CONSTRAINT

Ques. What is constraint. How to add and delete constraints?

SQL constraints are used to specify rules for data in a table.

OR

Constraints are rules and restrictions applied on a column or a table such that unwanted data can't be inserted into tables. This ensures the accuracy and reliability of the data in the database. We can create constraints on single or multiple columns of any

Syntax

CREATE TABLE table_name

```
(
   column1 datatype constraint,
   column2 datatype constraint,
   column3 datatype constraint,
   ....
);
```

Check Constraints

Select * from tblPerson

To check the table infromation select the table and press
Alt + F1

SQL Constraints

The following constraints are commonly used in SQL:

NOT NULL - Ensures that a column cannot have a NULL value
UNIQUE - Ensures that all values in a column are different
PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
FOREIGN KEY - Uniquely identifies a row/record in another table
CHECK - Ensures that all values in a column satisfies a specific condition
DEFAULT - Sets a default value for a column when no value is specified
INDEX - Used to create and retrieve data from the database very quickly

NOT NULL Constraint

By default, a column can hold NULL values.

The NOT NULL constraint enforces a column to NOT accept NULL values.

This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

Example

SQL NOT NULL on CREATE TABLE

```
CREATE TABLE tblPerson (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255) NOT NULL,
    Age int
);
```

SQL NOT NULL on ALTER TABLE

To create a NOT NULL constraint on the "Age" column when the "Persons" table is already created, use the following SQL:

```
ALTER TABLE tblPerson
ALTER COLUMN Age INT NOT NULL;
```

Alter New Column with NULL

```
ALTER TABLE tblPerson
ADD Gender char(10) NOT NULL;
```

SQL UNIQUE Constraint

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

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

SQL UNIQUE Constraint on CREATE TABLE

```
Create table tblPerson
(
EmpID int not null primary key,
Name varchar(20) not null,
Email varchar(10) UNIQUE,
GenderID int
)
```

SQL UNIQUE Constraint on ALTER TABLE

```
ALTER TABLE tblPerson
ADD CONSTRAINT UC_tblPerson_Email UNIQUE(Email)
```

SQL UNIQUE Constraint on DROP TABLE

```
ALTER TABLE TBLPERSON
DROP CONSTRAINT UC_TBLPERSON_EMAIL
```

SQL PRIMARY KEY Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table.
Primary keys must contain UNIQUE values, and cannot contain NULL values.
A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields)

SQL PRIMARY KEY on CREATE TABLE

```
CREATE TABLE Persons
(
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int
);
```

SQL PRIMARY KEY on ALTER TABLE

We can alter the primary key when the column is NOT NULL.

```
ALTER TABLE Persons
ADD CONSTRAINT PK_Person PRIMARY KEY (ID);
```

DROP a PRIMARY KEY Constraint

```
ALTER TABLE Persons
DROP CONSTRAINT PK_Person;
```

SQL DEFAULT Constraint

The DEFAULT constraint is used to provide a default value for a column.

The default value will be added to all new records IF no other value is specified.

```
Create table tblPerson
(
Id int primary key,
Name char (30),
Email varchar(50),
Age int default 3,
City char (30)
)
```

SQL DEFAULT on ALTER TABLE

```
ALTER TABLE tblPerson
ADD CONSTRAINT DF_ tblPerson_City DEFAULT 'Delhi' FOR CITY;
```

DROP a DEFAULT Constraint

```
ALTER TABLE PERSONS
DROP CONSTRAINT DF_ tblPerson_City;
```

SQL CHECK Constraint

The CHECK constraint is used to limit the value range that can be placed in a column.
If you define a CHECK constraint on a single column it allows only certain values for this column.
If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

SQL CHECK on CREATE TABLE

```
CREATE TABLE tblPersons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int check (age>18),
    City varchar(255)
);
```

Table without Constraint

```
Create table tblperson

(
Id int primary key,
Name char (30),
Email varchar(50),
Age int
)
```

SQL CHECK on ALTER TABLE

```
ALTER TABLE tblPerson
ADD CONSTRAINT CK_tblPerson_Age CHECK (Age>=18);
```

DROP a CHECK Constraint

```
ALTER TABLE tblPerson
DROP CONSTRAINT CK_tblPerson_Age ;
```

SQL FOREIGN KEY Constraint

A FOREIGN KEY is a key used to link two tables together.

A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.

The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table

Look at the following two tables:

"Persons" table:

PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

"Orders" table:

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3

3	22456	2
4	24562	1

SQL FOREIGN KEY on CREATE TABLE

```
Create table tblPerson
(
EmpID int not null primary key,
Name varchar(20) not null,
Email varchar(10),
GenderID int Foreign Key References tblGender(ID)
)

Create table tblGender
(
ID int not null primary key,
Gender nvarchar(50) not null
)
```

SQL FOREIGN KEY on ALTER TABLE

```
ALTER TABLE tblPerson
ADD CONSTRAINT FK_ tblPerson_GID
FOREIGN KEY (GenderID) REFERENCES tblGender (ID);
```

DROP a FOREIGN KEY Constraint

```
ALTER TABLE Orders
DROP CONSTRAINT FK_tblPerson_GID;
```

Cascading referential integrity

Cascade referntial intergity constraints allow to define the action Micorsoft SQL Server should take when a user attempts to delete or update a key to which an existing foregin keys points.

For example, if you delete row with ID=1 from tblGender table, then row with ID =3 from tblPerson table become an orphan record. You will not be able to Gender for this row. So,cascading referential integrity constraints can be use to define actions Micorsoft SQL Server should take when this happens. By default, we get an error and delete or update statement is roll back.

Ans A column default can be specified using default constraint. The DEFAULT constraint is used to insert a default value into a column. The default value will be added to all new records, if no other value is specified, including NULL.

SET NULL
If a delete statement affects rows in a foreign key table, those values will be set to NULL when the primary key record is deleted. If an update statement affects rows in the foreign key table, those rows will be updated with the value NULL after the primary key record has been updated. The foreign key columns affected must allow NULL values.

CASCADE
If a delete statement affects one or more rows in a foreign key table, those rows will be deleted when the primary key record is deleted. If an update statement affects rows in the foreign key table, those rows will be updated with the value from the primary key record after it has been updated.

SET DEFAULT
All the values that make up the foreign key in the rows that are referenced are set to their default value. All foreign key columns in the related table must have default constraints defined on them.

NO ACTION
This is the default action. This specifies that if an update or delete statement affects rows in foreign key tables, the action will be denied and rolled back. An error message will be raised.