

Data Type	
Exact Numerics	
bit	decimal
tinyint	money
smallint	numeric
bigint	
Approximate Numerics	
float	real
Date and Time	
smalldatetime	timestamp
datetime	
Strings	
char	text
varchar	
Unicode Strings	
nchar	ntext
nvarchar	
Binary Strings	
binary	image
varbinary	
Miscellaneous	
cursor	table
sql_variant	xml

String Functions	
ASCII	REPLICATE
CHAR	REVERSE
CHARINDEX	RIGHT
DIFFERENCE	RTRIM
LEFT	SOUNDEX
LEN	SPACE
LOWER	STR
LTRIM	STUFF
NCHAR	SUBSTRING
PATINDEX	UNICODE
REPLACE	UPPER
QUOTENAME	

Type Conversion
CAST (expression AS datatype)
CONVERT (datatype, expression, [style])

Grouping (Aggregate) Functions	
AVG	MAX
BINARY_CHECKSUM	MIN
CHECKSUM	SUM
CHECKSUM_AVG	STDEV
COUNT	STDEVP
COUNT_BIG	VAR
GROUPING	VARP

Date Functions	
DATEADD (datepart, number, date)	SYSDATETIME()
DATEDIFF (datepart, start, end)	SYSUTCDATETIME()
DATENAME (datepart, date)	SYSDATETIMEOFFSET()
DATEPART (datepart, date)	SWITCHOFFSET (DATETIMEOFFSET,time_zone)
DAY (date)	ISDATE (expression)
GETDATE()	MONTH (date)
GETUTCDATE()	YEAR (date)

Mathematical Functions	
ABS	LOG10
ACOS	PI
ASIN	POWER
ATAN	RADIANS
ATN2	RAND
COT	SIN
TAN	EXP
COS	DEGREES
LOG	CEILING
ROUND	FLOOR
SQRT	SQUARE
SIGN	

Dateparts	
Year	yy, yyyy
Quarter	qq, q
Month	mm, m
Day of Year	dy, y
Day	dd, d
Week	wk, ww
Hour	hh
Minute	mi, n
Second	ss, s
Millisecond	ms

Object Operation
Stored Procedure CREATE PROCEDURE <name> AS <sql_statement>
Views CREATE VIEW<name>[(<Column>, ...)] AS <SELECT_statement>
Triggers 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

Create a Stored Procedure
CREATE PROCEDURE name @variable AS datatype = value AS -- Comments SELECT * FROM table GO

Table Functions	
ALTER	DROP
CREATE	TRUNCATE

Ranking Functions	
RANK	NTILE
DENSE RANK	ROW_NUMBER

Foreign Key Construct

```
ALTER TABLE <table1> WITH CHECK
ADD CONSTRAINT <constraintName>
FOREIGN KEY (<table1col1>)
REFERENCES <table2> (<table2col2>)
```

Drop Constraint

```
ALTER TABLE <tablename> DROP
CONSTRAINT <constraintName>
```

Primary Key construct

```
ALTER TABLE <tablename>
ADDCONSTRAINT <constraintName>
PRIMARY KEY CLUSTERED
(column_List)
```

Create an Index

```
CREATE UNIQUE INDEX name
ON
table (columns)
```

T-SQL Statements

UPDATE Statement

```
UPDATE table_name
SET column_name=(expression |
DEFAULT | NULL [,...n]
[WHERE <search_condition>])
```

DELETE Statement

```
DELETE [FROM] table_name
[WHERE <search_condition>]
```

INSERT Statement

```
INSERT [INTO] table_name
[(column_list)]
VALUES ((DEFAULT | NULL |
expression 1[,...n]))
```

Select Statement Construct

```
SELECT [DISTINCT] [(TOP int | TOP
int PERCENT)]
Column list
[INTO new_table]
FROM table_source
[[[INNER | {{ LEFT | RIGHT | FULL}
[OUTER]}}] JOIN | CROSS APPLY]
table_source2
ON table_source.primary_key =
table_source2.foreign_key[,...n]
[WHERE search_condition]
[GROUP BY group_by_expression]
[HAVING search_condition]
[ORDER BY order_expression [ASC |
DESC]]
```