Indian Institute of Information Technology Sri City
Database Management Systems LAB-03
TOPIC : MySQL CONSTRAINTS and OPERATORS
Instructors: Dr. Odelu V, Dr. Rakesh Kumar S, Dr. Annushree B

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	● No null value.
UNIQUE	● No duplicate value
PRIMARY KEY	● No duplicate value ● No null value
FOREIGN KEY	● Refers to primary key of other tables, also can be same table (self referential)
CHECK	● Specifies a predicate that must satisfy by every tuple in relation e.g., `Age int, CHECK (Age>=18)`
DEFAULT	● Used to set a default value

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,
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 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. SELECT * FROM Products WHERE Price BETWEEN 10 AND 20;

EXISTS	It is used to search for the presence of a row in a table which satisfies a certain condition specified in the query.
	SELECT customername
	FROM customers
	WHERE EXISTS (SELECT ProductName FROM Products WHERE Price < 20);

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.
	SELECT * FROM Customers
	WHERE CustomerName LIKE 'a%';

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 constraints 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.

Customers table data:

```
insert into customers values(1000,'1989-10-10','m','John','Delhi','India');
insert into customers values(1100,'1979-01-10','f','Kens','Berlin','Germany');
insert into customers values(2505,'1960-03-10','f','Kelvin','Hamburg','Germany');
insert into customers values(1205,'1998-11-10','m','Monk','Delhi','India');
insert into customers values(1507,'1992-08-10','f','mars','Oslo','norway ');
insert into customers values(2004,'1966-03-29','f','johny','paris','france');
insert into customers values(125,'1999-12-10','m','Maggie','Delhi','India');
insert into customers values(150,'1991-12-10','f','don','Oslo','norway ');
insert into customers values(200,'1968-11-29','f','lambie','paris','france');
insert into customers values(1105,'1989-10-10','m','manish','Delhi','India');
insert into customers values(1106,'1979-01-10','f','Kensive','Berlin','Germany');
insert into customers values(25,'1944-03-10','f','mans','Hamburg','Germany');
insert into customers values(12,'1996-11-10','m','Monk','Hyderabad','India');
insert into customers values(15,'1972-08-10','f','mars','goa','India ');
insert into customers values(20,'1966-03-29','f','johny','paris','france');
insert into customers values(1259,'1999-12-10','m','Maggie','Delhi','India');
```

Products table data:

```
insert into products values(1,1000,'pen','yes',100);
insert into products values(2,1100,'toy','no',10);
insert into products values(3,2505,'pencil','yes',30);
insert into products values(4,1205,'duck','yes',50);
insert into products values(5,1500,'pearl','no',60);
insert into products values(6,200,'hat','yes',10);
insert into products values(7,2505,'pot','no',90);
insert into products values(8,159,'book','yes',100);
insert into products values(9,1100,'chalk','no',10);
```