

# SQLSERVER

## Quick Reference

### SQLSERVER

#### Conventions

UPPERCASE LETTERS	Denote keywords, with you must enter as shown. (You can use lowercase letters.)
<i>italics</i>	Denote terms for which you must substitute identifiers or expressions.
[ ]	Denote options. Vertical bars separate the options. Brackets ([ ]) mean you can, but are not required to, choose one option.
{ }	Denote options. Vertical bars separate the options. Braces ({ }) mean you must choose one option.
	Denotes a choice among several options.
. . .	Denotes optional repetition of the previous clause.
<u>UNDERLINE</u>	Denotes the default option.

#### Data Types

BIGINT	DATETIME
INT	SMALLDATETIME
SMALLINT	DATE
TINYINT	TIME
BIT	CHAR
DECIMAL	VARCHAR
NUMERIC	TEXT
MONEY	BINARY
SMALLMONEY	NCHAR
FLOAT	NVARCHAR
SMALLFLOAT	NTEXT
REAL	IMAGE

## Query Language

### Commands

CREATE VIEW *view-name* [(*column-list*)]  
AS *SELECT-statement*  
[WITH CHECK OPTION]

DELETE FROM *table-name*  
[WHERE *condition*]

DROP VIEW *view-name*

INSERT INTO *table-name* [(*column-list*)]  
{VALUES (*value-list*) | *SELECT-statement*}

SELECT <Ver sección de Sent. SELECT>

UPDATE *table-name* SET  
*column-name* = *expr* [, ...]  
[WHERE *condition*]

### SELECT Statement

SELECT [ALL | [DISTINCT]]  
*select-list*  
FROM  
{  
*table-name* [*table-alias*]  
|  
OUTER *table-name* [*table-alias*]  
} [, ...]  
[WHERE *condition*]  
[GROUP BY *column-list*]  
[HAVING *group-condition*]  
[ORDER BY *column-name* [ASC | DESC][, ...]]

*SELECT-statement* UNION [ALL]  
*SELECT-statement*  
[UNION [ALL] *SELECT-statement*] ...

### Conditions

*expr rel-op expr*  
*expr* [NOT] BETWEEN *expr* AND *expr*  
*expr* [NOT] IN({*value-list* | *SELECT-statement*})  
*column-name* [NOT] LIKE “*string*”

*column-name* IS [NOT] NULL

*column-name* {ALL | [ANY | SOME]}  
(*SELECT-statement*)

[NOT] EXISTS (*SELECT-statement*)

[NOT] *condition*

*condition* {AND | OR} *condition*

## Functions

*CURRENT\_TIMESTAMP*

*CURRENT\_USER*

*DAY*

*GETDATE*

*LENGTH*

*MONTH*

*SUBSTR*

*TODAY*

*USER\_NAME*

*WEEKDAY (dtime-expr)*

*YEAR*

## Aggregate Functions

*AVG( {[DISTINCT] column-name } )*

*COUNT( {[DISTINCT] column-name | \*} )*

*MAX( {[DISTINCT] column-name} )*

*MIN( {[DISTINCT] column-name} )*

*SUM( {[DISTINCT] column-name} )*

## Transact-SQL

Ejecutar un Procedure

```
EXEC [UTE] [@ReturnStatus = ]
{procname[;number] | @ProcNameVar }
[[@parameter = ]{value | @variable [OUTPUT] |
[DEFAULT]] [WITH RECOMPILE]
```

Transacciones

<b>BEGIN TRANSACTION</b>	Marca el comienzo de una transacción.
<b>COMMIT TRANSACTION</b>	Escribe de forma permanente todos los cambios realizados sobre los datos ejecutados desde el último BEGIN TRANSACTION pendiente.
<b>ROLLBACK TRANSACTION</b>	Cancela todos los cambios de datos desde el último BEGIN TRANSACTION pendiente.

Control de Flujo

Clausulas	Uso
<b>BEGIN/END</b>	Bloques de comandos
<b>GOTO</b>	Branch incondicional CACA
<b>IF/ELSE</b>	Branch condicional. Frecuentemente usado con BEGIN/END
<b>RETURN</b> [ (ReturnStatus) ]	Establece un valor de retorno de un Procedimiento.
<b>WAITFOR</b>	Pausa el procesamiento por una duració específica
<b>WHILE</b>	Loop Condicional. Frecuentemente usado con BEGIN/END
<b>CASE WHEN a THEN b</b> <b>WHEN c THEN d END</b>	Brach Conditional. BEGIN/END no es necesario.

Cursor

Clausula	Sintaxis
Declare	DECLARE CursorName CURSOR [LOCAL   GLOBAL] [FORWARD_ONLY   SCROLL] [STATIC   KEYSET   DYNAMIC] [READ_ONLY   SCROLL_LOCKS   OPTIMISTIC] FOR SelectStatement [FOR {READONLY   UPDATE (OF ColumnList)}]
Open	OPEN CursorName
Read 1	FETCH {FIRST   NEXT} FROM CursorName INTO VariableList
Positioned Update	UPDATE {table_name   view_name} SET {column=expression[,...]} WHERE CURRENT OF cursor_name
Positioned Delete	DELETE [FROM] {table_name   view_name} WHERE CURRENT OF cursor_name
Close 2	CLOSE CursorName
Deallocate 3	DEALLOCATE CursorName

Prefijos

Prefijo	Uso
@	Variable Local o Parametro
@@	Variable Global
#	Tabla temporal de sesión
##	Tabla temporal global

Object Reference¶

```
[[[server.]database.]owner.]object
```

Variables declaración y asignación¶

```
DECLARE @local_variable [AS] data_type [ = valor ]

SET @local_variable = [ valor | expresión]
SELECT @local_variable { = | += | -= | *= | /= |
%= | &= | ^= | |= } expresión
```