

MYSQL FOREIGN KEY

MySQL 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.

MySQL defines the foreign key in two ways:

- **CREATE TABLE** Statement
- **ALTER TABLE** Statement

1) Foreign Key Using CREATE TABLE Statement

If we want to create only one primary key column into the table, use the below syntax:

```
CREATE TABLE table_name(  
    col1 datatype PRIMARY KEY,  
    col2 datatype,  
    ...  
    [CONSTRAINT constraint_name]  
    FOREIGN KEY [foreign_key_name] (col_name, ...)  
    REFERENCES parent_tbl_name (col_name,...)  
    ON DELETE referenceOption  
    ON UPDATE referenceOption  
);
```

Example:

Table: customer

```
CREATE TABLE customer (  
    ID INT NOT NULL AUTO_INCREMENT,  
    Name varchar(50) NOT NULL,  
    City varchar(50) NOT NULL,  
    PRIMARY KEY (ID)  
);
```

Table: contact

```
CREATE TABLE contact (  
    ID INT,  
    Customer_Id INT,  
    Customer_Info varchar(50) NOT NULL,  
    Type varchar(50) NOT NULL,  
    INDEX par_ind (Customer_Id),  
    CONSTRAINT fk_customer FOREIGN KEY (Customer_Id)  
    REFERENCES customer(ID)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

2) Foreign Key Using ALTER TABLE Statement

If we want to create more than one primary key column into the table, use the below syntax:

```
ALTER TABLE table_name  
ADD [CONSTRAINT [symbol]] FOREIGN KEY  
[index_name] (column_name, ...)  
REFERENCES table_name (column_name,...)  
ON DELETE referenceOption  
ON UPDATE referenceOption
```

Example:

Table: Person

```
CREATE TABLE Person (  
    ID INT NOT NULL AUTO_INCREMENT,  
    Name varchar(50) NOT NULL,  
    City varchar(50) NOT NULL,  
    PRIMARY KEY (ID)  
);
```

Table: Contact

```
CREATE TABLE Contact (  
    ID INT,  
    Person_Id INT,  
    Info varchar(50) NOT NULL,  
    Type varchar(50) NOT NULL  
);
```

After creating a table, if we want to add a foreign key to an existing table, we need to execute the ALTER TABLE statement as below:

```
ALTER TABLE Contact ADD INDEX par_ind ( Person_Id );  
ALTER TABLE Contact ADD CONSTRAINT fk_person  
FOREIGN KEY ( Person_Id ) REFERENCES Person ( ID ) ON DELETE  
CASCADE ON UPDATE CASCADE;
```

3) DROP a Foreign Key

MySQL allows the ALTER TABLE statement to remove an existing foreign key from the table. The following syntax is used to drop a foreign key:

```
ALTER TABLE table_name DROP FOREIGN KEY fk_constraint_name;
```

Here, the table_name is the name of a table from where we are going to remove the foreign key. The constraint_name is the name of the foreign key that was added during the creation of a table.

If we have not known the name of an existing foreign key into the table, execute the following command:

```
mysql> SHOW CREATE TABLE contact;
```

Now, to delete this foreign key constraint (Say the constraint name is **fk_customer**) from the contact table, execute the statement as below:

```
mysql> ALTER TABLE contact DROP FOREIGN KEY fk_customer;
```

REFERENCE_OPTION: It is used to ensure how foreign key maintains referential integrity using ON DELETE and ON UPDATE clause between parent and child table.

MySQL contains **various** referential options, some of are given below:

- **CASCADE:** It is used when we delete or update any row from the parent table, the values of the matching rows in the child table will be deleted or updated automatically.
- **SET NULL:** It is used when we delete or update any row from the parent table, the values of the foreign key columns in the child table are set to NULL.
- **RESTRICT:** It is used when we delete or update any row from the parent table that has a matching row in the reference(child) table, MySQL does not allow to delete or update rows in the parent table.