**[CREATE TABLE](https://msdn.microsoft.com/en-us/library/ms174979.aspx)**

    [ database\_name . [ schema\_name ] . | schema\_name . ] table\_name

    [ AS FileTable ]

    ( { <column\_definition> | <computed\_column\_definition>

        | <column\_set\_definition> | [ <table\_constraint> ] [ ,...n ] } )

[ ON { partition\_scheme\_name ( partition\_column\_name ) | filegroup

| "default" } ]

[ { TEXTIMAGE\_ON { filegroup | "default" } ]

[ FILESTREAM\_ON { partition\_scheme\_name | filegroup

| "default" } ]

[ WITH ( <table\_option> [ ,...n ] ) ]

[ ; ]

<column\_definition> ::=

column\_name <data\_type>

[ FILESTREAM ]

    [ COLLATE collation\_name ]

    [ SPARSE ]

    [ NULL | NOT NULL ]

    [

[ CONSTRAINT constraint\_name ] DEFAULT constant\_expression ]

| [ IDENTITY [ ( seed ,increment ) ] [ NOT FOR REPLICATION ]

]

    [ ROWGUIDCOL ]

    [ <column\_constraint> [ ...n ] ]

<data type> ::=

[ type\_schema\_name . ] type\_name

[ ( precision [ , scale ] | max |

[ { CONTENT | DOCUMENT } ] xml\_schema\_collection ) ]

<column\_constraint> ::=

[ CONSTRAINT constraint\_name ]

{     { PRIMARY KEY | UNIQUE }

        [ CLUSTERED | NONCLUSTERED ]

        [

WITH FILLFACTOR = fillfactor

        | WITH ( < index\_option > [ , ...n ] )

        ]

        [ ON { partition\_scheme\_name ( partition\_column\_name )

| filegroup | "default" } ]

  | [ FOREIGN KEY ]

        REFERENCES [ schema\_name . ] referenced\_table\_name [ ( ref\_column ) ]

        [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]

        [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]

        [ NOT FOR REPLICATION ]

  | CHECK [ NOT FOR REPLICATION ] ( logical\_expression )

}

<computed\_column\_definition> ::=

column\_name AS computed\_column\_expression

[ PERSISTED [ NOT NULL ] ]

[

[ CONSTRAINT constraint\_name ]

    { PRIMARY KEY | UNIQUE }

        [ CLUSTERED | NONCLUSTERED ]

        [

WITH FILLFACTOR = fillfactor

          | WITH ( <index\_option> [ , ...n ] )

        ]

        [ ON { partition\_scheme\_name ( partition\_column\_name )

| filegroup | "default" } ]

   | [ FOREIGN KEY ]

        REFERENCES referenced\_table\_name [ ( ref\_column ) ]

        [ ON DELETE { NO ACTION | CASCADE } ]

        [ ON UPDATE { NO ACTION } ]

        [ NOT FOR REPLICATION ]

   | CHECK [ NOT FOR REPLICATION ] ( logical\_expression )

]

<column\_set\_definition> ::=

column\_set\_name XML COLUMN\_SET FOR ALL\_SPARSE\_COLUMNS

< table\_constraint > ::=

[ CONSTRAINT constraint\_name ]

{

{ PRIMARY KEY | UNIQUE }

        [ CLUSTERED | NONCLUSTERED ]

        (column [ ASC | DESC ] [ ,...n ] )

        [

WITH FILLFACTOR = fillfactor

         |WITH ( <index\_option> [ , ...n ] )

        ]

        [ ON { partition\_scheme\_name (partition\_column\_name)

| filegroup | "default" } ]

    | FOREIGN KEY

        ( column [ ,...n ] )

        REFERENCES referenced\_table\_name [ ( ref\_column [ ,...n ] ) ]

        [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]

        [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]

        [ NOT FOR REPLICATION ]

    | CHECK [ NOT FOR REPLICATION ] ( logical\_expression )

}

<table\_option> ::=

{

    [DATA\_COMPRESSION = { NONE | ROW | PAGE }

      [ ON PARTITIONS ( { <partition\_number\_expression> | <range> }

      [ , ...n ] ) ]]

    [ FILETABLE\_DIRECTORY = <directory\_name> ]

    [ FILETABLE\_COLLATE\_FILENAME = { <collation\_name> | database\_default } ]

    [ FILETABLE\_PRIMARY\_KEY\_CONSTRAINT\_NAME = <constraint\_name> ]

    [ FILETABLE\_STREAMID\_UNIQUE\_CONSTRAINT\_NAME = <constraint\_name> ]

    [ FILETABLE\_FULLPATH\_UNIQUE\_CONSTRAINT\_NAME = <constraint\_name> ]

}

<index\_option> ::=

{

PAD\_INDEX = { ON | OFF }

| FILLFACTOR = fillfactor

| IGNORE\_DUP\_KEY = { ON | OFF }

| STATISTICS\_NORECOMPUTE = { ON | OFF }

| ALLOW\_ROW\_LOCKS = { ON | OFF}

| ALLOW\_PAGE\_LOCKS ={ ON | OFF}

| DATA\_COMPRESSION = { NONE | ROW | PAGE }

       [ ON PARTITIONS ( { <partition\_number\_expression> | <range> }

       [ , ...n ] ) ]

}

<range> ::= <partition\_number\_expression> TO <partition\_number\_expression>

[**CREATE { PROC } [schema\_name.] procedure\_name [ ; number ]**](https://msdn.microsoft.com/en-us/library/ms187926.aspx)

    [ { @parameter [ type\_schema\_name. ] data\_type }

        [ VARYING ] [ = default ] [ OUT | OUTPUT | [READONLY]

    ] [ ,...n ]

[ WITH <procedure\_option> [ ,...n ] ]

[ FOR REPLICATION ]

AS { [ BEGIN ] sql\_statement [;] [ ...n ] [ END ] }

[;]

<procedure\_option> ::=

    [ ENCRYPTION ]

    [ RECOMPILE ]

    [ EXECUTE AS Clause ]

[**sp\_executesql**](https://msdn.microsoft.com/en-CA/library/ms188001.aspx) [ @stmt = ] statement

[

  { , [ @params = ] N'@parameter\_name data\_type [ OUT | OUTPUT ][ ,...n ]' }

     { , [ @param1 = ] 'value1' [ ,...n ] }

]

DECLARE @sql nvarchar(max);

DECLARE @param nvarchar(max);

DECLARE @count int;

DECLARE @lro int;

SET @lro = 197;

SET @sql = N'

SELECT @countOUT = COUNT(\*)

FROM DSDW.Community.ReferralFact AS ref\_fact

WHERE ref\_fact.LocalReportingOfficeID = @lroIN';

SET @param = N'

@lroIN int

,@countOUT int OUTPUT';

EXECUTE sp\_executesql @sql

,@param

,@lroIN = @lro, @countOUT = @count OUTPUT;

SELECT @count AS cnt;

[**CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX**](https://msdn.microsoft.com/en-us/library/ms188783.aspx) index\_name

    ON <object> ( column [ ASC | DESC ] [ ,...n ] )

    [ INCLUDE ( column\_name [ ,...n ] ) ]

[ WHERE <filter\_predicate> ]

    [ WITH ( <relational\_index\_option> [ ,...n ] ) ]

    [ ON { partition\_scheme\_name ( column\_name )

         | filegroup\_name

         | default

         }

    ]

[ FILESTREAM\_ON { filestream\_filegroup\_name | partition\_scheme\_name | "NULL" } ]

[ ; ]

<object> ::=

{

    [ database\_name. [ schema\_name ] . | schema\_name. ]

    table\_or\_view\_name

}

<relational\_index\_option> ::=

{

    PAD\_INDEX = { ON | OFF }

  | FILLFACTOR = fillfactor

  | SORT\_IN\_TEMPDB = { ON | OFF }

  | IGNORE\_DUP\_KEY = { ON | OFF }

  | STATISTICS\_NORECOMPUTE = { ON | OFF }

  | DROP\_EXISTING = { ON | OFF }

  | ONLINE = { ON | OFF }

  | ALLOW\_ROW\_LOCKS = { ON | OFF }

  | ALLOW\_PAGE\_LOCKS = { ON | OFF }

  | MAXDOP = max\_degree\_of\_parallelism

  | DATA\_COMPRESSION = { NONE | ROW | PAGE}

     [ ON PARTITIONS ( { <partition\_number\_expression> | <range> }

     [ , ...n ] ) ]

}

<filter\_predicate> ::=

<conjunct> [ AND <conjunct> ]

<conjunct> ::=

<disjunct> | <comparison>

<disjunct> ::=

column\_name IN (constant ,...n)

<comparison> ::=

column\_name <comparison\_op> constant

<comparison\_op> ::=

{ IS | IS NOT | = | <> | != | > | >= | !> | < | <= | !< }

<range> ::=

<partition\_number\_expression> TO <partition\_number\_expression>

[ WITH <common\_table\_expression> [,...n] ]

[**MERGE**](https://msdn.microsoft.com/en-us/library/bb510625.aspx)

    [ TOP ( expression ) [ PERCENT ] ]

    [ INTO ] <target\_table> [ WITH ( <merge\_hint> ) ] [ [ AS ] table\_alias ]

    USING <table\_source>

    ON <merge\_search\_condition>

    [ WHEN MATCHED [ AND <clause\_search\_condition> ]

        THEN <merge\_matched> ] [ ...n ]

    [ WHEN NOT MATCHED [ BY TARGET ] [ AND <clause\_search\_condition> ]

        THEN <merge\_not\_matched> ]

    [ WHEN NOT MATCHED BY SOURCE [ AND <clause\_search\_condition> ]

        THEN <merge\_matched> ] [ ...n ]

    [ <output\_clause> ]

    [ OPTION ( <query\_hint> [ ,...n ] ) ]

;

<target\_table> ::=

{

    [ database\_name . schema\_name . | schema\_name . ]

  target\_table

}

<merge\_hint>::=

{

    { [ <table\_hint\_limited> [ ,...n ] ]

    [ [ , ] INDEX ( index\_val [ ,...n ] ) ] }

}

<table\_source> ::=

{

    table\_or\_view\_name [ [ AS ] table\_alias ] [ <tablesample\_clause> ]

        [ WITH ( table\_hint [ [ , ]...n ] ) ]

  | rowset\_function [ [ AS ] table\_alias ]

        [ ( bulk\_column\_alias [ ,...n ] ) ]

  | user\_defined\_function [ [ AS ] table\_alias ]

  | OPENXML <openxml\_clause>

  | derived\_table [ AS ] table\_alias [ ( column\_alias [ ,...n ] ) ]

  | <joined\_table>

  | <pivoted\_table>

  | <unpivoted\_table>

}

<merge\_search\_condition> ::=

    <search\_condition>

<merge\_matched>::=

{ UPDATE SET <set\_clause> | DELETE }

<set\_clause>::=

SET

{ column\_name = { expression | DEFAULT | NULL }

| { udt\_column\_name.{ { property\_name = expression

| field\_name = expression }

| method\_name ( argument [ ,...n ] ) }

}

| column\_name { .WRITE ( expression , @Offset , @Length ) }

| @variable = expression

| @variable = column = expression

| column\_name { += | -= | \*= | /= | %= | &= | ^= | |= } expression

| @variable { += | -= | \*= | /= | %= | &= | ^= | |= } expression

| @variable = column { += | -= | \*= | /= | %= | &= | ^= | |= } expression

} [ ,...n ]

<merge\_not\_matched>::=

{

    INSERT [ ( column\_list ) ]

        { VALUES ( values\_list )

        | DEFAULT VALUES }

}

<clause\_search\_condition> ::=

    <search\_condition>

<search condition> ::=

    { [ NOT ] <predicate> | ( <search\_condition> ) }

    [ { AND | OR } [ NOT ] { <predicate> | ( <search\_condition> ) } ]

[ ,...n ]

<predicate> ::=

    { expression { = | < > | ! = | > | > = | ! > | < | < = | ! < } expression

    | string\_expression [ NOT ] LIKE string\_expression

  [ ESCAPE 'escape\_character' ]

    | expression [ NOT ] BETWEEN expression AND expression

    | expression IS [ NOT ] NULL

    | CONTAINS

  ( { column | \* } , '< contains\_search\_condition >' )

    | FREETEXT ( { column | \* } , 'freetext\_string' )

    | expression [ NOT ] IN ( subquery | expression [ ,...n ] )

    | expression { = | < > | ! = | > | > = | ! > | < | < = | ! < }

  { ALL | SOME | ANY} ( subquery )

    | EXISTS ( subquery ) }

<output\_clause>::=

{

    [ OUTPUT <dml\_select\_list> INTO { @table\_variable | output\_table }

        [ (column\_list) ] ]

    [ OUTPUT <dml\_select\_list> ]

}

<dml\_select\_list>::=

    { <column\_name> | scalar\_expression }

        [ [AS] column\_alias\_identifier ] [ ,...n ]

<column\_name> ::=

    { DELETED | INSERTED | from\_table\_name } . { \* | column\_name }

  | $action

[OPENROWSET](https://msdn.microsoft.com/en-CA/library/ms190312.aspx)

[OPENXML](https://msdn.microsoft.com/en-us/library/ms186918.aspx)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**decimal**[ **(***p*[ **,** *s*] **)**] and **numeric**[ **(***p*[ **,** *s*] **)**]](https://msdn.microsoft.com/en-us/library/ms187746.aspx)  Fixed precision and scale numbers. When maximum precision is used, valid values are from - 10^38 +1 through 10^38 - 1. The ISO synonyms for **decimal** are **dec** and **dec(***p*, *s***)**. **numeric** is functionally equivalent to **decimal**.  p (precision)  The maximum total number of decimal digits that will be stored, both to the left and to the right of the decimal point. The precision must be a value from 1 through the maximum precision of 38. The default precision is 18.  *s*(scale)   |  |  | | --- | --- | | Precision | Storage bytes | | 1 - 9 | 5 | | 10-19 | 9 | | 20-28 | 13 | | 29-38 | 17 |   The number of decimal digits that will be stored to the right of the decimal point. This number is substracted from *p* to determine the maximum number of digits to the left of the decimal point. The maximum number of decimal digits that can be stored to the right of the decimal point. Scale must be a value from 0 through *p*. Scale can be specified only if precision is specified. The default scale is 0; therefore, 0 <= *s* <= *p*. Maximum storage sizes vary, based on the precision. | [**sys.objects type:**](https://msdn.microsoft.com/en-us/library/ms190324.aspx)  AF = Aggregate function (CLR)  C = CHECK constraint  D = DEFAULT (constraint or stand-alone)  F = FOREIGN KEY constraint  FN = SQL scalar function  FS = Assembly (CLR) scalar-function  FT = Assembly (CLR) table-valued function  IF = SQL inline table-valued function  IT = Internal table  P = SQL Stored Procedure  PC = Assembly (CLR) stored-procedure  PG = Plan guide  PK = PRIMARY KEY constraint  R = Rule (old-style, stand-alone)  RF = Replication-filter-procedure  S = System base table  SN = Synonym  SO = Sequence object  SQ = Service queue  TA = Assembly (CLR) DML trigger  TF = SQL table-valued-function  TR = SQL DML trigger  TT = Table type  U = Table (user-defined)  UQ = UNIQUE constraint  V = View  X = Extended stored procedure |