--Types of Constraints

--NOT NULL – Specifies that a column cannot be blank

--DEFAULT – Uses a default value, when no other values is entered in the column

--UNIQUE – Provides a unique value for each row

--PRIMARY KEY – Provides an unique identity for each record in a table

--FOREIGN KEY – Provides the referential integrity of the data in one table to match values in another table

--CHECK - Ensures that the data entered in the column meets the condition set for that column

/\*

When you create a table, by default it will allow null values unless you specify otherwise by the NOT NULL constraint.

For example, in the following create table statement, PersonID will not allow nulls, meaning that when entering data into this table,

The PersonID MUST be included in the insert statement, while the Lname and Fname can include blank values.

\*/

--Example of NOT NULL, NULL CONSTRAINT

--CREATE DATABASE

CREATE DATABASE Constraints

Go

USE Constraints

Go

--CREATE TABLE WITH NOT NUL AND NULL CONSTRAINTS

CREATE TABLE Person

(PersonID int NOT NULL,--<< Note NOT NULL. Thus, this column must have a value entered

LName varchar (30) NULL,--<< Note NULL. Thus, this column can omit a value entered

FName varchar(30));

--The following SQL statement,

INSERT INTO Person (LName, FName)

VALUES ('Smith', 'Bob'); --Will result in an error because we omitted in integer for the PersonID. This error out because we put a NOT NULL constraint on this column.

INSERT INTO Person (PersonID, LName, FName)

VALUES (1,'Smith', 'Bob');--<< Note the ProductID (1) is entered.--This insert statement will not result in an error, as a value for PersonID in included in the insert statement.

SELECT \* FROM Person

--Example of DEFAULT CONSTRAINT

/\*The default constraint provides a 'set value' to be inserted into a column, when no other value is provided when using the INSERT INTO statement.

In this example, when we insert data into a table, and omit the score values, the DEFAULT constraint will automatically insert 80.\*/

USE Constraints

GO

CREATE TABLE Student

(StudentID int,

LName varchar (30),

FName varchar (30),

Score int DEFAULT 80);--<< states, if no value is added, then add 80 by default

SELECT \* FROM Student

INSERT INTO Student (StudentID, LName, FName)

VALUES (20, 'Smith’, ‘Tom');--<< No score is added, thus a default of 80 should replace the value

SELECT \* FROM Student

--Example of UNIQUE CONSTRAINT

--The UNIQUE constraint ensures that all values in a column are distinct.

USE Constraints

GO

CREATE TABLE Customer

(CustID int UNIQUE,--<< If a value is entered that violates this rule, the query will error out and terminate

LName varchar (30),

FName varchar(30));

SELECT \* FROM Customer

--Insert records

INSERT INTO Customer

VALUES

(1,'Smith','Henry'),

(2,'Jones','Tomas'),

(3,'Bart','Andy')

SELECT \* FROM Customer

INSERT INTO Customer

VALUES (3, 'Saunders', 'Andy')--<< Note: Will result in an error because it violates the UNIQUE constraint: we already have a CustID of 3.

--Example of PRIMARY KEY CONSTRAINT

--The PRIMARY KEY constraint uniquely identifies each record in a database table.

--Primary keys must contain UNIQUE values.

--A primary key column cannot contain NULL values.

--Most tables should have a primary key, and each table can have only ONE primary key.

CREATE TABLE Bridges

(BridgeID int identity (1,1) Not null Primary Key,--<< Note the identity (1,1). This auto increments the primary key BridgeID. Cannot be blank

BridgeName varchar (50) Null, --<<Note Null. This states that the column can contain omitted data

BridgeCity varchar (50) Null,

BridgeCountry varchar (50) Null)

Select \* from Bridges

--Example of FOREIGN KEY CONSTRAINT

--A FOREIGN KEY in one table points to a PRIMARY KEY in another table.

--When creating a table the references another table, we use the foreign key to make a relationship.

--This will be explored more when we discuss joins

CREATE TABLE BridgeBuilders

(BridgeBuildersId int NOT NULL,

BridgeBuildersName varchar (50) NULL,

BridgeID int,

PRIMARY KEY (BridgeBuildersId),

FOREIGN KEY (BridgeID) REFERENCES Bridges (BridgeID))

Select \* from BridgeBuilders

--Example of CHECK CONSTRAINT

/\*The CHECK constraint ensures that all values in a column satisfy certain conditions.

Once defined, the database will only permit a new row or update an existing row if the new value satisfies the CHECK constraint.

The CHECK constraint is used to ensure data quality.\*/

--To create a CHECK constraint on the 'CustID'column when the table is already created, use the following ALTER TABLE command

USE Constraints

GO

--Using the ALTER TABLE command after the table has been created

ALTER TABLE Customer

ADD CHECK ( CustID >0)

INSERT INTO Customer VALUES (-5, 'Shaffer', 'Georgina') --<<This insert will fail because it violates the CHECK constraint that CustID must be a positive value.

INSERT INTO Customer VALUES (5, 'Shaffer', 'Georgina')

SELECT \* FROM Customer

--Example 2

ALTER TABLE Student

ADD CHECK ( Score > 1 and Score < 100)

--Adding a value that does not violate the CHECK rule

INSERT INTO Student (StudentID, LName, FName, Score)

VALUES (20, 'Smith','Tom',98)

Select \* from Student

--Adding a value that does violate the CHECK rule

INSERT INTO Student (StudentID, LName, FName, Score)

VALUES (20, 'Smith','Tom',110)

Select \* from Student

Use master

Go

Drop Database Constraints