Primary key

- Uniquely identifies each row / record in a data base table
- Primary key must contains unique values
- A table can contain have only one primary key
- No duplicate values & NULL values

Foreign Key

- Main purpose of foreign key is to ensure integrity of data.
- Allows duplicate values & Null Values

Working with PRIMARY KEY & FOREIGN KEY CONSTRAINTS

Syntax:-

CREATE TABLE <TN> (C1 DATA TYPE PRIMARY KEY, C2 DATA TYPE)

CREATE TABLE <TN1> (C1 DATA TYPE , C2 DATA TYPE, C3 DATA TYPE FOREIGN KEY REFERENCES <TN>(C1))

Example:-

create table Nani (id int primary key, name varchar(20)) insert

nani values('senorita',56), ('raj',61) ← Allowed

insert nani (name) values('hike') ← Not Allowed

create table Nani1(name varchar(10), id int foreign key references nani (id)) insert

nani1 values('senorita',56), ('raj',61) ← Allowed

insert nani1 (name) values('hike') ← Allowed

In these 2 Types Again:- With Out Cascading & with Cascading

With out cascading:-

UPDATION

Not Possible

DELETION

With Cascading:-

UPDATION

Possible

DELETION

How to UPDATION & DELETION

CREATE TABLE RAJ(ID INT PRIMARY KEY, ENAME VARCHAR (50))

INSERT INTO RAJ VALUES(10, 'SRK'),(20, 'SENORITA'),(30, 'DDLJ')

CREATE TABLE SENORITA (PHNUM BIGINT VARCHAR(40), LOCATION VARCHAR(40), ID INT FOREIGN KEY REFERENCES RAJ(ID) ON UPDATE CASCADE ON DELETE CASCADE)

INSERT INTO SENORITA (9876543212, 'BNGLR' , 10),(1234567898 , 'MNGLR', 20),(7654334565, 'RAICHUR', 30)

SELECT * FROM RAJ

SELECT * FROM SENORITA

UPDATE RAJ SET ID = 225 WHERE ID = 10

DELETE FROM RAJ WHERE ID = 30

SELECT * FROM RAJ

SELECT * FROM SENORITA