**Primary key:**

**Let’s not go into the definition. Instead, Let’s see, what is done by this.**

It Makes table rows unique And if a field is set as primary key, it cannot be NULL.

Now, Primary Key can be based on a single field as well as multiple fields.

However, unless one field cannot uniquely define a row in a table, don’t use primary keys based on multiple fields.

**Primary key is generally chosen during the table creation.**

Check the following example:

**create table forumAnswers(answer\_id INT NOT NULL AUTO\_INCREMENT, answer\_content BLOB NOT NULL, author VARCHAR(80), posted\_date DATETIME PRIMARY KEY(answer\_id));**

Or,

**create table forumAnswers(answer\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, answer\_content BLOB NOT NULL, author VARCHAR(80), posted\_date DATETIME);**

Or,

**create table forumAnswers(answer\_id INT NOT NULL AUTO\_INCREMENT, answer\_content BLOB NOT NULL, author VARCHAR(80), posted\_date DATETIME, CONSTRAINT forumAnswersKey PRIMARY KEY(answer\_id));**

**Here, answer\_id is the primary key.**

Now, suppose,the same table does not have a answer\_id field.

So, in that case, author and posted\_date\_time combo (if only the posted\_date is changed into posted\_date\_time and the datatype of posted\_date is changed from DATE to DATETIME (A date and time combination in YYYY-MM-DD HH:MM:SS format)) can serve the job of a primary key.

**Create table forumAnswers(answer\_content BLOB NOT NULL, author VARCHAR(80), posted\_date DATETIME, PRIMARY KEY(author,posted\_date\_time));**

However, a different syntax is used and can be applied when the table has only one primary key. That syntax cannot be applied here:

**Create table forumAnswers(answer\_content BLOB NOT NULL, author VARCHAR(80) PRIMARY KEY, posted\_date DATETIME PRIMARY KEY);**

This will give error.

However, the following will work:

**create table forumAnswers(answer\_content BLOB NOT NULL, author VARCHAR(80),posted\_date DATETIME, CONSTRAINT forumAnswersKey PRIMARY KEY(author,posted\_date));**

However, if it is not set during creation of table, alter command can be used.

**Note about Primary Key:**

If on a table field AUTO\_INCREMENT is used, that field must be set as Primary key.

**Add A Single Field As A Primary Key:**

Suppose, a table is created named Person as following:

**create table Person(ID NOT NULL, Name VARCHAR(80) NOT NULL, Age TINYINT NOT NULL);**

Now, it is needed to set the ID as Primary key.

So,

The syntax is following:

alter table Person add PRIMARY KEY(ID);

**Add Multiple Fields As Primary Keys:**

**Create table forumAnswers(answer\_content BLOB NOT NULL, author VARCHAR(80), posted\_date\_time DATETIME);**

And if author and posted\_date\_time are required to add as primary keys:

**alter table forumAnswers add PRIMARY KEY(author,posted\_date\_time);**

**ALTER TABLE forumAnswers  
ADD CONSTRAINT forumAnswersKey PRIMARY KEY (author,posted\_date\_time);**

**Change Fields As Primary Keys:**

**Create table forumAnswers(answer\_content BLOB NOT NULL, author VARCHAR(80), posted\_date\_time DATETIME);**

And, Suppose, initially we set only author as PRIMARY KEY, and later see flaws in that model (We can neither set author nor set posted\_date\_time as primary key. Logic behind this is simple. For two answers posted by different author, posted\_date\_time can be same. And multiple answers can be posted by same author. However, a single author cannot post two answers at exact same datetime. So, that combo could serve as primary key).

**alter table forumAnswers add PRIMARY KEY(author);**

is done. And, now, it requires to be changed.

**alter table forumAnswers drop PRIMARY KEY, add PRIMARY KEY(author,posted\_date\_time));**

**Drop Primary Key:**

**Alter table forumAnswers drop PRIMARY KEY;**

**Drop Primary Key Associated Constraints:**

**alter table forumAnswers drop CONSTRAINT forumAnswersKey ;**

**Unique Key:**

The difference between unique key and primary key is, in unique key, null value is allowed.

The things mentioned in upper section which can be done with primary key, similar things can be done with unique key.

**Foreign Key:**

A FOREIGN KEY is a key used to link two tables together.

A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.

The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

**create table OrdersInformationTable(OrderID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,OrderInfo VARCHAR(80) NOT NULL, PersonID INT NOT NULL, OrderDate DATE NOT NULL, FOREIGN KEY(PersonID) REFERENCES Person(ID));**

**Person table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Age** | **Gender** | **PhoneNumber** | **Address** |
| 1 | Sayak Haldar | 23 | ‘M’ | 9674465435 | J-2, 102/B, DDA Flats, Kalkaji, New Delhi-110019 |
| 2 | Sayantan  Pandit | 23 | ‘M’ | 8697359734 | Dream Home Apartment, Andul Mouri, Howrah-711302 |
| 3 | Suman Banerjee | 24 | ‘M’ | 8697255735 | Andul Purbopara, Andul Mouri, Howrah-711302 |

**And OrdersInformation Table:**

|  |  |  |
| --- | --- | --- |
| **OrderID** | **OrderNumber** | **PersonID** |
| 1 | 77895 | 3 |
| 2 | 44678 | 3 |
| 3 | 22456 | 2 |
| 4 | 24562 | 1 |

The syntax can be improved, though:

**create table Orders(OrderID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,OrderInfo VARCHAR(80) NOT NULL, PersonID INT NOT NULL, OrderDate DATE NOT NULL, FOREIGN KEY(PersonID) REFERENCES Person(ID) ON UPDATE CASCADE ON DELETE RESTRICT);**

**Generic Syntax For Foreign Key:**

[CONSTRAINT [symbol]] FOREIGN KEY  
 [index\_name] (index\_col\_name, ...)  
 REFERENCES tbl\_name (index\_col\_name,...)  
 [ON DELETE reference\_option]  
 [ON UPDATE reference\_option]

**The Reference options are:**

RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT

* **SET NULL:** Delete or update the row from the parent table, and set the foreign key column or columns in the child table to NULL. Both ON DELETE SET NULL and ON UPDATE SET NULL clauses are supported.

If you specify a SET NULL action, make sure that you have not declared the columns in the child table as NOT NULL.

* **RESTRICT:** Rejects the delete or update operation for the parent table. Specifying RESTRICT (or NO ACTION) is the same as omitting the ON DELETE or ON UPDATE clause.
* **NO ACTION:** A keyword from standard SQL. In MySQL, equivalent to RESTRICT. The MySQL Server rejects the delete or update operation for the parent table if there is a related foreign key value in the referenced table. Some database systems have deferred checks, and NO ACTION is a deferred check. In MySQL, foreign key constraints are checked immediately, so NO ACTION is the same as RESTRICT.
* **SET DEFAULT:** This action is recognized by the MySQL parser, but both [InnoDB](https://dev.mysql.com/doc/refman/5.7/en/innodb-storage-engine.html" \o "Chapter 14 The InnoDB Storage Engine) and [NDB](https://dev.mysql.com/doc/refman/5.7/en/mysql-cluster.html" \o "Chapter 21 MySQL NDB Cluster 7.5 and NDB Cluster 7.6) reject table definitions containing ON DELETE SET DEFAULT or ON UPDATE SET DEFAULT clauses.

Let’s try to add a Foreign key using ALTER table syntax:

**Adding Foreign Key Using Alter Statement:**

Suppose, the Person table is created with following:

**create table Person(ID NOT NULL, Name VARCHAR(80) NOT NULL, Age TINYINT NOT NULL);**

And, another table is created:

**create table PersonAddressMap(PersonID INT NOT NULL, Address VARCHAR(80) NOT NULL);**

Now, after this, It is noticed, that no key is set for PersonAddressMap:

**alter table PersonAddressMap ADD FOREIGN KEY(PersonID) REFERENCES Person(ID);**

**Deleting Foreign Key Using Alter Statement:**

**alter table PersonAddressMap Drop FOREIGN KEY PersonId;**