

Data Type	String Functions	Date Functions	Mathematical Functions (cont)
Exact Numerics	ASCII REPLICATE	DATEADD (datepart, number, date) SYSDATETIME()	LOG CEILING
bit decimal	CHAR REVERSE		ROUND FLOOR
tinyint money	CHARINDEX RIGHT		SQRT SQUARE
smallint numeric	DIFFERENCE RTRIM		SIGN
bigint	LEFT SOUNDEX	DATEDIFF (datepart, start, end) SYSUTCDATETIME()	
Approximate Numerics	LEN SPACE		Dateparts
float real	LOWER STR	DATENAME (datepart, date) SYSDATETIMEOFFSET()	Year yy, yyyy
Date and Time	LTRIM STUFF	DATEPART (datepart, date) SWITCH-OFFSET (DATETIME-OFFSET, time_zone)	Quarter qq, q
smalldatetime timestamp	NCHAR SUBSTRING		Month mm, m
datetime	PATINDEX UNICODE		Day of Year dy, y
Strings	REPLACE UPPER		Day dd, d
char text	QUOTENAME		Week wk, ww
varchar			Hour hh
Unicode Strings	Type Conversion	DAY (date) ISDATE (expression)	Minute mi, n
nchar ntext	CAST (expression AS datatype)	GETDATE() MONTH (date)	Second ss, s
nvarchar	CONVERT (datatype, expression, [style])	GETUTCDATE() YEAR (date)	Millisecond ms
Binary Strings	Grouping (Aggregate) Functions		Object Operation
binary image	AVG MAX	Mathematical Functions	Stored Procedure
varbinary	BINARY_CHECKSUM MIN	ABS LOG10	CREATE PROCEDURE <name> AS <sql_statement>
Miscellaneous	CHECKSUM SUM	ACOS PI	Views
cursor table	CHECKSUM_AVG STDEV	ASIN POWER	CREATE VIEW<name>[(-<Column>, ...)]
sql_variant xml	COUNT STDEVP	ATAN RADIANS	AS <SELECT_statement>
	COUNT_BIG VAR	ATN2 RAND	Triggers
	GROUPING VARP	COT SIN	
		TAN EXP	
		COS DEGREES	



Object Operation (cont) <pre>> CREATE TRIGGER <name> ON <table> FOR INSERT, UPDATE, DELETE AS <sql_statement> Functions CREATE FUNCTION <name> RETURNS <data_type> AS BEGIN <sql_statement> RETURN <sql_expression> END</pre>	Ranking Functions <pre>RANK NTILE DENSE ROW_NUMBER RANK</pre>	Create an Index <pre>CREATE UNIQUE INDEX name ON table (columns)</pre>	Select Statement Construct <pre>SELECT [DISTINCT] [(TOP int TOP int PERCENT)] Column list [INTO new_table] FROM table_source [[[INNER {{ LEFT RIGHT FULL}}[OUT ER]]] JOIN CROSS APPLY] table_source2 ON table_source.pr - ima ry_key = table_source2.f ore - ign _ke y[,...n] [WHERE search _co ndi - tion] [GROUP BY group_by_ - exp res sion] [HAVING search _co ndi - tion] [ORDER BY order_ exp - ression [ASC DESC]]</pre>
Create a Stored Procedure <pre>CREATE PROCEDURE name @variable AS datatype = value AS -- Comments SELECT * FROM table GO</pre>	Foreign Key Construct <pre>ALTER TABLE <table1> WITH CHECK ADD CONSTRAINT <co nst - rai ntN ame> FOREIGN KEY (<t abl - e1c ol1 >) REFERENCES <ta ble 2> (<t abl e2c ol2 >)</pre>	T-SQL Statements UPDATE Statement <pre>UPDATE table_name SET column _na me= (ex - pre ssion DEFAULT NULL [,...n] [WHERE <se arc h_c ond - iti on>]</pre> DELETE Statement <pre>DELETE [FROM] table_name [WHERE <se arc h_c ond - iti on>]</pre> INSERT Statement <pre>INSERT [INTO] table_name [(colu mn_ list)] VALUES ((DEFAULT NULL expression 1[,...n])</pre>	
	Drop Constraint <pre>ALTER TABLE <tablename> DROP CONSTRAINT <co nst rai - ntN ame></pre>		
Table Functions <pre>ALTER DROP CREATE TRUNCATE</pre>	Primary Key construct <pre>ALTER TABLE <tablename> ADDCONSTRAINT <constraintName> PRIMARY KEY CLUSTERED (column n_List)</pre>		