**MySQL constraints**

♣ Applying data constraints- column level and table level

♣ Types of Data constraints- I/O constraints, Business rule constraints

• NOT NULL constraint, UNIQUE constraint, CHECK constraint

• Primary key and Foreign key constraint

• Disable foreign key checks

♣ Adding, Modify and drop constraints using alter table command

------------------------------------------------------------------------------------------------------------------------

* Constraints are the rules. These can be specified at the time of table creation or can be added after table creation by using alter table statement.
* Constraints prevent the table from deletion, if there is any dependency
* **Applying data constraints**

Constraints can be divided into following two types:

**1)** The constraints can be specified immediately after the column definition is called column-level definition.   
**2)** The constraints can be specified after all columns and end of table is called table-level definition.

**Data Integrity:**

* **Entity Integrity** ensures that no duplicate rows in a table. Ex: Unique, Primary Key
* **Domain Integrity** enforces valid entries for a given column by restricting the type, the format, or the range of possible values. Ex: check, Null, Not Null
* **Referential integrity** ensures that rows cannot be deleted, which are used by other records. Ex: Foreign Key
* **User-Defined Integrity** enforces some specific business rules on entity, domain, or referential integrity categories.

Constraints are:

* [**Not Null Constraint**](https://www.tutorialspoint.com/sql/sql-not-null.htm)**:** Ensures that a column have not NULL value.
* [**Default Constraint**](https://www.tutorialspoint.com/sql/sql-default.htm)**:** Provides a default value for a column when none is specified.
* [**Unique Constraint**](https://www.tutorialspoint.com/sql/sql-unique.htm)**:** Ensures that all values in a column are different (unique).
* [**Primary Key**](https://www.tutorialspoint.com/sql/sql-primary-key.htm)**:** Uniquely identified each rows/records in a database table.
* [**Foreign Key**](https://www.tutorialspoint.com/sql/sql-foreign-key.htm)**:** Uniquely identified a rows/records in any another database table.
* [**Check Constraint**](https://www.tutorialspoint.com/sql/sql-check.htm)**:** Ensures that all values in a column must satisfy certain conditions.
* **Not Null Constraint**
* NOT NULL constraint prevents inserting NULL values into a column. In database,

NULL means unknown or missing information.

* If you try to [insert](http://www.zentut.com/sql-tutorial/sql-insert/) or [update](http://www.zentut.com/sql-tutorial/sql-update/)a NULL value in the column, then database engine will

reject the change and issue an error.

* NOT NULL constraint applied only at column level not table level. You should manually define NOT NULL constraint because table column default set NULL value.
* When a column name is defined as not null, then that column becomes a mandatory column. It force to user for enter data into the column.

**Principles of NULL values**

* A NULL values is different from a blank or zero.
* A NULL value can be inserted into the columns of any Data type.

**Example:** To create an emp\_info table and enforces "Name" column to not null

create table emp\_info(no number(3) primary key,name char(10)not null,

address varchar2(30));

desc emp\_info;

Name Null? Type

---------------------------- --------- -----------------

NO NOT NULL NUMBER(3)

NAME NOT NULL CHAR(10)

ADDRESS VARCHAR2(30)

* **Add NOT NULL constraint in existing table column**

alter table emp\_info modify address varchar2(30) not null;

desc emp\_info;

Name Null? Type

---------------------------- --------- ---------------------

NO NOT NULL NUMBER(3)

NAME NOT NULL CHAR(10)

ADDRESS NOT NULL VARCHAR2(30)

* **Drop NOT NULL constraint from existing table column**

alter table emp\_info modify address varchar2(30) null;

desc emp\_info;

Name Null? Type

---------------------------- --------- ----------------------------

NO NOT NULL NUMBER(3)

NAME NOT NULL CHAR(10)

ADDRESS VARCHAR2(30)

* **Unique Key Constraint**
* The UNIQUE constraint is used for uniquely identify each record in a database. It

requires that each value in a column should be unique.

* It provides uniqueness guarantee of a column or set of columns. It cannot allow duplicate values but **unique key allow NULL (blank) values.**
* You can assign many unique key constraints in one table.
* UNIQUE constraint can be applied at column level and table level.

|  |  |
| --- | --- |
| **unique key at column level** | **Unique key at table level** |
| create table emp(eno number(5), name char(20),  **contact\_no number(10)unique,**  age number(2), salary number(10)); | create table emp(eno number(5),  name char(20), contact\_no number(10)  age number(2), salary number(10),  **unique (contact\_no)**); |

Now we are inserting record into this table to check unique key how to work

Example try to insert same name as soon as fire error for unique constraint violated.

sql> insert into emp(no,name) values(1,'om');

sql> insert into emp(no,name) values(2,'om');

\*ERROR line 1:ORA-00001:unique constraint (SYSTEM.SYS\_C004081) violated.

* **Defining a unique key constraint on multiple columns:**

**Example:** create table students (sid int not null, name varchar2(25) not null,

city varchar2(25), constraint uniqcon\_student unique (sid, name));

* **UNIQUE KEY constraint on ALTER TABLE:**
* Defining a unique key constraint on single column:

alter table students add unique (sid);

* Defining a unique key constraint on multiple columns:

alter table students add constraint uniqcon\_student unique (sid, name);

* **Primary Key Constraints:**
* Primary Key apply on columns for uniquely identifies each record from the table.
* Primary Key constraint column cannot accept duplicate data and NULL values.
* Each table can have only ONE primary key.
* A primary key is assigned one or more columns in a table is called **composite primary key.**
* It defines a mandatory column and data through the column must be unique.
* Primary key constraint create automatic index file for identifying unique records.
* Primary key constraint can be defined at the column and table level.
* Primary key constraint combination of NOT NULL and UNIQUE Constraints

When we specify a primary key constraint for column, database engine automatically creates a unique index file for identify unique records.

|  |  |
| --- | --- |
| **at table level (primary key for single columns)** | **At (single) column level** |
| create table stud(sid number(5),  sname char(20), age number(2),  contact\_no number(10),**primary key(sid)**); | create table stud(**sid number(5) primary key,**  sname char(20),  age number(2),contact\_no number(10)); |

* **Primary key on one column:**

alter table stud add primary key (sid);

* **Primary key on multiple columns:**

alter table stud add constraint pk\_studentid primary key (sid, contact\_no);

* **DROP a PRIMARY KEY constraint**

alter table stud drop primary key;

alter table stud drop constraint pk\_studentid;

* **Enable or Disable a Primary Key**

alter table students disable constraint pk\_studentid;

alter table students enable constraint pk\_studentid;

* **Foreign Key (referential) Constraint**
* Foreign key represents relationship between one or more tables. FOREIGN KEY apply on columns who references values in another table column.
* FOREIGN KEY constraints also known as relationship (referential) constraints.
* A foreign key is a column whose values are derived from the primary key of the other table.
* FOREIGN KEY constraint applied column must have same data type as they reference on another table column.

**Foreign key/references constraints:**

* Rejects an INSERT or UPDATE of a value, if a corresponding value does not exist in the primary key table.
* Rejects a DELETE, if it would invalidate a REFERENCES constraint.

**Syntax:** [CONSTRAINT constraint\_name] FOREIGN KEY(column\_name) REFERENCES

referenced\_table\_name(column\_name);

* **A FOREIGN KEY constraint specify at column level,**

create table emp\_info(no number(3) primary key,

name char(20), address varchar(30),

contact\_no number(12));

create table emp\_salary(no number(3) primary key,

users\_no number(3) references emp\_info(no),

salary number(12));

* **A FOREIGN KEY constraint specify at table level,**

create table emp\_info (no number(3) primary key,

name char(20),address varchar(30),

contact\_no number(12));

create table emp\_salary(no number(3) primary key,

user\_no number(3),salary number(12),

constraint fk\_user foreign key (user\_no) references emp\_info(no));

* **To add foreign key in existing table**

When you adding foreign key constraint, SQL check any existing data violate the foreign key constraint or not. If not violate constraint added successfully otherwise you have to update invalid data to prevent foreign key constraint violating.

alter table emp\_salary add foreign key (user\_no) references emp\_info(no);

* **To drop foreign key in existing table**

alter table emp\_salary drop constraint fk\_userno;

* **To drop a FOREIGN KEY constraint:**

alter table orders drop foreign key;

* **On Delete Cascade:** delete the dependent rows from the child table, only when parent table row is deleted.
* **On Delete Set Null:** when you want to convert dependent foreign key values to null.

|  |  |
| --- | --- |
| **Primary key** | **Foreign key** |
| Primary key cannot be null | Foreign key can be null. |
| Primary key is always unique. It uniquely identify a record in a table | Foreign key can be duplicated Because foreign key is a field refer primary key in another table. |
| only one primary key in the table | More than one foreign key in the table. |

* **Check Constraints:**
* Check constraint is used to check the value which is entered into a record. It is used to define condition which each row must satisfy.
* If the condition value evaluates to false, then the record violates the constraint and

you cannot enter it into the table.

* We cannot define check constraint in SQL view, sub queries or sequences.
* It performs check on the values, before storing them into the database. It’s like condition checking before saving data into a column.
* The constraint can be applied for a single column or a group of columns.

**Syntax:** [CONSTRAINT constraint name] CHECK (condition)

* **Define CHECK constraint at column level with other column attributes**

create table student\_info(stu\_code varchar(6) primary key check(stu\_code like 'st%'),

name varchar(30) check(name=upper(name)),

city varchar(30) check(city in('mumbai','pune','delhi','chennai')),

scholarship number(5) check (scholarship between 5000 and 20000),

class varchar(15));

insert into student\_info values ('st001','NAYANTARA','pune',8900,'ecs-II');

We are creating new **student\_info** table name with following check constraints:

1. Values inserted into **stu\_code** column must be start with the lower letter **'ST'**.
2. Values inserted into **name** column must be capitalized.
3. Values inserted into **city** column only allow 'Mumbai','Pune','Delhi','Chennai' as valid legal values.
4. Values inserted into scholarship column between 5000 and 20000.

* **CHECK constraint apply in table level.**

create table student\_info(stu\_code varchar(10) primary key,

name varchar(30),

city varchar (30),scholarship number(5), class varchar(10),

check (stu\_code like 'st%'),

check (name = upper(name)),

check (city in ('mumbai','pune','delhi','chennai')),

check (scholarship between 5000 and 20000));

insert into student\_info values ('st002','NAYAN','delhi',9000,'bca-II');

* **To add CHECK constraint in existing table column**

alter table student\_info add constraint check\_name check(class=upper(class));

* **To drop CHECK constraint in existing table column**

alter table student\_info drop constraint check\_name;

* **Default Constraint:**
* The DEFAULT constraint provides a default value to a column when the INSERT statement. You do not provide a specific value.
* The data type of default value should be match with the type of the column.
* DEFAULT constraint specified only at column level.
* **Specify DEFAULT constraint at column level with other column**

create table stud(roll\_no number(3) primary key,

stud\_name varchar(30) not null, class varchar(15) default 'M.Sc.(CS)',

fees\_pay number(5));

* **To add DEFAULT constraint in existing table column**

alter table stu\_info modify fees\_pay number(5) default 15000;

* **To drop DEFAULT constraint in existing table column**

You need to just redefine (or modify) column attribute.

alter table stu\_info modify fees\_pay number(5);