Forgotten T-SQL Cheat Sheet

LOGICAL PROCESSING ORDER OF SELECT

```
1.  FROM       table
2.  ON         join condition
3.  JOIN       table
4.  WHERE      clauses
5.  GROUP BY   columns
6.  WITH CUBE / WITH ROLLUP
7.  HAVING     condition
8.  SELECT     columns
9.  DISTINCT
10. ORDER BY   columns
11. TOP        % or number
```

CTEs

```
; WITH cteName ( columnList )
AS ( SELECT statement )
SELECT   columns
FROM     cteName
INNER JOIN table ON condition
```

RECURSIVE CTEs

```
; WITH cteName ( columnList )
AS ( -- Anchor statement:
     SELECT columns FROM table…
     UNION ALL
     -- Recursion statement:
     SELECT columns FROM table…
     INNER JOIN cteName ON …
     )
SELECT   columns
FROM     cteName
```

OVER AND PARTITION BY

```
/* Aggregate functions include COUNT,
MIN, MAX, AVG, ROW_COUNT(), etc. */

SELECT
  agg_func(col1) OVER(),
  agg_func(col1)
          OVER(PARTITION BY col2),
  columns
FROM table…
```

XML TRICK: LIST OF DETAILS

```
/* Table2 holds detail
rows for Table1; e.g., order details to
order headers. */

SELECT  columns,
        colname = STUFF(
    ( SELECT  ','
      + Name
      FROM    Table2
      WHERE Table1.ID = Table2.ID
      ORDER BY Name
      FOR XML PATH('')
    ), 1, 1, '')
FROM    Table2
```

EXCEPT/INTERSECT

```
SELECT   col1, col2 FROM Table1
EXCEPT
SELECT   col3, col4 FROM Table2


SELECT   col1, col2 FROM Table1
INTERSECT
SELECT   col3, col4 FROM Table2
```

MERGE

```
DECLARE @Changes
    TABLE(Change VARCHAR(20))

; MERGE INTO DestTable
  USING
  ( SELECT from sourceTable
  ) AS Source ( columnList )
  ON DestTable.ID = Source.ID

  WHEN MATCHED THEN
      Action on destination
      /* E.g., UPDATE SET col1 = 1 */

  WHEN NOT MATCHED BY TARGET|SOURCE
      Action on destination
      /* E.g., INSERT (col1) VALUES(1) */

  OUTPUT $action INTO @Changes

SELECT  * FROM @Changes
```