Laboratory 4

Title of the Laboratory Exercise: implement constraints and built in functions

1. Introduction and Purpose of Experiment

Constraints are the rules which are enforced on the data being stored in a table. There are constraints that can be applied to a table such as NOT NULL, UNIQUE, PRIMARY KEY and FOREIGN KEY. SQL has many built-in functions for performing calculations on data. In SQL, a built-in function is a piece for programming that takes zero or more inputs and returns a value. By doing this lab, students will be able to implement constraints and built in functions on the database.

2. Aim and Objectives

Aim

- To design and implement constraints on the data using SQL commands
- To implement built in functions in SQL

Objectives

At the end of this lab, the student will be able to

- Identify different types of constraints on the data
- Apply constraints on the data in different ways
- Implement built-in functions in SQL

3. Experimental Procedure

i. Analyse the problem statement
ii. Design SQL commands using appropriate constraints
iii. Execute the SQL commands
iv. Test the executed commands
v. Document the Results
vi. Analyse and discuss the outcomes of your experiment

4. Questions

Consider the relational schema given below.

PLAYER_DETAILS (PCode, PName, DOB, City, Score)

MATCH_DETAILS (MatchID, MatchName, PCode)

a. **Apply the following constraints on the given database schema. Enter appropriate tuples to show the purpose of each constraint.**
   i. Not NULL
   ii. Default
   iii. Unique
   iv. Primary key
   v. Foreign key

b. Execute the following built-in functions in SQL using Netbeans IDE
   i. String functions
   ii. Date functions
   iii. Numeric functions

5. Presentation of Results

```
mysql> create table PLAYER_DETAILS(PCode int primary key,PName char(20) not null,DOB date,City char(20) default 'Bengaluru',Score int);
Query OK, 0 rows affected (0.01 sec)

mysql> Desc PLAYER_DETAILS;
+-------+----------+------+-----+-----------+-------+
| Field | Type     | Null | Key | Default   | Extra |
+-------+----------+------+-----+-----------+-------+
| PCode | int(11)  | NO   | PRI | NULL      |       |
| PName | char(20) | NO   |     | NULL      |       |
| DOB   | date     | YES  |     | NULL      |       |
| City  | char(20) | YES  |     | Bengaluru |       |
| Score | int(11)  | YES  |     | NULL      |       |
+-------+----------+------+-----+-----------+-------+
5 rows in set (0.01 sec)
```

Figure4.5.1 Created table PLAYER_DETAILS

Created Relational Schema Named **PLAYER_DETAILS** with PCode, PName, DOB, City, Score as attributes

```
mysql> create table MATCH_DETAILS(MatchID int(6) unique,MatchName char(20),Pcode int(6),foreign key(Pcode) references PLAYER_DETAILS(PCode));
Query OK, 0 rows affected (0.08 sec)

mysql> Desc MATCH_DETAILS;
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| MatchID   | int(6)   | YES  | UNI | NULL    |       |
| MatchName | char(20) | YES  |     | NULL    |       |
| Pcode     | int(6)   | YES  | MUL | NULL    |       |
+-----------+----------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

Figure 4.5.2 Created table MATCH_DETAILs

Created Relational Schema Named **Match_Details** with MatchID, MatchName, PCode as Attributes

```
mysql> insert into PLAYER_DETAILS VALUES(101,'DEEPAK R','2000-10-12','Bengaluru',5);
Query OK, 1 row affected (0.01 sec)

mysql> insert into PLAYER_DETAILS VALUES(101,NULL,'1999-1-1','Bengaluru',5);
ERROR 1048 (23000): Column 'PName' cannot be null
```

Figure 4.5.3 NOT NULL Constraint

NOT NULL constraint enforces a column to **NOT** accept **NULL** values. This enforces a field to always contain a value, which means that we cannot insert a new record, or update a record without adding a value to this field.

```
mysql> insert into PLAYER_DETAILS VALUES(102,'Virat','2012-12-18','Delhi',6);
Query OK, 1 row affected (0.01 sec)

mysql> insert into PLAYER_DETAILS (PCode,PName,DOB,Score) Values(103,'Warner','2020-12-1',8);
Query OK, 1 row affected (0.03 sec)

mysql> select * from PLAYER_DETAILS;
+-------+----------+------------+-----------+-------+
| PCode | PName    | DOB        | City      | Score |
+-------+----------+------------+-----------+-------+
|   101 | DEEPAK R | 2000-10-12 | Bengaluru |     5 |
|   102 | Virat    | 2012-12-18 | Delhi     |     6 |
|   103 | Warner   | 2020-12-01 | Bengaluru |     8 |
+-------+----------+------------+-----------+-------+
3 rows in set (0.00 sec)
```

Figure 4.5.4 DEFAULT Constraint

DEFAULT constraint is used to provide a **default** value for a column. The **default** value will be added to all new records IF no other value is specified .

```
mysql> insert into MATCH_DETAILS VALUES(201,'CRICKET',101);
Query OK, 1 row affected (0.01 sec)

mysql> insert into MATCH_DETAILS VALUES(201,'FOOTBALL',102);
ERROR 1062 (23000): Duplicate entry '201' for key 'MatchID'
```

Figure 4.5.5 Unique Constraint

The **UNIQUE constraint** ensures that all values in a column are different if not it it displays error.

```
mysql> insert into Player_Details Value(104,'Dhawan','2020/4/1','Kolkatta',8);
Query OK, 1 row affected (0.00 sec)

mysql> insert into Player_Details Value(104,'Pandya','2000/8/4','Bengaluru',9);
ERROR 1062 (23000): Duplicate entry '104' for key 'PRIMARY'
```

Figure 4.5.6 Primary Key Constraint

The **PRIMARY KEY constraint** uniquely identifies each record in a table. **Primary keys** must contain UNIQUE values, and cannot contain NULL values. If its NULL it displays error

```
mysql> select * from player_details;
+-------+----------+------------+-----------+-------+
| PCode | PName    | DOB        | City      | Score |
+-------+----------+------------+-----------+-------+
|   101 | DEEPAK R | 2000-10-12 | Bengaluru |     5 |
|   102 | Virat    | 2012-12-18 | Delhi     |     6 |
|   103 | Warner   | 2020-12-01 | Bengaluru |     8 |
|   104 | Dhawan   | 2020-04-01 | Kolkatta  |     8 |
+-------+----------+------------+-----------+-------+
4 rows in set (0.00 sec)

mysql> select * from match_details;
+---------+-----------+-------+
| MatchID | MatchName | Pcode |
+---------+-----------+-------+
|     201 | CRICKET   |   101 |
+---------+-----------+-------+
1 row in set (0.00 sec)
```

```
mysql> insert into Match_details Values(203,'Football',109);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`student_details`.`match_details`,
 CONSTRAINT `match_details_ibfk_1` FOREIGN KEY (`Pcode`) REFERENCES `player_details` (`PCode`))
```

Figure 4.5.6 Foreign Key Constraint

The **foreign key constraint** ensures referential integrity. It means that we can only insert a row into the child table if there is a corresponding row in the parent table.if we try to add Value which is not in PCode column of Player_details in Match_details it displays error.

6. Analysis and Discussions

Relational Integrity constraints is referred to conditions which must be present for a valid relation. These integrity constraints are derived from the rules in the mini-world that the database represents. Constraints are restrictions on the actual values in a database state

There are various restrictions on data that can be specified on a relational database in the form of constraints.

Constraints that can be directly expressed in schemas of the data model, typically by specifying them in the DDL are called schema-based constraints.

‘

7. Conclusions

- While creating tables, by default the rows can have null value. The enforcement of not null constraint in a table ensures that the table contains values.
- The DEFAULT constraint inserts a default value into a column of a table. When you insert a new row into the table, no need to specify the value for the column.
- UNIQUE is used to ensure that information in the column for each record is unique, as with telephone or driver's license numbers. It prevents the duplication of value with rows of a specified column in a set of columns. A column defined with the constraint can allow null value.
- A primary key avoids duplication of rows and does not allow null values. It can be defined on one or more columns in a table and is used to uniquely identify each row in a table. These values should never be null. Better to use only one primary key at the table level.
- Referential integrity constraint enforces relationship between tables. Foreign key is a column or combination of column included in the definition of referential integrity, which would refer to a referenced key. They create a parent child relationship between two tables. Referenced key is a unique or primary key upon which is defined on a column belonging to the parent table.

8. Comments

1. Limitations of Experiments

We cannot use the User defined functions, shortly called as UDF in SQL Server to modify the database state.

SQL UDF can not return multiple result sets.

2. Limitations of Results

The SQL UDF does not support error handling, such as TRY..CATCH, RAISEERROR, or @ERROR.

We cannot call a stored Procedure from SQL UDF, but we can call extended Stored Procedure.

3. Learning happened

Learnt the different types of schema-based constraints.

Learnt the syntax and purpose of basic inbuilt functions in SQL.

4. Recommendations

Use commands related to MySQL and not Oracle SQL, they are quite different and have a different implementation of SQL