**PRIMARY KEY**

**Uniqueness:** Ensures that each record in a table is unique.

**Nullability:** Cannot be NULL (each record must have a value).

**Purpose:** Identifies each record in the table uniquely.

**Number:** There can only be one primary key per table, but it can consist of one or more columns (composite key).

Example:

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY, -- Ensures unique EmployeeID

FirstName VARCHAR(50),

LastName VARCHAR(50)

);

**UNIQUE KEY:**

**Uniqueness:** Ensures the values in the column(s) are unique, similar to the primary key.

**Nullability:** Can accept \*\*one\*\* NULL value (different from the primary key).

**Purpose:** Enforces uniqueness but is more flexible than the primary key (can allow NULL).

**Number:** A table can have multiple unique keys.

**Example:**

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

Email VARCHAR(100) UNIQUE -- Ensures that no two employees have the same email

);

**FOREIGN KEY:**

**Uniqueness:** Does not have to be unique.

**Nullability:** Can accept NULL values, depending on the table design.

**Purpose:** Establishes a relationship between two tables. It references the \*\*primary key\*\* (or unique key) of another table.

**Number:** A table can have multiple foreign keys.

**Example:**

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

EmployeeID INT,

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)

);

In this case, `EmployeeID` in the `Orders` table is a foreign key that references the `EmployeeID` in the `Employees` table, establishing a relationship between the two tables.

**Summary:**

**Primary Key:** Uniquely identifies each record, no NULLs allowed.

**Unique Key:** Ensures uniqueness but allows one NULL value.

**Foreign Key:** Links two tables together, references a primary or unique key in another table.