Syntax Reference

CREATE TABLE table\_name

(

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

)

<column\_definition> ::=

column\_name data\_type

[NULL | NOT NULL]

[[CONSTRAINT constraint\_name] DEFAULT constant\_expression]

| [IDENTITY [(seed, increment)]

]

[<column\_constraint> [ ,...n ]]

<column\_constraint> ::=

CONSTRAINT constraint\_name

{ {PRIMARY KEY | UNIQUE} [CLUSTERED | NONCLUSTERED]

| [FOREIGN KEY] REFERENCES referenced\_table\_name (ref\_column)

| CHECK (logical\_expression)

}

< table\_constraint > ::=

CONSTRAINT constraint\_name

{

{PRIMARY KEY | UNIQUE} [CLUSTERED | NONCLUSTERED] (column\_name [ ,...n ])

| FOREIGN KEY (column\_name [ ,...n ])

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

| CHECK (logical\_expression)

}

ALTER TABLE table\_name

{ [{check | nocheck} CONSTRAINT constraint\_name]

| [with {CHECK | NOCHECK}]

{{ADD {<column\_definition> | <table\_constraint>}

| ALTER COLUMN <column\_definition>}

| drop {COLUMN column\_name | CONSTRAINT constraint\_name}

}

CREATE [clustered | nonclustered] INDEX index\_name

ON table\_name (column\_name1 [, column\_name2 [, ...n ]])

SELECT [ALL | DISTINCT] column\_name1 [, column\_name2 [, ...n ]][FROM table\_name1 [, table\_name2 [, ...n ]][WHERE clause][GROUP BY clause][HAVING clause][ORDER BY clause]

SELECT fields

FROM table\_name1 {INNER | {RIGHT | LEFT} OUTER} JOIN table\_name2

ON table\_name1.join\_attribute = table\_name2.join\_attribute

SELECT fields FROM table\_name1, table\_name2

WHERE table\_name1.join\_attribute = table\_name2.join\_attribute

Syntax Reference

Insert

INSERT [INTO] table\_name

[(column list)]

{VALUES

({DEFAULT | NULL | expression}, . . . )

|

SELECT…}

Update

UPDATE table\_name

SET column = expression [ , column = expression . . . ]

[WHERE…]

Delete

DELETE [FROM] table\_name

[FROM…]

[WHERE…]

DATE Functions

GETDATE () returns the system date

DATEADD (**xx**, n, date1) returns a new date adding the number of **xx** to date1

DATEDIFF (**xx**, date1, date2) returns the number of **xx** from date1 (older) to date2 (newer)

DATENAME (**xx**, date1) returns a string representation of the **xx** of date1

DATEPART (**xx**, date1) returns an integer representation of the **xx** of date1

DATENAME (**xx**, date1) returns a string representation of the **xx** of date1

NOTE: xx represents the Datepart from the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Datepart | Abbreviation | Minimum | Maximum |
| Year | yy | 1753 | 9999 |
| Quarter | qq | 1 | 4 |
| Month | mm | 1 | 12 |
| Week | wk | 1 | 53 |
| Day of year | dy | 1 | 366 |
| WeekDay | dw | 1 (Sun.) | 7 (Sat.) |
| Day | dd | 1 | 31 |
| Hour | hh | 0 | 23 |
| Minute | mi | 0 | 59 |
| Second | ss | 0 | 59 |
| Millisecond | ms | 0 | 999 |

STRING Functions

**LEN (column|expression)** returns the number of characters (excluding trailing blanks).

**LEFT (column|expression, xx)** returns part of a character string starting from the left returning **xx** number of characters. Can NOT use negative numbers.

**RIGHT (column|expression, xx)** returns part of a character string starting from the right returning **xx** number of characters. Can NOT use negative numbers.

**SUBSTRING (column|expression, start, length) r**eturns part of a character string. The second parameter indicates at what character the substring starts. The third parameter indicates the number of characters to return. Can NOT use negative numbers

**REVERSE** **(column|expression)** returns a character string in reverse order.

**LTRIM** **(column|expression)** removes leading blanks.

**RTRIM** **(column|expression)** removes trailing blanks.

**UPPER** **(column|expression)** converts all characters to uppercase.

**LOWER** **(column|expression)** converts all characters to lowercase.