Create New Database

```
CREATE DATABASE DB_Name
```

Rename Database

1) `ALTER DATABASE DB_Name modify name = New_DB_Name`

2) `EXEC sp_renameDB 'DB_Name','New_DB_Name'`

Delete Database

```
DROP DATABASE DBName
```

Create New Table

```
CREATE TABLE TN (CN1 Data_Type(size), CN2 Data_Type(size), ..... CN3
Data_Type(Size))
```

Rename Column properties (dsata type / size)

```
ALTER TABLE TN ALTER COLUMN CN New_Data_Type(New_Size)
```

Rename Column

```
Sp_Rename 't1.C1', 'C7', 'COLUMN'
```

Add new Column

```
ALTER TABLE TN ADD CN Data_Type(Size)
```

Delete Column

```
ALTER TABLE TN DROP COLUMN CN
```

Insert Data

1) `INSERT INTO TN (CN1, CN2,.. CNn) VALUES ('x', 'y', .. 'z')`

2) `INSERT INTO TN VALUES ('x', 'y', .. 'z')`

Retriving all rows & all columns

```
SELECT CN, * FROM TN
```

Updating the data in table

```
UPDATE TN SET CN1 = 'Value1', CN2 = 'Value2',.... CNn = 'Valuen'
WHERE CN4 = 'condition'
```

Retriving selected rows & all columns

```
SELECT * FROM TN WHERE condition(s)
```

Retriving selected columns & all rows

```
SELECT CN1, CN2,... CNn FROM TN
```

Retriving selected columns & all rows

```
SELECT CN1, CN2,... CNn FROM TN WHERE condition(s)
```

Retriving distinct (unique) data

```
SELECT DISTINCT CN1, CN2, ...CNn FROM TN
```

Retriving distinct (unique) data

```
SELECT DISTINCT CN1, CN2, ...CNn FROM TN
```

Select with AND / OR

```
SELECT Column_List, * FROM TN WHERE X >= 10 OR (Y < 5 AND Y >= 15)
```

Select with IN / Not IN

```
SELECT * FROM TN WHERE CN IN / NOT IN('abc', 'xyz')
```

Select with BETWEEN

```
SELECT * FROM TN WHERE CN BETWEEN '10' AND '100'
```

Select with ORDER BY

1) `SELECT CN4 FROM TN [WHERE condition] ORDER BY CN2 ASC / DESC`
2) `SELECT CN4, CN2, CN7, CN1 FROM TN [WHERE condition] ORDER BY 1, 3`

Select with LIKE

1) `SELECT * FROM TN WHERE CN LIKE '_ABC%'`
2) `SELECT CN1 FROM TN WHERE CN4 LIKE '%XY_'`

Select with AGGRIGATE FUNCTIONS

```
SELECT Function (CN) FROM TN
```

1) AVG : Average of all records in the column
2) COUNT : Number of records (rows) in the column
3) MAX : Maximum value in the column
4) MIN : Minimum value in the column
5) SUM : Sum of all records in the column

Column Name / Table Name Alias

1) `SELECT CN as CN_Alias FROM TN as TN_Alias`
2) `SELECT CN CN_Alias FROM TN TN_Alias`

NOT NULL constraint

```
CREATE TABLE TN (C1 int NOT NULL, C2 varchar(25) NOT NULL)
```

CHECK constraint

```
CREATE TABLE TN (C1 int NOT NULL CHECK (C1 > 0), C2 varchar(25) NOT NULL)
```

PRIMARY KEY constraint

```
create table TN1 (C1 int primary key, C2 Char(50), C3 Varchar(50), C4 int)
```

FOREIGN KEY constraint

```
Create Table TN2 (A1 int, A2 Char(50), C1 int FOREIGN KEY REFERENCES TN1 (C1))
```

Explicit INNOR JOIN (matching rows of both the tables)

```
SELECT TA.CA1, TA.CA5, TB.CB2, TB.CB7
FROM TA INNER JOIN TB
ON TA.CA1 = TB.CB1
```

Implicit INNOR JOIN (matching rows of all tables)

```
SELECT TA.CA1, TA.CA5, TB.CB2, TB.CB7, TB.CB4, TC.CC3
FROM TA, TB, TC
WHERE TA.CA1 = TB.CB1 AND TB.CB4 = TC.CC4
```

LEFT OUTER JOIN (All rows of TA and matched rows of TB)

```
SELECT TA.CA1, TA.CA5, TB.CB2, TB.CB7
FROM TA LEFT OUTER JOIN TB
ON TA.CA1 = TB.CB1
```

RIGHT OUTER JOIN (All rows of TB and matched rows of TA)

```
SELECT TA.CA1, TA.CA5, TB.CB2, TB.CB7
FROM TA RIGHT OUTER JOIN TB
ON TA.CA1 = TB.CB1
```

FULL OUTER JOIN (all (matched & unmatched) rows in both tables)

```
SELECT TA.CA1, TA.CA5, TB.CB2, TB.CB7
FROM TA FULL OUTER JOIN TB
ON TA.CA1 = TB.CB1
```

CROSS JOIN (TA rows * TB rews)

```
SELECT TA.CA1, TA.CA5, TB.CB2, TB.CB7
FROM TA CROSS JOIN TB
```

Concatination

```
Select CN1 + CN2 as CN_Alias From TN
```

Display length of string

```
SELECT LEN ('string')
```

Remove all white spaces from the beginning of string

```
SELECT LTRIM ('   string   ')
```

Remove all white spaces at the end of string

```
SELECT RTRIM ('   string   ')
```

Replace (in "CN", find where "Data" occurs, and replace it with "New_Data")

```
SELECT Replace (CN, 'Data', 'New-Data') from TN
```

Retrive current DB date & time

```
SELECT GETDATE()
```

DATEDIFF

Returns the number of date and time boundaries crossed between two specified dates.

Syntax : `SELECT DATEDIFF (datepart, end_date, start_date )`

EX. : `SELECT DATEDIFF (day, '02-20-2010', '03-30-2010')`

Datepart	Abbreviations
Year	yy, yyyy
quarter	qq, q
Month	mm, m
	dy, y
Day	dd, d
Week	wk, ww
Hour	hh
minute	mi, n
second	ss, s
millisecond	ms