MySQL constraints

==========
NOT NULL
:
The NOT NULL constraint is a column constraint that ensures values stored in a column are not NULL.

A column may contain only one NOT NULL constraint which specifies a rule that the column must not contain any NULL value. In other words, if you update or insert NULL into a NOT NULL column, MySQL will issue an error.
eX:

```
create table temp(
                        name varchar(30) ,
                          id  int  not null ,
                          city varchar(35)
                );
```

```
create table emp(
        id int not null,
        name varchar(40) not null,
        sal double(16,4) not null
        );

create table xyz(
         name varchar(30) not null,
        id int not null
        );

create table emp1(
        id int
        name varchar(40) not null,
        sal double(16,4) not null
        );
```

```
CREATE TABLE tasks (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE


 alter table test modify id int not null;
 alter table emp modify id int not null;


mysql> desc employee;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| empid  | int         | YES  |     | NULL    |       |
| ename  | varchar(40) | YES  |     | NULL    |       |
| salary | double      | YES  |     | NULL    |       |
| city   | varchar(40) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+);

 insert into employee values(477, null ,2345,"pune");
ERROR 1048 (23000): Column 'ename' cannot be null

Add a NOT NULL constraint to an existing column
Check the current values of the column if there is any NULL.
Update the NULL  to non-NULL if NULLs exist.
Modify the column with a NOT NULL constraint.


create table empDetails( empid int not null, name varchar(40)  not null , city varchar(40) not
null);

insert into empDetails values(null,null, null);



primary key
=========
not null+ Unique= Primary Key
================================
MySQL primary key constraint to create the primary key for a table.
```

primary key is a column or a set of columns that uniquely identifies each row in the table. The primary key follows these rules:

A primary key must contain unique values. If the primary key consists of multiple columns, the combination of values in these columns must be unique.
A primary key column cannot have NULL values. Any attempt to insert or update NULL to primary key columns will result in an error. Note that MySQL implicitly adds a NOT NULL constraint to primary key columns.
A table can have one an only one primary key.

MySQL works faster with integers, the data type of the primary key column should be the integer e.g., INT, BIGINT. And you should ensure sure that value ranges of the integer type for the primary key are sufficient for storing all possible rows that the table may have.

Define a PRIMARY KEY constraint in CREATE TABLE

```
create table emp1(
                        id int primary key,
                        name varchar(30) not null,
                        city  varchar(30) not null
                    );
```

```
CREATE TABLE table_name(
   primary_key_column datatype PRIMARY KEY,
   ...
);
```

```
create table emp4(id int primary key, name varchar(30)not null);
```

Define PRIMARY KEY constraints using ALTER TABLE

```
ALTER TABLE table_name
ADD PRIMARY KEY(column_list);
```

```
alter table temp add primary key(id);
```

```
mysql> alter table emp add primary key(id);
ERROR 1062 (23000): Duplicate entry '123' for key 'emp.PRIMARY'
mysql> select * from emp;
+-----+-------+
| id  | name  |
+-----+-------+
| 123 | Abhay |
| 123 | Abhay |
| 123 | Abhay |
+-----+-------+
3 rows in set (0.00 sec)

mysql> delete from emp where id =123;
Query OK, 3 rows affected (0.01 sec)

mysql> select * from emp;
Empty set (0.00 sec)

mysql> alter table emp add primary key(id);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
CREATE TABLE pkdemos(
   id INT,
   title VARCHAR(255) NOT NULL
);
```

DROP Primary Key
==========

```
ALTER TABLE table_name  DROP PRIMARY KEY;
```

alter table temp drop primary key ;

Alter Table emp drop PRIMARY KEY;

ALTER TABLE Login DROP PRIMARY KEY;

ALTER TABLE emp_details DROP PRIMARY KEY;

ALTER TABLE emp_details ADD PRIMARY KEY(id);

ALTER TABLE emp1
ADD PRIMARY KEY(name, id , city);

Foreign Key

The foreign key is used to link one or more than one table together. It is also known as the referencing key. A foreign key matches the primary key field of another table. It means a foreign key field in one table refers to the primary key field of the other table. It identifies each row of another table uniquely that maintains the referential integrity in MySQL

A foreign key makes it possible to create a parent-child relationship with the tables. In this relationship, the parent table holds the initial column values, and column values of child table reference the parent column values. MySQL allows us to define a foreign key constraint on the child table.

MySQL Composite Key

A composite key in MySQL is a combination of two or more than two columns in a table that allows us to identify each row of the table uniquely. It is a type of candidate key which is formed by more than one column. MySQL guaranteed the uniqueness of the column only when they are combined. If they have taken individually, the uniqueness cannot maintain.

Any key such as primary key, super key, or candidate key can be called composite key when they have combined with more than one attribute. A composite key is useful when the table needs to identify each record with more than one attribute uniquely. A column used in the composite key can have different data types. Thus, it is not required to be the same data type for the columns to make a composite key in MySQL.

MySQL UNIQUE constraint

A UNIQUE constraint is an integrity constraint that ensures values in a column or group of columns to be unique. A UNIQUE constraint can be either a column constraint or a table constraint.

```
CREATE TABLE table_name(
    ...,
    column_name data_type UNIQUE,
    ...
);
```

```
CREATE TABLE  abc(

    id int UNIQUE,
   name varchar(30) unique

);
```

```
mysql> select * from emp_details;
+-------+-------+---------+
| empid | name  | city    |
+-------+-------+---------+
|     1 | abhay | Pune    |
|     2 | Kumar | Delhi   |
|     3 | Mohan | Kolkata |
|     4 | Jay   | MumBai  |
+-------+-------+---------+
foreign key(empid) references emp_details(empid)

mysql> select * from dep;
+-----+---------+-------+
| did | depname | empid |
+-----+---------+-------+
| 101 | IT      |     3 |
| 102 | HR      |     1 |
| 103 | admin   |     2 |
| 104 | Sales   |     4 |
+-----+---------+-------+
4 rows in set (0.00 sec)
```

==========
check Constraints
==========

MySQL CHECK Constraint
The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a column it will allow only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

```
-- apply the CHECK constraint to the amount column
CREATE TABLE Orders (
  order_id INT not null,
  amount INT CHECK (amount > 0)
);

-- amount equal to 100
-- record is inserted
INSERT INTO Orders(order_id,amount) VALUES(123,100);
```

votecheck name age >=18
city ='Pune'

==========

```
create table emp1 (age int check(age>=18)  not null, name varchar(30) not null

                        );

insert into emp values (null,null);



CREATE TABLE empval (
  emp_id INT not null ,
  age INT CHECK (age > 18)
);
```

==========
==========

```
create table test (
                    id int not null,
                    sal double not null,
                    city varchar(50) not null
                  );



create table test2 (
                    id int primary key,
                    sal double not null,
                    city varchar(50) not null
                  );
```

list , Numpy, Array