PRIMARY KEY, NOT NULL, UNIQUE

```
CREATE TABLE CUSTOMER(
      ID INT PRIMARY KEY,
      BRANCH_ID INT NOT NULL,
      FIRSTNAME VARCHAR(50),
      LASTNAME VARCHAR(50),
      DOB DATE,
      GENDER CHAR(1) NOT NULL,
      PHONE CHAR(10) UNIQUE
);

CREATE TABLE CUSTOMER(
      ID INT,
      BRANCH_ID INT NOT NULL,
      FIRSTNAME VARCHAR(50),
      LASTNAME VARCHAR(50),
      DOB DATE,
      GENDER CHAR(1) NOT NULL,
      PHONE CHAR(10) UNIQUE,
      PRIMARY KEY(ID)
);
```

FOREIGN KEY, CHECK AND DEFAULT

CHECK

SYNTAX: CONSTRAINT CONSTRAINT_NAME CHECK(EXPRESSION)

```
CREATE TABLE ACCOUNT(
      ID INT PRIMARY KEY,
      BRANCH_ID INT NOT NULL,
      CUTOMER_ID NOT NULL,
      ACCOUNT_BALANCE DOUBLE,
      ACCOUNT_TYPE VARCHAR(50) DEFAULT 'SAVINGS',
      STATUS VARCHAR(20) DEFAULT 'ACTIVE'
      FOREIGN KEY(CUTOMER_ID) REFERENCES CUSTOMER(ID),
      CONSTRAINT ACCOUNT_BALANCE_CHECK CHECK(ACCOUNT_BALANCE >= 5000)
);
```

REFERENTIAL INTEGRITY CONSTRAINT

1. CASCADE

DELETE CASCADE

When a row in the referenced table is deleted, all rows in the referencing (child) table that have a foreign key referencing the deleted row will also be deleted.

UPDATE CASCADE

When a value in the referenced table is updated, all rows in the referencing (child) table that have a foreign key referencing the updated value will have their foreign key values updated to match the new value.

This cascading effect continues to child tables if they have foreign key constraints as well.

```
CREATE TABLE ACCOUNT(
      ID INT PRIMARY KEY,
      BRANCH_ID INT NOT NULL,
      CUTOMER_ID NOT NULL,
      ACCOUNT_BALANCE DOUBLE,
      ACCOUNT_TYPE VARCHAR(50) DEFAULT 'SAVINGS',
      STATUS VARCHAR(20) DEFAULT 'ACTIVE'
      FOREIGN KEY(CUTOMER_ID) REFERENCES CUSTOMER(ID)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
      CONSTRAINT ACCOUNT_BALANCE_CHECK CHECK(ACCOUNT_BALANCE >= 5000)
);
```

2. SET NULL

   DELETE
   When a row in the referenced table is deleted, the foreign key column(s) in the referencing table will be set to NULL.

   UPDATE
   When a row in the referenced table is deleted, the foreign key column(s) in the referencing table will be set to NULL.

   This is useful when you want to allow "orphaned" rows in the child table.

```
   CREATE TABLE ACCOUNT(
       ID INT PRIMARY KEY,
       BRANCH_ID INT NOT NULL,
       CUTOMER_ID NOT NULL,
       ACCOUNT_BALANCE DOUBLE,
       ACCOUNT_TYPE VARCHAR(50) DEFAULT 'SAVINGS',
       STATUS VARCHAR(20) DEFAULT 'ACTIVE'
       FOREIGN KEY(CUTOMER_ID) REFERENCES CUSTOMER(ID)
       ON DELETE SET NULL
       ON UPDATE SET NULL,
       CONSTRAINT ACCOUNT_BALANCE_CHECK CHECK(ACCOUNT_BALANCE >= 5000)
);
```

3. SET DEFAULT: SIMILAR TO SET NULL

```
   CREATE TABLE ACCOUNT(
       ID INT PRIMARY KEY,
       BRANCH_ID INT NOT NULL,
       CUTOMER_ID NOT NULL,
       ACCOUNT_BALANCE DOUBLE,
       ACCOUNT_TYPE VARCHAR(50) DEFAULT 'SAVINGS',
       STATUS VARCHAR(20) DEFAULT 'ACTIVE'
       FOREIGN KEY(CUTOMER_ID) REFERENCES CUSTOMER(ID)
       ON DELETE SET DEFAULT
       ON UPDATE SET DEFAULT,
       CONSTRAINT ACCOUNT_BALANCE_CHECK CHECK(ACCOUNT_BALANCE >= 5000)
);
```

4. NO ACTION

```
  CREATE TABLE ACCOUNT(
      ID INT PRIMARY KEY,
      BRANCH_ID INT NOT NULL,
      CUTOMER_ID NOT NULL,
      ACCOUNT_BALANCE DOUBLE,
      ACCOUNT_TYPE VARCHAR(50) DEFAULT 'SAVINGS',
      STATUS VARCHAR(20) DEFAULT 'ACTIVE'
      FOREIGN KEY(CUTOMER_ID) REFERENCES CUSTOMER(ID)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
      CONSTRAINT ACCOUNT_BALANCE_CHECK CHECK(ACCOUNT_BALANCE >= 5000)
);
```