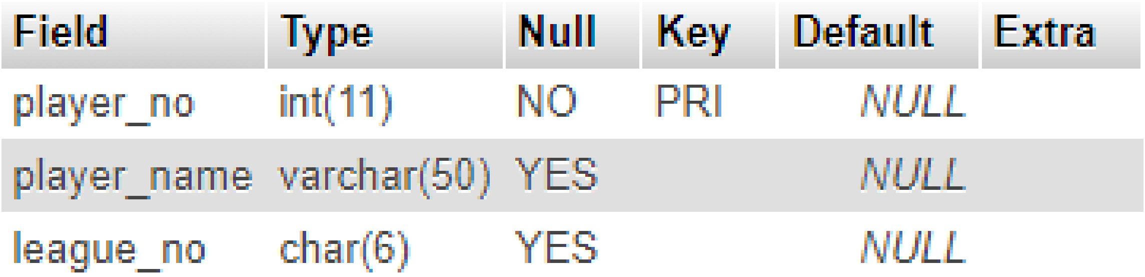
Now, to create a table named "**Players**" in database **lab3** with attributes like **player_no (int), player_name (varchar), league_no (char)** where **player_no** would be the **Primary key**, we have to write command in SQL editor space like below:

```
CREATE TABLE Players(player_no int PRIMARY KEY, player_name varchar(50), league_no
char(6));
```

To describe the structure of **Players** table, we have to write command as bellow:

```
DESCRIBE players;
```

We will get table like figure 4 on the browser tab.



Field	Type	Null	Key	Default	Extra
player_no	int(11)	NO	PRI	*NULL*	
player_name	varchar(50)	YES		*NULL*	
league_no	char(6)	YES		*NULL*	

Figure 4: Description of **Players** Table

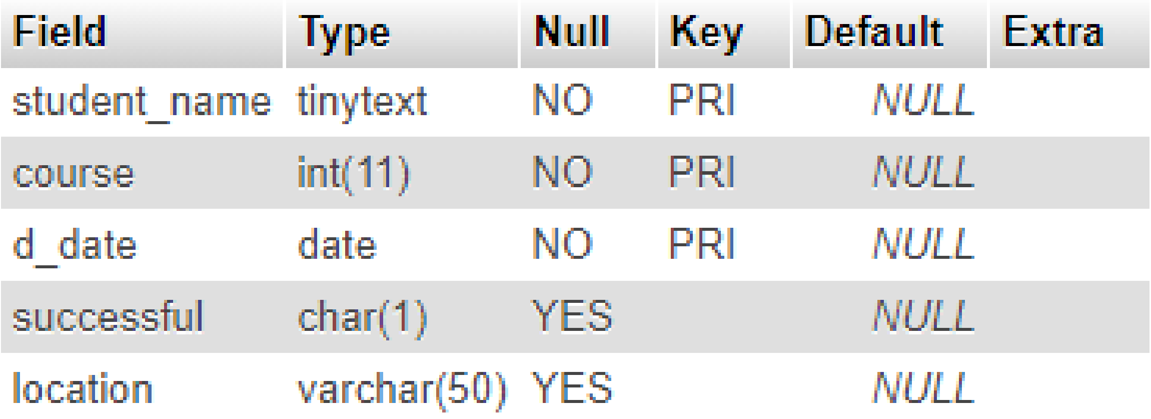
The **Student** table is ready to insert data.

4.4 Create Composite Key

Composite Key consists more than one keys which can uniquely identify each tuple in the table. To example it, we want to crate a table named **diplomas** with attributes named **student_name (text), course (int), d_date (date), successful (char), location (varchar)** where **student_name, course, d_date** consist composite key and they don't contain null value. The command is as bellow:

```
CREATE TABLE diplomas(student_name text(20) NOT NULL, course int NOT NULL, d_date DATE NOT NULL, successful char(1), location varchar(50), PRIMARY KEY(student_name(20),course,d_date))
```

Description of **diplomas** table is like the figure 5



Field	Type	Null	Key	Default	Extra
student_name	tinytext	NO	PRI	*NULL*	
course	int(11)	NO	PRI	*NULL*	
d_date	date	NO	PRI	*NULL*	
successful	char(1)	YES		*NULL*	
location	varchar(50)	YES		*NULL*	

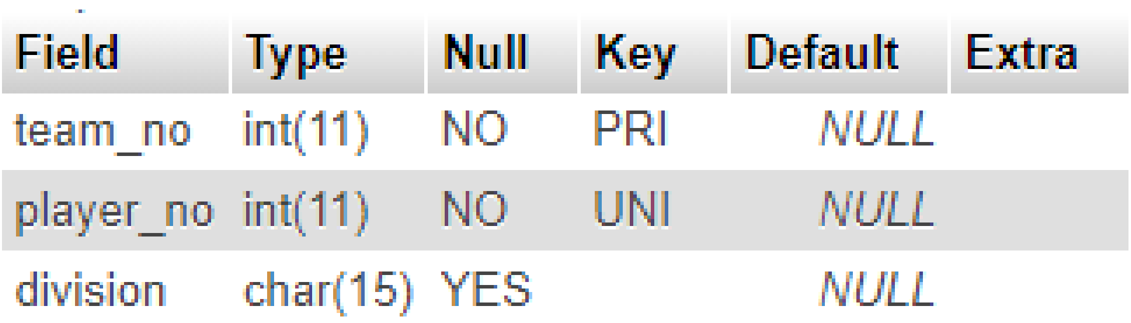
Figure 5: Description of **diplomas** Table

4.5 Implementation of Unique

We want to create a table named **teams** with attributes named **team_no (int), player_no (int), division (char)** where team_no is primary key and player_no is unique key. The command is as bellow:

```
CREATE TABLE teams(team_no int NOT NULL, player_no int NOT NULL, division char(15), PRIMARY KEY(team_no), UNIQUE(player_no));
```

Description of **teams** table is like figure 6



Field	Type	Null	Key	Default	Extra
team_no	int(11)	NO	PRI	NULL	
player_no	int(11)	NO	UNI	NULL	
division	char(15)	YES		NULL	

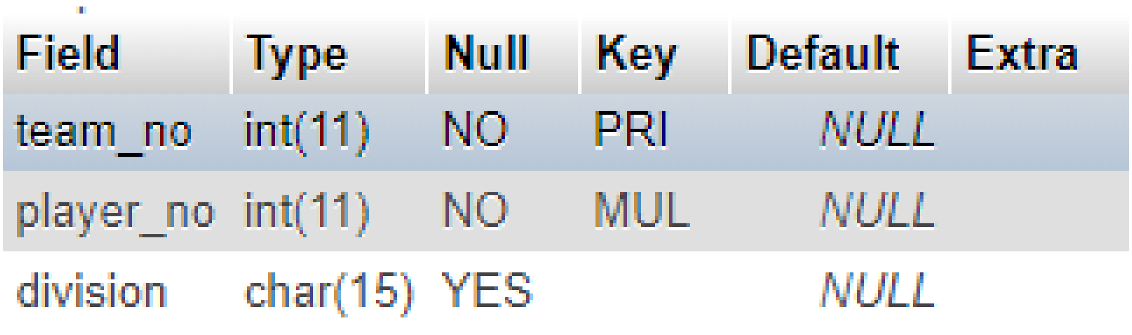
Figure 6: Description of **teams** Table

4.6 Implementation of Foreign Key

Now, We want to create a table named **teams2** with attributes named **team_no (int), player_no (int), division (char)** where team_no is primary key and player_no is a foreign key referenced from **Players** table. The command is as bellow:

```
CREATE TABLE teams2(team_no int NOT NULL, player_no int NOT NULL, division char(15),
PRIMARY KEY(team_no), FOREIGN KEY (player_no) REFERENCES players(player_no))
```

Description of **teams2** table is like figure 7



Field	Type	Null	Key	Default	Extra
team_no	int(11)	NO	PRI	NULL	
player_no	int(11)	NO	MUL	NULL	
division	char(15)	YES		NULL	

Figure 7: Description of **teams2** Table

4.7 Data Insertion

To insert data in the table, we have to write proper SQL command on the SQL editor space. Suppose, we want to insert a student details in **players** table like **player_no: 75,player_name: Sakib Al Hasan, league_no: A2**, we have to write command as bellow:

```
INSERT INTO ‘players‘ (‘player_no‘, ‘player_name‘, ‘league_no‘) VALUES (’75’, ’Sakib Al Hasan’,
’A2’);
```

While inserting tuple, we must take care of **NOT NULL** field. If we don’t input in this filed, we will get **warning message**.

4.8 MySQL CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column. If you define a CHECK constraint on a column it will allow only certain values for this column. If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

- **Create a table Person:**

```
CREATE TABLE  Person(
ID int(11) NOT NULL AUTO_INCREMENT,
First_Name varchar(255) NOT NULL,
Last_Name varchar(255) ,
Address varchar(255) NOT NULL,
Age INT NOT NULL,
CHECK(Age>=18),
PRIMARY KEY(ID)
);
```

- To allow naming of a CHECK constraint, and for defining a CHECK constraint on multiple columns, use the following SQL syntax:

```
CREATE TABLE  Person(
ID int(11) NOT NULL AUTO_INCREMENT,
First_Name varchar(255) NOT NULL,
Last_Name varchar(255) ,
Address varchar(255) NOT NULL,
Age INT NOT NULL,
Salary INT NOT NULL,
CHECK(Age>=18 AND Salary>=20000)  ,
PRIMARY KEY(ID)
);
```

4.9 MySQL CASE Examples

The CASE statement goes through conditions and returns a value when the first condition is met (like an if-then-else statement). So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause.

```
SELECT ID, Last_Name,
CASE WHEN Age > 18 THEN ’He or She is eligible for voting’
WHEN Age = 18 THEN ’He or She has applied for NID’
ELSE ’He or She is not eligible for voting’
END AS Feedback
FROM Person;
```

If there is no ELSE part and no conditions are true, it returns NULL.

5 Discussion & Conclusion

In this lab, we have created a database with four tables. We have created primary key in different tables, created composite key for **diplomas** table, implemented unique for **teams** table and foreign key for **teams** table. Finally, we have inserted data for **Players** table. That’s meant, we have achieved our lab objectives.

6 Lab Task (Please implement yourself and show the output to the instructor)

1. Insert at least five tuples in each table.
2. Try to violate the **NOT NULL** constraint
3. Try to violate the **Unique** constraint

6.1 Problem analysis

1. You have to insert at least five tuples for each table created in this lab.
2. You have to input NULL values for the NOT NULL field and try to explain the warning.
3. You have to input duplicate values for the Unique field and try to explain the warning.

7 Lab Exercise (Submit as a report)

- Create a Database with three tables.
- Assign primary key for each table.
- Assign a unique in at least two tables.
- Implement foreign key and CHECK constraint in at least one table.
- Insert Data in each table.
- Browse data for each table.

8 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.