

Chapter 17:



Table & Integrity Constraints

Informatics Practices Class XII

By- Rajesh Kumar Mishra

PGT (Comp.Sc.)

KV No.1, AFS, Suratgarh

e-mail : rkmalld@gmail.com

Integrity Constraints

One of the major responsibility of a DBMS is to maintain the Integrity of the data i.e. Data being stored in the Database must be correct and valid.

An Integrity Constraints are the rules, condition or check applicable to a column or table which ensures the integrity or validity of data.

The following constraints are commonly used in MySQL.

- ❑ **NOT NULL**
- ❑ **PRIMARY KEY**
- ❑ **UNIQUE**
- ❑ **DEFAULT**
- ❑ **CHECK**
- ❑ **FOREIGN KEY**



Most of the constraints are applied with Column definition which are called **Column-Level (in-line Constraints)** ,but some of them may be applied at column Level as well as **Table-Level (Out-line constraints)** i.e. after defining all the columns e.g. Primary Key & Foreign Key

Type of Integrity Constraints

S.N	Constraints	Description
1	NOT NULL	Ensures that a column cannot have NULL value.
2	DEFAULT	Provides a default value for a column, when nothing is given.
3	UNIQUE	Ensures that all values in a column are different.
4	CHECK	Ensures that all values in a column satisfy certain condition.
5	PRIMARY KEY	Used to identify a row uniquely.
6	FOREIGN KEY	Used to ensure Referential Integrity of the data.

UNIQUE v/s PRIMARY KEY

- **UNIQUE** allows NULL values but **PRIMARY KEY** does not.
- Multiple column may have **UNIQUE** constraints, but there is only one **PRIMARY KEY** constraints.

How to Defining Table Constraints

Generally, Integrity Constraints are defined at the time of Table creation using **Create Table** command. Constraints may be added, modified later by using **Alter Table** command.

```
mysql> CREATE TABLE Student
      ( StCode  char(3)  NOT NULL PRIMARY KEY,
        Stname  char(20) NOT NULL,
        StAdd   varchar(40),
        AdmNo   char(5)  UNIQUE,
        StSex   char(1)  DEFAULT = 'M',
        StAge   integer  CHECK (StAge >= 10),
        Stream  char(1)  CHECK Stream IN ('S', 'C', 'A')
      );
```

Defining Primary Key


❖ Defining Primary Key at Column Level:

```
mysql> CREATE TABLE Student
      ( StCode char(3) NOT NULL PRIMARY KEY,
        Stname char(20) NOT NULL,
        .....
      );
```

❖ Defining Primary Key at Table Level:

```
mysql> CREATE TABLE Student
      ( StCode char(3) NOT NULL ,
        Stname char(20) NOT NULL,
        .....
        PRIMARY KEY (StCode)
      );
```

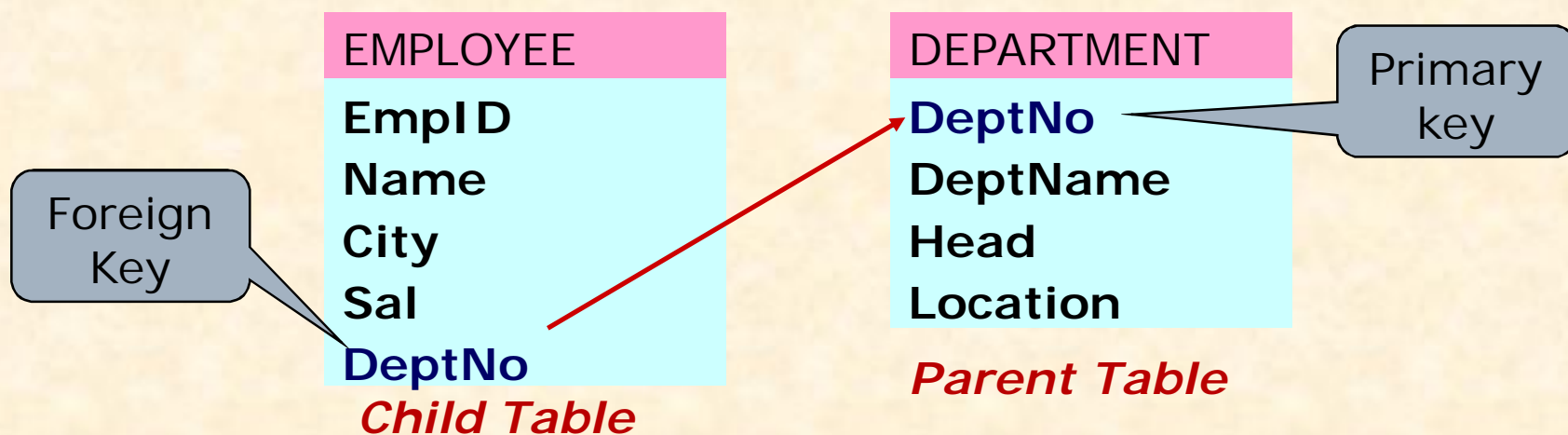
Constraint is defined as the last line

 A Composite (multi-column) Primary key can be defined as only a Table level whereas Single-column Primary key can be defined in both way i.e. Column level or Table level.

FOREIGN Key Constraints

□ FOREIGN KEY REFERENCES

- A Foreign key is non-key column in a table whose value is derived from Primary key of some other table.
- This constraints requires two tables in which Reference table (having Primary key) called **Parent table** and table having Foreign key is called **Child table**.
- Each time when record is inserted/updated the Child Table, the Parent Table is referred. This constraints is also called Referential Integrity Constraints.
- A child table may have multiple Foreign key constraints also.



Defining FOREIGN Key Constraints

```
CREATE TABLE Department
( DeptNo      char(2)  NOT NULL PRIMARY KEY,
  DeptName    char(10) NOT NULL,
  Head        char(30) );
```

Parent table should be created first.

```
CREATE TABLE Employee
( EmpNo char(3) NOT NULL PRIMARY KEY,
  Name  char(30) NOT NULL,
  City  char(20),
  Sal   decimal(8,2),
  DeptNo char(2),
  FOREIGN KEY (DeptNo) REFERENCES Department (DeptNo));
```

Foreign key as table level constraints... written after all columns.

Alternatively, Foreign Key can be applied as column-level, without using *Foreign Key* keyword.

➔ DeptNo char(2) REFERENCES Department (DeptNo),

Options with Foreign Key Constraints

What will happen, when a primary key value is deleted or updated in parent table, which is being referred in child table?

We can set the following options with FOREIGN KEY.

ON DELETE CASCADE / RESTRICT/ SET NULL/ NO ACTION

ON UPDATE CASCADE / RESTRICT/ SET NULL/ NO ACTION

CASCADE:

The values of Foreign key in child table, will be updated or referenced record will be deleted automatically.

RESTRICT:

This option will reject any Delete or Update operation on Primary key of Parent table.

SET NULL:

This option will set NULL value in referenced records of Foreign key in the child table, when Primary Key value is changed in Parent table.

NO ACTION:

No action will be taken when any Delete/Update operation is carried in Primary key of Parent table.

```
CREATE TABLE Employee
```

```
( .....
```

```
FOREIGN KEY DeptNo REFERENCE Department (DeptNo)
```

```
ON DELETE CASCADE ON UPDATE CASCADE );
```


Naming & Referring Self-Table with Foreign Key

❑ Naming Foreign Key Constraints

You can give your own name to Foreign Key Constraints at the time of creating table by placing CONSTRAINT keyword.

CONSTRAINT <Name> FOREIGN KEY

Ex:

Create Table EMP

(.....

CONSTRAINT Fkey1 FOREIGN KEY (DeptNo) REFERENCES DEPT(DeptID))

❑ Self-Referencing Table with Foreign Key Constraints

You can refer the any column of the same table in which Foreign Key is being defined. Also you can define multiple Foreign keys in a child table.

Ex:

Create Table EMP

(EmpID Integer **NOT NULL PRIMARY KEY,**

.....

CONSTRAINT Fkey2 FOREIGN KEY (MgrNo) REFERENCES EMP(EmpID)

CONSTRAINT Fkey1 FOREIGN KEY (DeptNo) REFERENCES DEPT(DeptNo))

The same table to be referred (Self-referencing)

Multiple-Foreign keys may be defined in a table.

Implementing Foreign Key in MySQL

In MySQL, Foreign Keys can be implemented only for the tables having **InnoDB** storage engine (storage mechanism). The MySQL uses default storage engine as **MyISAM**, which doesn't support Foreign key. You must change your storage engine of the table for which Foreign keys to be implemented, if InnoDB storage engine is not used.

❑ To check the existing engine:

The following command shows (at the last line of output) the engine on which EMP table is working.

```
mysql> Show Create Table Emp;
```

❑ To Change the Storage Engine:

The following command will change the storage engine of EMP table.

```
mysql> Alter Table EMP ENGINE=InnoDB;
```

❑ To Define Storage Engine with Create Table command:

The simplest way is to define the storage engine at the time of creating a table in which Foreign key to be implemented.

```
mysql> Create Table EMP
```

```
(.....
```

```
Foreign Key DeptNo References DEPT(DeptID)) ENGINE= INNODB;
```

Modifying Table Structure

You can alter (modify) the structure of existing table by the using **ALTER TABLE....** Command of MySQL.

You can do the following with the help of ALTER TABLE.. Command.

- **Add a new Column**
- **Modifying existing column (name, data type, size etc.)**
- **Delete an existing column**
- **Changing Column Name**
- **Adding / Removing Constraints**

ALTER TABLE <Table Name>

ADD|MODIFY|DROP|CHANGE <Column> [<Constraints>]

Modifying Table Structure

cont..

□ Adding new column

ALTER TABLE <Table Name>

ADD <Column><data type> <size> [<Constraints>]

```
mysql> ALTER TABLE Student ADD (TelNo Integer);
```

```
mysql> ALTER TABLE Student  
      ADD (Age Integer CHECK (Age >= 5) );
```

```
mysql> ALTER TABLE Emp  
      ADD (Sal Number(8,2) DEFAULT 5000 );
```

□ Modifying Existing Column

ALTER TABLE <Table Name>

MODIFY <Column><data type> <size> [<Constraints>]

```
mysql> ALTER TABLE Student  
      MODIFY (Name VARCHAR(40));
```

```
mysql> ALTER TABLE Emp MODIFY (Sal DEFAULT 4000 );
```

Modifying Table Structure

cont..

❑ Removing Column

ALTER TABLE <Table Name>

DROP [COLUMN] <Column name> <Constraints>

```
mysql> ALTER TABLE Student DROP TelNo;
```

```
mysql> ALTER TABLE Student DROP COLUMN Add;
```

❑ Changing Column Name of Existing Column

ALTER TABLE <Table Name>

CHANGE [COLUMN] <Old name> <New Name> [<datatype>]

```
mysql> ALTER TABLE Student  
CHANGE COLUMN Name StName;
```

```
mysql> ALTER TABLE Student  
CHANGE COLUMN Add Address Char(50);
```

Modifying Table Structure

cont..

□ Adding Constraints

```
ALTER TABLE <Table Name>  
ADD [CONSTRAINT <name>] [<Constraints>]
```

```
mysql> ALTER TABLE Student ADD PRIMARY KEY StCode;
```

```
mysql> ALTER TABLE Student  
ADD CONSTRAINT pk1 PRIMARY KEY (StCode) ;
```

```
mysql> ALTER TABLE Emp  
ADD FOREIGN KEY DeptNo REFERENCES DEPT(DeptID);
```

□ Removing Existing Constraints

```
ALTER TABLE <Table Name>  
DROP <Constraints>
```

```
mysql> ALTER TABLE Student DROP PRIMARY KEY;
```

```
mysql> ALTER TABLE EMP DROP FOREIGN KEY fkey1;
```

Viewing & Disabling Constraints

❑ To View the Constraints

The following command will show all the details like columns definitions, constraints and Storage engine of EMP table.

```
mysql> SHOW CREATE TABLE EMP;
```

❑ Enabling / Disabling Foreign Key Constraint

- ✓ You may enable or disable Foreign key constraints by setting the value of FOREIGN_KEY_CHECKS variable.
- ✓ You can't disable Primary key, however it can be dropped (deleted) by Alter Table... command.

▪ To Disabling Foreign Key Constraint

```
mysql> SET FOREIGN_KEY_CHECKS = 0;
```

▪ To Enable Foreign Key Constraint

```
mysql> SET FOREIGN_KEY_CHECKS = 1;
```

Dropping (Deleting) Table

❑ Deleting a Table:

You can delete a table including its records and structure by the following DDL command.

DROP TABLE [IF Exist] <Table Name>

```
mysql> DROP TABLE Emp;
```

```
mysql> DROP TABLE IF EXIST Emp;
```

❑ Deleting Records from a Table:

You can delete all or selected records from a table without deleting its structure by the following DDL command.

DELETE FROM <Table Name> [WHERE <Condition>

```
mysql> Delete From Emp ;
```

```
mysql> Delete From Emp Where EmpID='e5';
```

```
mysql> Delete From Emp Where Pay >= 10000;
```

❑ Deleting a Database:

You can delete the whole database including its all the table by the following DDL command.

DROP DATABASE [IF Exist] <Database Name>

```
mysql> DROP DATABASE School;
```
