SQL Server CREATE TABLE Statement

This SQL tutorial explains how to use the **CREATE TABLE statement** in SQL Server. This tutorial is the first part of two posts describing DDL (Data Definition Language) statements in SQL Server.

The DDL statements are a subset of SQL statements used to create, modify, or remove database structures. In this post you will learn how to create and delete tables.

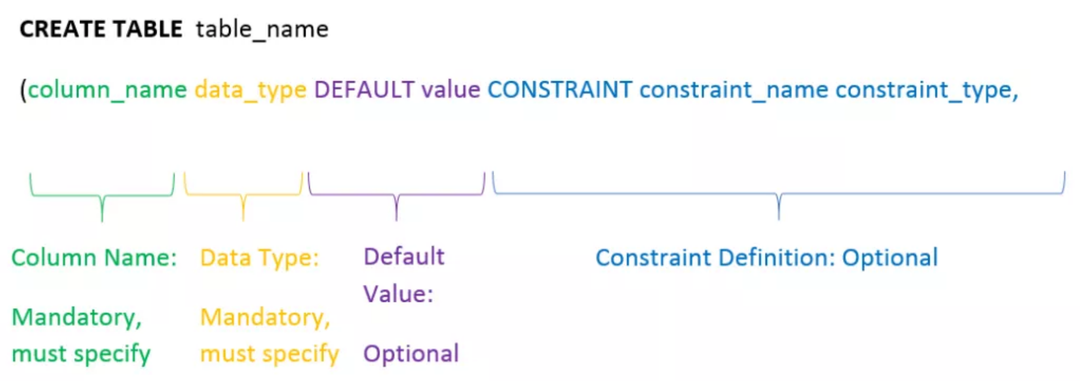
This tutorial allows you to become familiar with the following topics:

- Creating new tables
  - Defining Data Types
  - Defining Default Value
  - Defining Constraints
  - Column Level Constraints
    - Primary Key
    - Unique
    - Not Null
    - Check
    - Foreign Key
  - Table Level Constraints
- Drop Existing Table

The next post will describe how to use the SQL Server ALTER TABLE statement.

SQL Server CREATE TABLE Statement

SQL Server CREATE TABLE statement is used to create new tables in the database.


CREATE TABLE table_name
(column_name data_type DEFAULT value CONSTRAINT constraint_name constraint_type,
Column Name: Mandatory, must specify
Data Type: Mandatory, must specify
Default Value: Optional
Constraint Definition: Optional

Data Types

Column Type	Description	Example
VARCHAR (size)	String column. The value within the brackets indicates the maximum size of each field in the column (in characters)	VARCHAR(3) → 'ABC' VARCHAR(3) → 'AB'
Decimal (p,s)	Numeric column. **P**recision – number of digits, **S**cale – how many of the digits are located after the decimal point	DECIMAL(5,2) → 476.29 DECIMAL(5,2) → 6.29
DATE	Date format column	'YYYY-MM-DD'

SQL Server Default Value

A column can be given a default value using the DEFAULT keyword. The DEFAULT keyword provides a default value to a column when the SQL Server INSERT INTO statement does not provide a specific value. The default value can be a literal value, an expression, or a SQL Function, such as GETDATE().

To define a Default value, use this syntax:

```
DEFAULT default_value
```

For example:

```
CREATE TABLE demo_tbl
(
salary DECIMAL(8,2) DEFAULT 9500,
hire_date DATE DEFAULT '2011-01-27' ,
birthdate DATE DEFAULT GETDATE()
)

INSERT INTO demo_tbl
VALUES (DEFAULT, DEFAULT, DEFAULT)

SELECT *
FROM demo_tbl

salary   hire_date   birthdate
------   ---------   ----------
9500     2011-01-27  2014-01-13
```

Creating SQL Server Constraints

Constraints enforce rules on the data in a table whenever a row is inserted, deleted, or updated. Constraints can be defined at the column or table level.

Defining Constraints at the Column Level

Constraint enforced at the column level:

- Is created as part of the column definition

- Always refers to a single column
- A constraint at the column level has the following structure:

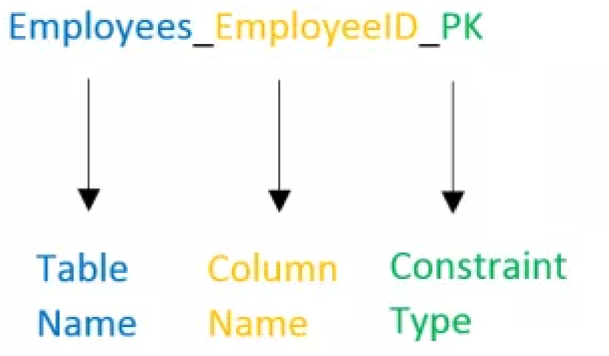
```
CONSTRAINT constraint_name constraint_type
```

- Constraint_type – the type of the constraint to be enforced on the column (for example, Unique or Not Null)
- Constraint_name – although not mandatory, it is always advisable to give the constraint a name, thereby allowing you to easily identify it.

The following naming convention is commonly used by many database developers :

<table name>_<column_name>_<constraint abbreviation>

For example:



Primary Key (PK)

In SQL Server, the Primary Key constraint is a column (or a set of columns) that uniquely identifies each row in the table, this constraint enforces uniqueness and ensures that no column that is part of the Primary Key can hold a NULL value. Only one Primary Key can be created for each table.

The syntax for defining a Primary Key Constraint is as follows:

```
column_name column_DataType [DEFAULT value] [CONSTRAINT
```

For example:

```
CREATE TABLE emps
```

```
(emp_id decimal(3) CONSTRAINT emps_empid_pk PRIMARY KEY
 emp_name varchar(25))
```

Please note – the square brackets in this demonstration (and in those that follow) indicate that what enclosed within them is optional, the square brackets are not part of the CREATE TABLE statement.

Not Null (NN)

In SQL Server, the Not Null constraint ensures that the column contains no NULL values. The syntax for defining a Not Null constraint is as follows:

```
column_name column_DataType [DEFAULT value] [CONSTRAINT
```

For example:

```
CREATE TABLE emps
(emp_id decimal(3)    CONSTRAINT emps_empid_pk PRIMARY
 emp_name varchar(25) CONSTRAINT emps_emnm_nn NOT NULL)
```

This constraint can only be defined at the column level

UNIQUE (UQ)

In SQL Server, the Unique constraint requires that every value in a column (or set of columns) be unique. The syntax for defining a UNIQUE Constraint is as follows:

```
column_name column_DataType [DEFAULT value] [CONSTRAINT
```

For example:

```
CREATE TABLE emps
(emp_id decimal(3)      CONSTRAINT emps_empid_pk PRIMAR
 emp_name varchar(25) CONSTRAINT emps_emnm_nn NOT NULL,
 emp_phone varchar(25)CONSTRAINT emps_empn_uq UNIQUE)
```

CHECK (CK)

In SQL Server, the Check constraint defines a condition that each row must satisfy. The syntax for defining a Check Constraint is as follows:

```
column_name column_DataType [DEFAULT value] [CONSTRAINT
```

- The condition written in the CHECK is quite similar in its structure to each of the conditions written in a WHERE statement.
- The condition in the CHECK part must not include:
  - Values that are returned as a result of using SEQUENCES
  - Functions such as GETDATE()
  - Subqueries

For Example:

```
CREATE TABLE emps
(emp_id decimal(3)        CONSTRAINT emps_empid_pk PRIN
 emp_name varchar(25)   CONSTRAINT emps_emnm_nn NOT NUL
 emp_phone varchar(25)   CONSTRAINT emps_empn_uq UNIQUE
 emp_mail    varchar(25) CONSTRAINT emps_emml_ck CHECK
```

Another example:

```
CREATE TABLE emps
(emp_id decimal(3)      CONSTRAINT emps_empid_pk PRIMAR
 emp_name varchar(25)    CONSTRAINT emps_emnm_nn NOT NU
 emp_phone varchar(25)   CONSTRAINT emps_empn_uq UNIQUE
 emp_mail    varchar(25) CONSTRAINT emps_emml_ck CHECK
 emp_sal      decimal(8,2)  CONSTRAINT emp_sal_ck CHECK
```

FOREIGN KEY (FK)

In SQL Server, the Foreign Key constraint designates a column (or a set of columns) as a Foreign Key and establishes a relationship between a Primary Key (or Unique) in different table (or in the same table). The syntax for defining a Check Constraint is as follows:

```
column_name  … [CONSTRAINT constraint_name] REFERENCES
```

Example:

The Parent Table

```
CREATE TABLE deps
(dep_id decimal(3) CONSTRAINT deps_id_pk PRIMARY KEY ,
 dep_name varchar(25))
```

The Child Table

```
CREATE TABLE emps
(emp_id decimal(3)      CONSTRAINT emps_empid_pk PRIMAR
 emp_name varchar2(25) CONSTRAINT emps_emnm_nn NOT NULL
 emp_phone varchar2(25)   CONSTRAINT emps_empn_uq UNIQU
 emp_mail    varchar2(25) CONSTRAINT emps_emml_ck CHECK
 emp_sal      decimal(8,2)  CONSTRAINT emp_sal_ck CHECK
 dep_id decimal(3) CONSTRAINT emp_depid_fk REFERENCES d
```

Table Level Constraints

- Created after defining the various column.
- Can refer to more than one column (a constraint that comprises two columns together).
- Allows creating several constraints on the same column.
- It is not possible to create a NOT NULL constraint by using this method.

For example:

```
CREATE TABLE emps
(emp_id DECIMAL(3),
 emp_f_name VARCHAR(25) ,
 emp_l_name VARCHAR(25) ,
 emp_phone VARCHAR(25) CONSTRAINT emps_empn_nn NOT NUL
 emp_mail    VARCHAR(25) ,
 emp_sal      DECIMAL(8,2) ,
 dep_id       DECIMAL(3),
 CONSTRAINT emps_empid_pk  PRIMARY KEY (emp_id),
 CONSTRAINT emps_empn_uq   UNIQUE(emp_f_name, emp_l_nc
 CONSTRAINT emps_emml_ck1 CHECK (emp_mail LIKE '_%@%.%
 CONSTRAINT emps_emml_ck2 CHECK (LENGTH(emp_mail) &amp
 CONSTRAINT emps_emml_uq  UNIQUE (emp_mail) ,
 CONSTRAINT emp_sal_ck    CHECK (emp_sal &amp;amp;amp;
 CONSTRAINT emp_depid_fk  FOREIGN KEY (dep_id)
 REFERENCES deps(dep_id)  )
```

Drop an Existing Table

The syntax used for deleting an existing table in SQL Server is as follows :

```
DROP TABLE table_name
```

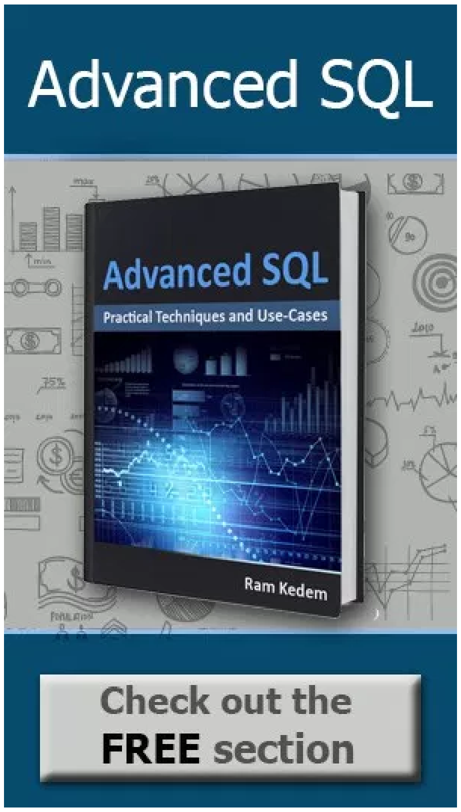
For example

```
DROP TABLE employees
```

Share this:

Google Facebook LinkedIn Twitter Print
Email

Login


Advanced SQL
Advanced SQL
Practical Techniques and Use-Cases
Ram Kedem
Check out the
FREE section