Indian Institute of Information Technology Sri City
Database Management Systems LAB-03
Date: 27 Aug 2020
TOPIC : MySQL CONSTRAINTS and OPERATORS
Instructors: Dr. Odelu Vanga, Sarvani CH, Rangachary K

MySQL CONSTRAINTS and OPERATORS

MySQL CONSTRAINT

MySQL CONSTRAINT is used to define rules to allow or restrict what values can be stored in columns. The purpose of inducing constraints is to enforce the integrity of a database.

CONSTRAINT	**DESCRIPTION**
NOT NULL	In MySQL NOT NULL constraint allows to specify that a column can not contain any NULL value. MySQL NOT NULL can be used to CREATE and ALTER a table.
UNIQUE	The UNIQUE constraint in MySQL does not allow to insert a duplicate value in a column. The UNIQUE constraint maintains the uniqueness of a column in a table. More than one UNIQUE column can be used in a table.
PRIMARY KEY	A PRIMARY KEY constraint for a table enforces the table to accept unique data for a specific column and this constraint creates a unique index for accessing the table faster.
FOREIGN KEY	A FOREIGN KEY in MySQL creates a link between two tables by one specific column of both tables. The specified column in one table must be a PRIMARY KEY and referred by the column of another table known as FOREIGN KEY.
CHECK	A CHECK constraint controls the values in the associated column. The CHECK constraint determines whether the value is valid or not from a logical expression.
DEFAULT	In a MySQL table, each column must contain a value (including a NULL). While inserting data into a table, if no value is supplied to a column, then the column gets the value set as DEFAULT.

Examples:

NOT NULL:

```
 CREATE TABLE IF NOT EXISTS newauthor
(aut_id int NOT NULL,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL );
```

CHECK:

```
 CREATE TABLE IF NOT EXISTS newauthor
(aut_id int NOT NULL, CHECK (aut_id >= 100),
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL );
```

UNIQUE:

```
 CREATE TABLE IF NOT EXISTS newauthor
(aut_id int NOT NULL, CHECK (aut_id >= 100),
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL,
UNIQUE (aut_id));
```

DEFAULT CONSTRAINT:

```
 CREATE TABLE IF NOT EXISTS newauthor
(aut_id int NOT NULL CHECK (aut_id >= 1000) DEFAULT 1200,
aut_name varchar(50) NOT NULL DEFAULT 'Anoos',
country varchar(25) NOT NULL DEFAULT 'USA',
```

home_city varchar(25) NOT NULL **DEFAULT 'Boston'**,

UNIQUE (aut_id));

OPERATORS:

Arithmetic: +, -, %, * and /

Comparison: <, >, =,!=, <=, >=, !< , !> and <>

Logical operators:

Operator	Description
BETWEEN	It is used to search within a set of values, by the minimum value and maximum value provided.
EXISTS	It is used to search for the presence of a row in a table which satisfies a certain condition specified in the query.
OR	It is used to combine multiple conditions in a statement by using the WHERE clause.
AND	It allows the existence of multiple conditions in an SQL statement's WHERE clause.
NOT	It reverses the meaning of the logical operator with which it is used. (Examples: NOT EXISTS, NOT BETWEEN, NOT IN, etc.)
IN	It is used to compare a value in a list of literal values.
ALL	It compares a value to all values in another set of values.
ANY	It compares a value to any value in the list according to the condition specified.
LIKE	It uses wildcard operators to compare a value to similar values.
IS NULL	It compares a value with a NULL value.
UNIQUE	It searches for every row of a specified table for uniqueness (no duplicates).

PRACTICE PROBLEMS:

```
create table customers(CustomerID Int,
Dateofbirth date,
Gender varchar(5),
CustomerName varchar(30),
City varchar(20),
country varchar(10));
```

Create table “Products” with the following fields;

```
create table products(CustomerID Int,
ProductID int,
ProductName varchar(20),
Availability varchar(20),
Price int);
```

Answer the following questionnaires;

1. Alter table and set CustomerID in Customers table as primary key
2. Set ProductID in the products table as Foreignkey which references CustomerID in Customers table.
3. Set a default value for Availability as “YES;
4. Selects all customers with a CustomerName that have "or" in any position
5. Insert CustomerID =2505 in Products table (Assume CustomerID=2505 is not there in Customers table.
6. Select all fields from "Customers" where country is "Germany" and city is "Berlin".
7. Selects all fields from "Customers" where the country is not "Germany".
8. Select all records where the value of the City column starts with the letter "a".
9. Select all records where the value of the City column starts with letter "a" and ends with the letter "b".
10. Select all records where the value of the City column does NOT start with the letter "a".
11. Select all records where the first letter of the City is an "a" or a "c" or an "s".
12. Select all records where the first letter of the City starts with anything from an "a" to an "f".
13. Select all the records where the Country is NOT "Norway" and NOT "France".
14. Select all the records where the value of the ProductName column is alphabetically between 'Hat' and 'Pen'.
15. List the customer names if it finds Any records in the products table that price = 100:
16. List the customer names if it finds all records in the products table that price = 1000:
17. Drop primary and foreign key constraint from both the tables.
18. Delete the record of all customers who are from country Brazil and city Victoria.

REFERENCES:

1. Silberschatz, H. Korth & S. Sudarshan, Database System Concepts, McGraw-Hill Education, 6th Edition, 2010.
2. 2.R. Elmasri & S.B. Navathe, Fundamentals of Database Systems, Pearson Education, 6th edition, 2010.
3. https://www.w3schools.com/sql/default.asp