SSMS Shortcuts

Display the context menu	- SHIFT+F10
Display the Query Designer	- CTRL+SHIFT+Q
Toggle full screen mode	- SHIFT+ALT+ENTER
Cycle through Child windows	- CTRL+SHIFT+TAB
Display Object Explorer	- F8
Display the Summary Window	- F7
Set or remove a bookmark	- CTRL+K
Save all	- CTRL+SHIFT+S
Complete word	- ALT+RIGHT ARROW

Type Conversion

CAST (<<expression>> AS datatype)
CONVERT (datatype, expression [, style])

Date time function

GETDATE()	GETUTCDATE()
SYSDATETIME()	SYSDATETIMEOFFSET()
SYSUTCDATETIME()	DATENAME (datepart, date)
DATEPART (datepart, date)	DAY (date)
MONTH (date)	YEAR (date)
DATEDIFF (datepart, startdate, enddate)	
DATEADD (datepart, number, date)	
SWITCHOFFSET (DATETIMEOFFSET, time_zone)	
ISDATE (expression)	

Datepart extensions

Year	- yy, yyyy
quarter	- qq, q
month	- mm, m
dayofyear	- dy, y
day	- dd, d
week	- wk, ww
weekday	- dw
hour	- hh
minute	- mi, n
second	- ss, s
millisecond	- ms
microsecond	- mcs
nanosecond	- ns
TZoffset	- tz
ISO_WEEK	- isowk, isoww

Date& Time datatype

time
date
smalldatetime
datetime
datetime2
datetimeoffset

Index Creation construct

CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name
ON <object>(column [ASC | DESC] [,...n])
[INCLUDE (column_name [,...n])]
[WHERE<filter_predicate>]
[ON { partition_scheme_name (column_name)
    | filegroup_name | default }]
[FILESTREAM_ON
    { filestream_filegroup_name | partition_scheme_name | "NULL" }]

Index Creation construct

ALTER INDEX { index_name | ALL } ON <object>
{ REBUILD
  | DISABLE | REORGANIZE
[WITH (LOB_COMPACTION = { ON | OFF })]
  }

ColumnStore Index Construct

CREATE [NONCLUSTERED] COLUMNSTORE INDEX index_name
  ON <object>(column [,...n])

Primary Key construct

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

New SQL 2012 Function

PARSE
    (string_value AS data_type [USING culture])
TRY_CONVERT
    (data_type [(length)], expression [, style])
TRY_PARSE
    (string_value AS data_type [USING culture])
DATEFROMPARTS
    (year, month, day)
DATETIME2FROMPARTS
    (year, month, day, hour, minute, seconds, fractions, precision)
DATETIMEFROMPARTS
    (year, month, day, hour, minute, seconds, milliseconds)
DATETIMEOFFSETFROMPARTS
    (year, month, day, hour, minute, seconds, fractions, hour_offset, minute_offset, precision)
EOMONTH (start_date
    [, month_to_add])
SMALLDATETIMEFROMPARTS
    (year, month, day, hour, minute)
TIMEFROMPARTS
    (hour, minute, seconds, fractions, precision)
CHOOSE
    (index, val_1, val_2 [, val_n])
IIF
    (boolean_expression, true_value, false_value)
CONCAT
    (string_value1, string_value2 [, string_valueN])
FORMAT
    (value, format [, culture])

Ranking functions

RANK ( ) OVER ([partition_by_clause] order_by_clause)
DENSE_RANK ( ) OVER ([<partition_by_clause>] <order_by_clause>)
NTILE (integer_expression) OVER ([<partition_by_clause>] <order_by_clause>)
ROW_NUMBER ( )   OVER ([PARTITION BY value_expression , .. [n]] order_by_clause)

SQL 2012 Analytic Functions

CUME_DIST( )
OVER ([partition_by_clause] order_by_clause)
FIRST_VALUE ([scalar_expression)
OVER ([partition_by_clause] order_by_clause [rows_range_clause])
LAG (scalar_expression [,offset] [,default])
OVER ([partition_by_clause] order_by_clause)
LAST_VALUE ([scalar_expression)
OVER ([partition_by_clause] order_by_clause rows_range_clause)
LEAD (scalar_expression [,offset] , [default])
OVER ([partition_by_clause])
PERCENTILE_CONT (numeric_literal) WITHIN GROUP
(ORDER BY order_by_expression [ASC | DESC]) OVER ([<partition_by_clause>])
PERCENTILE_DISC (numeric_literal) WITHIN GROUP
(ORDER BY order_by_expression [ASC | DESC]) OVER ([<partition_by_clause>])
PERCENT_RANK( )
OVER ([partition_by_clause] order_by_clause)

T-SQL Statements

UPDATE Statement
    UPDATE tabLe_name
    SET coLumn_name = (expression I
    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])

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

Foreign Key construct

ALTER TABLE <tabte1> WITH CHECK
ADD CONSTRAINT <constraintName>
FOREIGN KEY (<table1col1>)
REFERENCES <table2> (<table2col2>)

Drop Constraint

ALTER TABLE <tablename> DROP
CONSTRAINT <constraintName>

Pseudo code for CTE

WITH cte_name(column_name [,...n])
AS
(CTE_query_definition – Anchor member
UNION ALL
CTE_query_definition –
Recursive member referencing cte_name)
SELECT *FROM cte_name

SELECT Statement construct

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

PIVOT and UNPIVOT Construct

SELECT <non-pivoted column>,
[pivoted column] AS <column name>, ...
FROM
(<SELECT query that produces the data>)
AS <alias for the source query>
PIVOT
(<aggregation function>(<column being aggregated>)
FOR
[<column that contains the values that will become column headers>]
    IN ([pivoted column], …) AS <alias for the pivot table>
<optional ORDER BY clause>;