**IDENTITY**

In SQL Server, you can set a column to auto-increment by using the IDENTITY property. Here's how you can define an auto-incrementing primary key:

**1. Creating a Table with Auto-Increment ID**

CREATE TABLE Users (

Id INT IDENTITY(1,1) PRIMARY KEY,

Name NVARCHAR(100) NOT NULL

);

* IDENTITY(1,1) means:
  + 1: The starting value.
  + 1: The increment step.

**2. Inserting Data (No Need to Specify ID)**

INSERT INTO Users (Name) VALUES ('John Doe'), ('Jane Doe');

**3. Retrieving Data**

SELECT \* FROM Users;

**4. Modifying an Existing Table to Add Auto-Increment**

If you already have a table and want to add an identity column:

ALTER TABLE Users DROP COLUMN Id; -- If it exists without IDENTITY

ALTER TABLE Users ADD Id INT IDENTITY(1,1) PRIMARY KEY;

**5. Reseeding (Reset Auto-Increment Value)**

To reset the IDENTITY value:

DBCC CHECKIDENT ('Users', RESEED, 100); -- Next ID will be 101

**GUID**

In SQL Server, you can generate a new GUID (Globally Unique Identifier) using the NEWID() or NEWSEQUENTIALID() function.

**1. Using NEWID() (Random GUID)**

You can set a column to automatically generate a new GUID by default:

CREATE TABLE Users (

Id UNIQUEIDENTIFIER DEFAULT NEWID() PRIMARY KEY,

Name NVARCHAR(100) NOT NULL

);

* NEWID() generates a completely random GUID.

**2. Using NEWSEQUENTIALID() (Sequential GUID)**

For better performance in indexes, use NEWSEQUENTIALID():

CREATE TABLE Users (

Id UNIQUEIDENTIFIER DEFAULT NEWSEQUENTIALID() PRIMARY KEY,

Name NVARCHAR(100) NOT NULL

);

* NEWSEQUENTIALID() generates GUIDs in increasing order, which improves index performance.

**3. Manually Generating a GUID**

If you want to insert a GUID manually:

INSERT INTO Users (Id, Name) VALUES (NEWID(), 'John Doe');

**4. Retrieving Data**

SELECT \* FROM Users;