

Syntax

SQL (;) .

Table , . 'ABC' 'abc'

SQL .

SQL , , .

, (),

(")

1 (--)

(/* */)

database, table, column , , .

' ' .

() . .

: SELECT → FROM → WHERE → GROUP BY → HAVING → ORDER BY

: FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

. Index Scan .

Summary

<< Backup >>

BACKUP **USE** tempDB;
BACKUP DATABASE ShopDB **TO DISK** = 'D:\sqlDB2019.bak' **WITH INIT**;

RESTORE **USE** tempDB;
RESTORE DATABASE ShopDB **FROM DISK** = 'D:\ShopDB.bak' **WITH REPLACE**;
USE ShopDB;

<< Database >>

CREATE **USE** tempdb;
CREATE DATABASE shopDB;

EXECUTE **EXEC** sp_helpdb;

USE **USE** shopDB;

DROP **USE** tempdb;
DROP DATABASE shopDB; -- Table Data .

<< Schema >> **CREATE SCHEMA** userSchema;
CREATE TABLE userSchema.userTbl (id CHAR(8));

<< Table >>

EXECUTE **EXEC** sp_tables @table_type = " ' TABLE' ";



Summary (cont)

```
CREATE USE shopDB;
CREATE TABLE userTbl (
    userID CHAR(8) NOT NULL PRIMARY KEY, -- CONSTRAINT PF_name PRIMARY KEY CLUSTERED
    name CHAR(8) NOT NULL UNIQUE,
    birthYear SMALLINT NOT NULL DEFAULT YEAR(GETDATE( )) CHECK (birthYear>1900),
    hobby NVARCHAR(10) SPARSE NULL);
```

```
CREATE TABLE buyTbl (
    num INT NOT NULL IDENTITY(1, 1),
    userID CHAR(8) NOT NULL FOREIGN KEY REFERENCES userTbl(userID),
    prodName NVARCHAR(20) NOTNULL,
    amount INT,
    CONSTRAINT PK_num PRIMARY KEY CLUSTERED (userID) ),
    CONSTRAINT CK_birthYear CHECK (birthYear>1900);
```

```
DROP DROP TABLE userTbl;
```

```
RENAME sp_rename 'userTbl', 'userTbl1';
```

<< Alter - Column >>

```
EXEC EXEC sp_help buyTbl;
```

```
ADD colname ALTER TABLE userTbl ADD weight SMALLINT NULL;
```

```
ALTER COLUMN ALTER TABLE userTbl ALTER COLUMN weight INT NULL; -- NOT NULL
```

```
DROP COLUMN ALTER TABLE userTbl DROP COLUMN weight; -- Data
```

```
RENAME EXEC sp_rename 'userTbl.weight', 'myWeight', 'COLUMN';
```

<< Alter - Constraint >>

```
ADD -- DEFAULT( ), CHECK / UNIQUE ( ),
CONSTRAINT
```

```
PRIMARY KEY ALTER TABLE userTbl ADD CONSTRAINT PK_userTbl_userID PRIMARY KEY (userID, name);
```

```
FOREIGN KEY ALTER TABLE buyTbl ADD CONSTRAINT FK_buyTbl_userTbl FOREIGN KEY (userID) REFERENCES userTbl(userID);
```

```
ON UPDATE ALTER TABLE buyTbl ADD CONSTRAINT FK_buyTbl_userTbl FOREIGN KEY (userID) REFERENCES userTbl(userID)
ON DELETE ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
UNIQUE ALTER TABLE userTbl ADD CONSTRAINT UN_name UNIQUE(name);
```

```
CHECK ALTER TABLE userTbl ADD CONSTRAINT CK_birth CHECK (birthYear >= 1900 AND birthYear <= YEAR(GETDATE-
()));
```



By **nollae93**

cheatography.com/nollae93/

Not published yet.

Last updated 7th September, 2020.

Page 2 of 33.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Summary (cont)

DEFAULT ALTER TABLE userTbl ADD CONSTRAINT DF_birthYear DEFAULT YEAR(GETDATE()) FOR birthYear;
 -- INSERT INTO userTbl VALUES (2016, DEFAULT, 195); -- DEFAULT .
 -- INSERT INTO userTbl(birtyYear, height) VALUES (2016, 195); -- .
 -- INSERT INTO userTbl VALUES (2016, N' ', 195); -- DEFAULT .

DROP CONSTRAINT ALTER TABLE buyTbl DROP CONSTRAINT FK_buyTbl_userTbl;
 ALTER TABLE userTbl DROP CONSTRAINT PK_userTbl_userID;
 Constraint DROP ADD CONSTRAINT .

NOCHECK CONSTRAINT ALTER TABLE userTbl NOCHECK CONSTRAINT CK_mobile; -- .

CHECK CONSTRAINT ALTER TABLE userTbl CHECK CONSTRAINT CK_mobile;

<< Index >>

CREATE INDEX - ON CREATE INDEX IDX_userTbl_addr ON userTbl(addr);
 CREATE UNIQUE INDEX IDX_userTbl_addr ON userTbl(addr);

DROP INDEX - ON DROP INDEX IDX_userTbl_addr ON userTbl; -- CREAT TABLE PRIMARY KEY / UNIQUE

EXEC EXEC sp_helpindex userTbl;

<< Identity >>

SET IDENTITY INSERT - ON CREATE TABLE identTbl (num INT NOT NULL IDENTITY, name CHAR(3));
 SET IDENTITY_INSERT identTbl ON;
 INSERT INTO identTbl(num, name) VALUES (10, 'CCC'); -- 'num' .

SET IDENTITY INSERT - OFF SET IDENTITY_INSERT identTbl OFF;

SELECT IDENT_CURRENT() SELECT IDENT_CURRENT('identTbl'); -- (')

SELECT @@IDENTITY SELECT @@IDENTITY; --

<< Insert >>

BEGIM TRAN BEGIN TRANSACTION;

INSERT INTO - VALUES INSERT INTO Goods VALUES ('0001', ' ', NULL, DEFAULT); -- NULL . DEFAULT .



Summary (cont)

```
INSERT INTO Goods(id, name) VALUES(0001, ' '); -- NOT NULL
```

```
INSERT INTO Goods(id, name) VALUES (0001, ' '), (0002, ' ');;
```

```
INSERT INTO - CREATE TABLE newTbl (e INT, f INT, g INT); --
SELECT        INSERT INTO newTbl SELECT a, b, c FROM oldTbl;
```

```
INSERT INTO newTbl(e, f) SELECT b, a FROM oldTbl;
```

```
INSERT INTO targetTbl SELECT * FROM updateTbl WHERE NOT EXISTS (SELECT a FROM targetTbl WHERE target-
Tbl.a = updateTbl.a);
```

```
SAVE TRAN     SAVE TRANSACTION;
```

```
ROLLBACK TRAN ROLLBACK TRANSACTION; COMMIT . CHECK . ROLL BACK .
```

```
COMMIT TRAN   COMMIT TRANSACTION;
```

<< Update >>

```
UPDATE - SET  BEGIN TRY
              BEGIN TRAN
                UPDATE userTbl SET mData = '0000-00-00'; --
                UPDATE userTbl SET height = height * 0.01, mData = NULL WHERE addr = ' ';
                -- NULL CLEAR : NOT NULL
              COMMIT TRAN
            END TRY
            BEGIN CATCH
              ROLLBACK TRAN
              SELECT ERROR_NUM( )
              SELECT ERROR_MESSAGE( )
            END CATCH
```

<< Delete >>

```
TRUNCATE TABLE TRUNCATE TABLE buyTbl; -- .
```

```
DELETE FROM     DELETE FROM userTbl; -- . . .
```

```
DELETE FROM userTbl WHERE addr = ' ';
```

```
DELETE TOP(3) FROM userTbl WHERE addr = ' ' ; -- 3 .
```



Summary (cont)

```
DELETE FROM userTbl WHERE userID IN (SELECT TOP(2) userID FROM userTbl WHERE addr = ' ' ORDER BY height);
-- .DELETE -WHERE SELECT KEY .
```

DataBase

< BACKUP >

```
BACKUP USE tempDB; DATABASE . .
BACKUP DATABASE ShopDB TO DISK = 'D:\sqlDB2019.bak' WITH INIT;
```

```
RESTORE USE tempDB;
RESTORE DATABASE ShopDB FROM DISK = 'D:\ShopDB.bak' WITH REPLACE;
USE ShopDB;
```

< EXECUTE >

```
EXECUTE sp_helpdb; SQLServer
```

```
EXECUTE sp_help Goods;
```

```
EXECUTE sp_depends userTbl; Table
```

```
EXECCUTE sp_helpindex userTbl; index
```

```
SELECT OBJECT_NAME(object_id) as [ ], definition FROM sys.sql_modules; View / Procedure
```

```
SHOW SHWO DATABASES; MySQL
```

```
SHOW TABLES; MySQL
```

```
EXECUTE sp_tables @table_type = '' TABLE' '';
```

```
EXECUTE sp_columns @table_name = 'Department', @table_owner = 'HumanResources';
```

```
USE USE shop;
```

< Database >

```
CREATE DATABASE USE tempdb;
CREATE DATABASE shop;
```

```
USE USE shopDB;
```

```
DROP USE tempdb;
DROP DATABASE shopDB;
```



DataBase (cont)

< Schema >

CREATE SCHEMA **CREATE SCHEMA** userSchema;

CREATE TABLE **CREATE TABLE** userSchema.userTbl (id INT);

SELECT **SELECT * FROM** userSchema.userTbl;

< View >

Syntax Table NOT NULL Column INSERT .

JOIN View INSERT . INSTEAD OF TRIGGER .

VIEW WITH CHECK OPTION data Error .

VIEW Table . VIEW . EXEC sp_depends .

CREATE VIEW - AS **CREATE VIEW** v_userTbl **AS SELECT** userID, name, addr **FROM** userTbl

CREATE VIEW v_sum **AS SELECT** userID, **SUM**(price*amount) AS [total] **FROM** buyTbl **GROUP BY** userID;

- WITH CHECK OPTION **CREATE VIEW** v_height177 **AS SELECT * FROM** userTbl **WHERE** height >= 177 **WITH CHECK OPTION;**

-- CAUTION : WITH CHECK OPTION INSERT height 177 data

- WITH ENCRYPTION **ALTER VIEW** v_userTbl **WITH ENCRYPTION AS SELECT** userID, name, addr **FROM** userTbl ()

SELECT **SELECT * FROM** v_userTbl;



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 6 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

DataBase (cont)

```

View      SELECT
ALTER VIEW - AS      ALTER VIEW v_suerTbl AS SELECT userID AS [ ], name AS [ ], addr FROM userTbl
DROP VIEW            DROP VIEW v_userTbl;
                    SELECT OBJECT_NAME(object_id) as [ ], definition FROM sys.sql_modules;
                    EXEC sp_depends userTbl;
UPDATE              UPDATE v_userTbl SET addr = N' ' WHERE userID = 'JKW';
INSERT              INSERT INTO v_userTbl (userID, name, addr) VALUES ('KBM', ' ', ' ');
                    -- userTbl birthYear NOT NULL      Error      . NOT NULL NULL      .      INSERT      .
    
```

DDL - Data Definition Language

< Create >

```

Syntex  NULL      . NULL      SPARSE NULL      . NULL      . 60%      . SELECT      Table      .
        '#'      tempdb      .      .      '##'      .
    
```



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 7 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

DDL - Data Definition Language (cont)

NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK 5 .

```
CREATE TABLE Goods
( id CHAR(4) NOT NULL PRIMARY KEY, name VARCHAR(100) NOT NULL, price INTEGER NULL DEFAULT 0, date DATE NULL);
```

```
CREATE TABLE Goods
( num INT IDENTITY NOT NULL, id CHAR(8) NOT NULL FOREIGN KEY REFERENCES Goods(id), amount SMALLINT NULL);
```

```
- SPARSE NULL
CREATE TABLE userTbl (name NCHAR(8) NOT NULL, hobby NVARCHAR(10) SPARSE NULL);
```

< PRIMARY KEY >

```
CREATE TABLE userTbl ( userID CHAR(8) NOT NULL )
```

```
- CREATE TABLE userTbl ( userID CHAR(8) NOT NULL PRIMARY KEY);
```

```
- CREATE TABLE userTbl ( userID CHAR(8) NOT NULL CONSTRAINT PK_useTbl_userID PRIMARY KEY);
```

```
- CREATE TABLE userTbl ( userID CHAR(8) NOT NULL, CONSTRAINT PK_useTbl_userID PRIMARY KEY( userID ));
```

```
ALTER - ADD ALTER TABLE userTbl ADD CONSTRAINT PK_useTbl_userID PRIMARY KEY(userID );
```

```
CREATE TABLE prodTbl ( prodCode CHAR(3) NOT NULL, prodID CHAR(4) NOT NULL,
CONSTRAINT PK_prodTbl_prodCode_prodID PRIMARY KEY( prodCode, prodID ) )
```



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 8 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

DDL - Data Definition Language (cont)

```
ALTER TABLE prodTbl ADD CONSTRAINT PK_prodTbl_prodCode_prodID PRIMARY KEY(prodCode, prodID);
```

< FOREIGN KEY >

```
CREATE PRIMARY / UNIQUE KEY .
```

```
- CREATE TABLE buyTbl (userID CHAR(8) NOT NULL FOREIGN KEY REFERENCES userTbl(userID));
```

```
- CREATE TABLE buyTbl
(userID CHAR(8) NOT NULL CONSTRAINT FK_userTbl_buyTbl FOREIGN KEY REFERENCES userTbl(userID));
```

```
- CREATE TABLE buyTbl (userID CHAR(8) NOT NULL,
CONSTRAINT FK_userTbl_buyTbl FOREIGN KEY(userID) REFERENCES userTbl(userID));
```

```
ALTER - ADD ALTER TABLE buyTbl ADD CONSTRAINT FK_userTbl_buyTbl FOREIGN KEY(userID) REFERENCES userTbl(userID);
```

```
- ON UPDATE ALTER TABLE buyTbl ADD CONSTRAINT FK_userTbl_buyTbl FOREIGN KEY (userID) REFERENCE userTbl(u- PK
CASCADE serID) ON UPDATE CASCADE;
```

```
- ON DELETE ALTER TABLE buyTbl DROP CONSTRAINT FK_userTbl_buyTbl;
```

```
RESTRICT ALTER TABLE buyTbl ADD CONSTRAINT FK_userTbl_buyTbl FOREIGN KEY (userID) REFERENCE userTbl(u-
serID) ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
--CASCADE / SET NULL / SET DEFAULT / NO ACTION / RESTRICT
```

```
SSMS - Table - - Table sheet - - - - ... - ( / ) & ( / )
```

< UNIQUE >

```
CREATE . NULL
```



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 9 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

DDL - Data Definition Language (cont)

-	CREATE TABLE userTbl (userID CHAR(8) NOT NULL PRIMARY KEY , email CHAR(30) NULL UNIQUE);
-	CREATE TABLE userTbl (userID CHAR(8) NOT NULL PRIMARY KEY , email CHAR(30) NULL CONSTRAINT AK_email UNIQUE);
-	CREATE TABLE userTbl (userID CHAR(8) NOT NULL PRIMARY KEY , email CHAR(30) NULL, CONSTRAINT AK_email UNIQUE (email));
ALTET - ADD	ALTER TABLE userTbl ADD CONSTRAINT UN_name UNIQUE (name);
< CHECK >	
CREATE	CREATE TABLE userID (birthYear NULL CHECK (birthYear>1900));
	CREATE TABLE userID (birthYear NULL, CONSTRAINT CK_birthYear CHECK (birthYear>1900));
ALTER - ADD	ALTER TABLE userTbl ADD CONSTRAINT CK_birh CHECK (birthYear >= 1900 AND birthYear <= YEAR(GETDATE()));
	ALTER TABLE userTbl ADD CONSTRAINT CK_mobile1 CHECK (mobile1 IN('010', '011', '016'));
	ALTER TABLE DocExc ADD ColumnD INT NULL CONSTRAINT CHK_ColumnD_DocExc CHECK (ColumnD > 10 AND ColumnD < 50);
ALTER - WITH NOCHECK ADD	ALTER TABLE userTbl WITH NOCHECK ADD CONSTRAINT CK_mobile CHECK (mobile1 IN('010', '011', '016'));
	-- WITH NOCHECK CONSTRAINT .
< DEFALUT >	
CREATE	CREATE TABLE userTbl (birthYear INT NOT NULL DEFAULT YEAR(GETDATE()), addr NCHAR(2) NOT NULL DEFALUT N' ', height SMALLINT DEFAULT 170);
ALTER	ALTER TABLE userTbl ADD CONSTRAINT DF_birthYear DEFAULT YEAR(GETDATE()) FOR birthYear;
INSERT	INSERT INTO userTbl VALUES (2016, DEFALUT, 195);



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 10 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

DDL - Data Definition Language (cont)

INSERT INTO userTbl(birthYear, height) **VALUES** (2016, 195);

INSERT INTO userTbl **VALUES** (2016, N' ', 195);

< IDENTITY >

(Server) .INSERT . NOT NULL .

INT IDENTITY

INT IDENTITY(1, 2) 1, 2

SSMS TABLE - - ID '' PRIMARY KEY ID .(num)

- SEQUENCE IDENTITY SEQUENCE . . Transact-SQL (5) 20:00

< > -----

- CHAR(5)/VARCHAR(10) / VARCHAR(max) () 5 / 10 / 8

NCHAR(5) / NVARCHAR(10) / NVARCHAR(max) 5 / 10 / 4

VARBINARY(max)

CAST(@MovieScript AS NVARCHAR(MAX)) 2GB

- BIT 0 or 1

TINYINT / SMALLINT / INT / BIGINT +255 / ±3.2 (+ 6.5) / ±21 (+42) /

DECIMAL(5,2) 5 2

FLOAT / DOUBLE /

- DATE/TIME/DATETIME2

- CURSOR T-SQL

TABLE . .

XML XML . 2GB

< Rename > -----



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 11 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

DDL - Data Definition Language (cont)

RENAME TABLE	sp_rename 'Gods', 'Goods';	SQL Server
	-- RENAME TABLE Gods to Goods;	MySQL
DROP TABLE	DROP TABLE Goods;	Table , FK .
< ALTER TABLE - Column >		
ADD <i>colname</i>	ALTER TABLE Goods ADD name_eng VARCHAR(100);	SQL Server Only
	-- SSMS drag .	
	-- ALTER TABLE Goods ADD COLUMN name_eng VARCHAR(100);	MySQL
ALTER COLUMN	ALTER TABLE userTbl ALTER COLUMN hobby NVARCHAR(10) NOT NULL;	
	-- , . , . . NULL NOT NULL ERROR. .	
	UPDATE userTbl SET hobby = ' ' WHERE hobby IS NULL;	
DROP COLUMN	ALTER TABLE Goods DROP COLUMN name_eng;	
EXEC	EXEC sp_rename 'userTbl.uesrID', 'ID', 'COLUMN';	
< ALTER TABLE - Constraint >		
ADD CONSTRAINT	ALTER TABLE userTbl ADD CONSTRAINT DF_birthYear DEFAULT YEAR(GETDATE()) FOR birthYear;	
DROP CONSTRAINT	ALTER TABLE userTbl DROP CONSTRAINT DF_birthYear;	
NOCHECK CONSTRAINT	ALTER TABLE buyTbl NOCHECK CONSTRAINT FK_userTbl_buyTbl;	FK .()
CHECK CONSTRAINT	ALTER TABLE buyTbl CHECK CONSTRAINT FK_userTbl_buyTbl;	CONSTRAINT
	Constraint DROP ADD CONSTRAINT .	

Index

Syntax



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 12 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Index (cont)

WHERE Index Index . WHERE INDEX .

WHERE . . INSERT / UPDATE /DELETE . INSERT .

INDEX SQL . . ()

Nonclustered Index . .

JOIN .

Index Clustered NonClustered .

Clustered db index , NonClustered Index Page db .

Clustered Index table . PRIMARY KEY UNIQUE .

Clustered Index , ORDER BY . .



By **nollae93**
cheatography.com/nollae93/

Not published yet.
Last updated 7th September, 2020.
Page 13 of 33.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Index (cont)

Insert Clustered index clustered index . PRIMARY KEY NONCLUSTERED Nonclustered

Clustered Insert, Alter Nonclustered

Index

Index

Primary Key Unique Index

Index CONSTRAINTS (col1, col2)

PRIMARY KEY CLUSTERED index . NONCLUSTERED

PRIMARY KEY NONCLUSTERED UNIQUE CLUSTERED

PRIMARY KEY Clustered Index UNIQUE Nonclustered Index

PRIMARY KEY UNIQUE index DROP INDEX . ALTER TABLE - DROP CONSTRAINT index

CREATE NONCLUSTERED
INDEX



By **nollae93**
cheatography.com/nollae93/

Not published yet.
Last updated 7th September, 2020.
Page 14 of 33.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Index (cont)

```

CREATE INDEX - ON      CREATE INDEX idx_userTbl_addr ON userTbl(addr);
CREATE UNIQUE INDEX idx_userTbl_name ON userTbl(name);
DROP INDEX PRIMARY KEY UNIQUE index DROP INDEX . ALTER TABLE - DROP CONSTRAINT index
DROP INDEX idx_userTbl_addr ON userTbl;
DROP INDEX userTbl.idx_userTbl_addr;
ALTER INDEX ALTER INDEX
EXEC sp_helpindex userTbl;
SSMS - - - SQL Server - - SET STATISTICS IO
  
```

Memory Table

```

. NONCLUSTERED PRIMARY KEY .
1. dataBase
2. dataBase - - - -
3. dataBase - - - - ( ) - ' ' DILESTREAM - ' ' 2 .
CREATE TABLE CREATE TABLE memoryTbl (a INT PRIMARY KEY NONCLUSTERED, b NCHAR(100)) WITH (MEMORY_OPTIMIZED=ON);
CREATE PROCEDURE CREATE PROCEDURE usp_diskInsert
@data NCHAR(100)
AS
DECLARE @I INT = 1;
WHILE @i <= 500
BEGIN
INSERT INTO dbo.diskTable(a, b) VALUES (@i, @data);
SET @I += 1;
END
  
```



Memory Table (cont)

```
CREATE PROCEDURE usp_memoryInsert
@data NCHAR(100)
WITH NATIVE_COMPLATION, SCHEMABINDING
AS
BEGIN ATOMIC WITH (TRANSACTION ISOLATION LEVEL=SNAPSHOT, LANGUAGE=N'Korea')
DECLARE @I INT = 1;
WHILE @i <= 500
BEGIN
    INSERT INTO dbo.diskTable(a, b) VALUES (@i, @data);
    SET @I += 1;
END
END

DECLARE @sendData nchar(100) = REPLICATE(N' ', 100);
EXECUTE usp_diskInsert @sendData
```

DML - Data Manipulation Language

BEGIN TRANSACTION	BEGIN TRANSACTION; / START TRANSACTION;	SQL Server / MySQL Only
INSERT INTO - VALUES	INSERT INTO Goods VALUES ('0001', ' ', 1000, '2019-09-20')	
	INSERT INTO Goods VALUES ('0001', ' ', NULL, '2019-09-20')	NULL
	INSERT INTO Goods VALUES ('0001', ' ', 1000, '2019-09-20'), ('0002', ' ', 500, '2009-02-03');	
	INSERT INTO Goods VALUES ('0001', ' ', DEFAULT, '2019-09-20')	DEFALUT
	--	
	INSERT INTO Goods(id, name) VALUES (0001, ' ');	
	-- NOT NULL	
INSERT INTO - SELECT - FROM	INSERT INTO NewTable (id, name, price) SELECT id, name, price FROM Goods;	
	--	
	INSERT INTO GoodClassify (classify, sum_price) SELECT classify, SUM(price) FROM Goods GROUP BY classify;	



DML - Data Manipulation Language (cont)

	-- INSERT SELECT WHERE, GROUP BY . ORDER BY	
	SELECT classify, sum_price INTO NewTable FROM Goods;	
	--	
SET IDENTITY_INSERT - OFF	SET IDENTITY_INSERT Tbl1 ON ; INSERT INTO Goods(num, id, name) VALUE (11, 0001, ' ');	num error .
SET IDENTITY_INSERT - ON	SET IDENTITY_INSERT Tbl1 OFF ;	OFF. 12 .
SELECT IDENT_CURRENT	SELECT IDENT_CURRENT ('Goods');	IDENTITY , (')
SELECT @@IDENTITY	SELECT @@IDENTITY	IDENTITY
COMMIT TRAN	COMMIT TRAN ;	SQL Server
	COMMIT ;	
UPDATE - SET	UPDATE Goods SET date = '2009-1-2';	. . .
UPDATE - SET - WHERE	UPDATE Goods SET peice = price * 10 WHERE classify = ' ';	
	UPDATE Goods SET date = NULL WHERE id = '0008';	NULL . NOT NULL
	UPDATE Goods SET peicd = price * 10, date = '2019-01-02' WHERE classify = ' ';	
TRUNCATE TABLE	TRUNCATE TABLE Goods;	Table () , DELETE
DELETE FROM	DELETE FROM Goods;	< > Table
DELETE FROM - WHERE	DELETE FROM Goods WHERE price >= 4000;	WHERE
DELETE TOP() FROM - WHERE	DELETE TOP (10) FROM Goods WHERE name = 'Kim';	.
	DELETE FROM Goods WHERE id IN (SELECT TOP (10) id FROM Goods ORDER BY price);	key .
MERGE	MERGE memberTBL AS M	
USING	USING changeTbl AS C	
ON	ON M.userID = C.userID	



By **nollae93**
cheatography.com/nollae93/

Not published yet.
Last updated 7th September, 2020.
Page 17 of 33.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

DML - Data Manipulation Language (cont)

```

WHEN NOT MATCHED AND changeType = ' ' THEN
.....INSERT(userID, name, addr) VALUES(C.userID, C.name, C.addr)

WHEN MATCHED AND changeType = ' ' THEN
.....UPDATE SET M.addr = C.addr

WHEN MATCHED AND changeType = ' ' THEN
.....DELETE;

```

WITH - Common Table Expression

```

Syntax          SELECT .

WITH - AS       WITH cte_Tbl1(addr, maxHeight) AS (SELECT addr, MAX(height) FROM Tbl1 GROUP BY addr)
.....SELECT AVG(maxHeight*1.0) AS ' ' FROM cte_Tbl1;

```

SELECT

```

Syntax          .

SELECT FROM WHERE ;

SELECT SELECT * FROM Goods;
SELECT id, name FROM Goods;

-- AS          SELECT id, name AS nm FROM Goods;
              SELECT id AS " ", name AS " " FROM Goods;
              SELECT ' ' AS " ", 10 AS num, '2009-02-24' AS date FROM Goods;          COLUMN
              - , num, date ' ', 10, '2009-02-24' .

-- DISTINCT   SELECT DISTINCT classify, date FROM Goods;

-- TOP()      SELECT TOP(10) CreditCardID FROM Sales WHERE Type = 'Vista' ORDER BY ExpYear;
              SELECT TOP(SELECT COUNT(*)/100 FROM Sales) CreditCardID FROM Sales WHERE Type = 'Vista' ORDER BY ExpYear;

```



SELECT (cont)

-- TOP() PERCENT	SELECT TOP(0.1) PERCENT CreditCardID FROM Sales WHERE Type = 'Vista' ORDER BY ExpYear;
-- TOP() WITH TIES	SELECT TOP(0.1) PERCENT WITH TIES CreditCardID FROM Sales WHERE Type = 'Vista' ORDER BY ExpYear;
-- INTO	TABLE . Primary Key, Foreign Key .
	SELECT * INTO Tbl2 FROM Tbl1;
	SELECT ID, name INTO Tbl2 FROM Tbl1;
FROM	SLEECT * FROM Sales TABLESAMPLE(5 ROWS) ;
-- TABLESAMPLE(5 ROWS)	SLEECT TOP(500) * FROM Sales TABLESAMPLES(5 PERCENT) ;

WHERE

LIKE	SELECT * FROM Sample WHERE srt LIKE '__ddd%';	
	-- str ' + ddd + ' .	
	<CAUTION> SELECT * FROM Sample WHERE srt = 'ddd%';	'=' .
IS NULL	SELECT * FROM Goods WHERE price IS NULL ;	price NULL column
IS NOT NULL	SELECT * FROM Goods WHERE price IS NOT NULL ;	price NULL column
AND	SELECT name, classify FROM Goods WHERE classify != ' ' AND price >= 1000;	FROM
OR	SELECT name, classify FROM Goods WHERE classify = ' ' OR price >= 1000;	
BETWEEN	SELECT * FROM Goods WHERE price BETWEEN 100 AND 1000	
IN(=or)	SELECT * FROM Goods WHERE price IN (320, 500)	320 or 500 column. NULL
NOR IN	SELECT * FROM Goods WHERE price NOT IN (320, 500)	320 500 column. NULL .
Subscript	SELECT name, Height FROM userTBL WHERE height > (SELECT height FROM userTBL WHERE Name = ' ');	



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 19 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

WHERE (cont)

```
-- ANY      SELECT name, height FROM userTBL WHERE height = ANY (SELECT height FROM userTBL WHERE add = ' ');
-- ALL      SELECT name, height FROM userTBL WHERE height > ALL (SELECT height FROM userTBL WHERE add = ' ');
-- IN       SELECT name, price FROM Goods WHERE id IN(SELECT id FROM Store WHERE store_id = ' ');
- EXISTS   SELECT name, price FROM Goods AS S WHERE EXISTS (SELECT * FROM Store AS TS WHERE TS.id = ' ' AND TS.id = S.id);
-- EXISTS
```

GROUP BY

```
Syntax      KEY NULL      ( )
SELECT      , , GROUP BY      . 1:1
SELECT
      sort      . order by
WHERE      GROUP BY      . - HAVING
GROUP BY    SELECT userID, SUM(amount) AS ' ' FROM Goods GROUP BY userID; NULL
            SELECT userID, SUM(amount*price) AS ' ' FROM Goods GROUP BY userID;
            SELECT userID, SUM(amount*price) AS ' ' FROM Goods WHERE classify = ' ' GROUP BY userID;
-- ROLLUP   SELECT classify, SUM(amount*price) AS ' ' FROM Goods GROUP BY ROLLUP(classify);
            SELECT userID, classify, SUM(amount*price) AS ' ' FROM Goods GROUP BY ROLLUP(classify, userID);
```



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 20 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

GROUP BY (cont)

```
-- classify userID ' ' classify ,
-- CUBE      SELECT productName, color, SUM(amount) AS ' ' FROM Tbl1 GROUP BY CUBE(color, productName);
--          , ,
-- GROUPING_ID      SELECT groupName, SUM(price*amount) AS ' ', GROUPING_ID(groupName) AS ' ' FROM Tbl1 GROUP BY ROLLUP(-
NG_ID      grouponName);
--          "1"
```

HAVING

```
SELECT → FROM → WHERE → GROUP BY → HAVING → ORDER BY
Syntax      WHERE ' ' , HAVING ' ' .
            KEY      HAVING WHERE .
            HAVING , , GROUP BY KEY .
HAVING      SELECT classify, COUNT(*) FROM Goods GROUP BY classify HAVING COUNT(*) = 2;
            SELECT classify, AVG(price*1.0) FROM Goods GROUP BY classify HAVING AVG(price*1.0) >= 2500;
            SELECT classify, AVG(price*1.0) FROM Goods WHERE store = ' ' GROUP BY classify HAVING AVG(price*1.0) >= 2500;
```

ORDER BY

```
Syntax      SELECT
            <CAUTION> .
            KEY NULL .
            SELECT .
            SELECT .
ORDER BY      SELECT id, name, price FROM Goods ORDER BY price DESC;
```



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 21 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

ORDER BY (cont)

SELECT id, name, price FROM Goods **ORDER BY** price DESC, id;

SELECT classify, COUNT(*) FROM Goods **GROUP BY** classify **ORDER BY** COUNT(*);

SELECT namd **AS** nm FROM Goods **ORDER BY** nm;

OFFSET / FETCH NEXT **SELECT** ID, nsme **FROM** userTBL **ORDER BY** birth **OFFST** 4 **ROWS** **FETCH** NEXT 3 **ROWS** 4 3
ONLY;

/ /

SELECT name, price * 2 **AS** "price_x2" FROM Goods;

+=

10%7 -> 3

<> SELECT name, price FROM Goods WHERE price <> 1000;

NOT SELECT name, price FROM Goods WHERE **NOT** price = 1000;

'1-3' < '2'

IS NULL SELECT name, price FROM Goods WHERE price IS NULL;

IS NOT NULL SELECT name, price FROM Goods WHERE price IS NOT NULL;

NULL NULL NULL .

NULL/0 NULL .

NULL .

< >

CAST() **SELECT** price, amount, **CAST**(**CAST**(price **AS** FLOAT)/amount **as** DECIMAL(10,2)) **AS** ' / ' **FROM** Tbl1;

-- . 2

CONVERT() **SELEC** ANG(**CONVERT**(FLOAT, amount)) **AS** ' ' **FROM** Tbl1;

STR() **SELECT** STR(123);

PARSE() **SELECT** **PARSE**('2019 9 9' **AS** DATE);

TRY_PARSE() **SELECT** **TRY_PARSE**('3.14' **AS** INT);

-- **PARSE**() **TRY_PARSE** Null .

+ '100'+200 -> '100200'

+ '100' + 200 -> 300

'100' + 200.0 -> 300.0

-> / DECIMAL .

-> ERROR

< >

STATISTICS SET STATISTICS TIME ON;



syntax

```

DECLARE - SET      DECLARE @myVar1 INT, @myVar2 DECIMAL(5,2), @myVar3 NCHAR(20);
                   DECLAER @point INT = 77;
                   SET @myVar1 = 5;
                   SET @myVar2 = 4.52;
                   SET @myVar3 = '  => ';
                   SET @ myVar1 = GETDATE( );
                   SELECT @myVar1 = HireDate FROM Tbl1 WHERE ID = 111;
                   SELECT @myVar1 + @myVar2;
                   SELECT @myVar3, Name from Tbl1 WHERE height > 180;
                   SELECT TOP(@myVar1) Name, height FROM Tbl1 ORDER BY height;
    
```

Table

```

Table      Table      Table      . CREATE TABLE #
.
DECLARE @tblVar TABLE (id CHAR(8), name NVARCHAR(10), addr NCHAR(2));
INSERT INTO @tblVar SELECT userID, name, addr FROM userTbl WHERE birthYear >= 1970;
SELECT * FROM @tblVar;
    
```

CREATE TABEL

```

CREATE TABLE #tempTbl (id CHAR(8), name NVARCHAR(10), addr NCHAR(2));
INSERT INTO #tempTbl SELECT userID, name, addr FROM userTbl WHERE birthYear >= 1970;
SELECT * FROM #tempTbl;
    
```

DateTime

GETDATE()	SELECT GETDATE();	now
SYSDATETIME ()	SELECT SYSDATETIME();	now()
DATEADD	SELECT DATEADD(day, 100, '2019/01/01');	100
	SELECT DATEADD(hour, 100, '2019/01/01');	100
DATEDIFF	SELECT DATEDIFF(week, GETDATE(), '2027/10/9');	DayOfWeek
DATEPART	SELECT DATEPART(year, GETDATE());	
MONTH	SELECT YEAR(GETDATE());	()
	-- year / month / week / hour / minute / second	
DATENAME	SELECT DATENAME(weekday, GETDATE());	DayOfWeek
DATEFROMPARTS	SELECT DATEFROMPARTS('2019', '10', '09');	
	-- TIMEFROMPARTS() / DATETIME2FROMPARTS()	
EOMONTH	SELECT EOMONTH('2019-01-03');	1
	SELECT EOMONTH(GETDATE(), 3);	+ 3



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 23 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

/ / Str		
< >		
ROUND	SELECT ROUND(153.246, 2);	(153.250)
	SELECT ROUND(153.246, -2);	(200)
RAND	SELECT RAND();	0~1
FLOOR	SELECT FLOOR(3.14);	(3)
CEILING	SELECT CEILING(3.14);	(4)
ABS	SELECT ABS(-100);	
SQRT	SELECT SQRT(10);	
POWER	SELECT POWER(3, 2);	
< >		
CHOOSE	SELECT CHOOSE(2, 'a', 'b', 'c', 'd');	
IIF	SELECT IIF(1>2, 'TRUE', 'FALSE');	, ,
< String >		
CHARINDEX	SELECT CHARINDEX('Server', 'SQL Server 2017');	-> 5
RIGHT / LEFT	SELECT RIGHT('SQL Server 2017', 4);	3 -> '2017'
SUBSTRING	SELECT SUBSTRING(' ', 3, 2);	3 2 -> ' '
LEN	SELECT LEN('SQL Server 2017');	-> 15
LOWER / UPPER	SELECT LOWER('ABDdef');	
LTRIM / RTRIM	SELECT LTRIM(' ');	-> ' '
REPLACE	SELECT REPLACE('SQL Server 2017', 'Server', ' ');	'SQL 2017'
REPLICATE	SELECT REPLICATE('SQL', 5);	'SQL'
REVERSE	SELECT REVERSE('SQL Server 2017');	7102 revreS LQS
SPACE	SELECT SPACE(5);	5
STR	SELECT STR(123);	'123'
STUFF	SELECT STUFF('SQL 2017', 5, 2, 'Server');	5 2 'Server' -> 'SQL Server 2017'
FORMAT	SELECT FORMAT(GETDATE(), 'dd/MM/yyyy');	'16-09-2017'

NULL . COUNT(*) NULL .		
SELECT, HAVING, ORDER BY .		



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 24 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

(cont)

GROUP BY

COUNT()	SELECT COUNT(*) FROM Goods;	NULL	Data
	SELECT COUNT(price) FROM Goods;	NULL	
-- COUNT(DISTINCT...)	SELECT COUNT(DISTINCT classify) FROM Goods;	COUNT	
COUNT_BIG()	SELECT COUNT_BIG(DISTINCT classify) FROM Goods;	BIGINT (21)	
SUM()	SELECT SUM(amount) FROM Goods;		
	SELECT classify, SUM(amount) AS ' ' GROUP BY classify;		
AVG()	SELECT AVG(price) FROM Goods;	NULL	Count
	<CAUTION> : Data . TRUNC .		
	SELECT AVG(price*1.0) FROM Goods;		
	SELECT AVR(CAST(price AS DECIMAL(10,6))) AS ' ' FROM Tbl1;	10	6
STDEV()	SELECT STDEV(return) FROM Stock;		
VAR()	SELECT VAR(return) FROM Stock;		
MAX(). MIN()	SELECT MAX(date), MIN(date) FROM Goods;		
	SELECT name, height FROM Tbl1 WHERE height = (SELECT MAX(height) FROM Tbl1) ;		
	-- Tbl1 height name, height		

/

ROW_NUMBER()	SELECT ROW_NUMBER() OVER (ORDER BY height DESC, name ASC) ' ', name, addr, height FROM Tbl;	1-2-3-4
	SELECT addr, ROW_NUMBER() OVER (PARTITION BY addr ORDER BY height DESC, name ASC)[], name, height FROM Tbl;	



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 25 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

/ (cont)

```
SELECT top(10) *
  FROM (SELECT *, ROW_NUMBER() OVER(PARTITION BY stockCode ORDER BY logDate DESC, logTime DESC) AS
rnum
  FROM LogMinute) AS l2
  WHERE rnum = 1;
```

RANK() SELECT **RANK()** OVER (**ORDER BY** height DESC)[], name, addr, height FROM Tbl; 1-2-2-4 1-2-2-4

DENSE_- SELECT **DENSE_RANK()** OVER (**ORDER BY** height DESC)[], name, addr, height FROM 1-2-2-3
RANK() Tbl;

NTILE() SELECT **NTILE(4)** OVER (**ORDER BY** height DESC)[], name, addr, height FROM Tbl; 4 (1,2 3)

LEAD() SELECT name, addr, height, height - (**LEAD**(height, 1, 0) OVER (**ORDER BY** height DESC)) ' ()
OVER ' FROM Tbl1;

LGR() OVER

FIRST_- SELECT addr, name, height, height - (**FIRST_VALUE**(height) OVER (**PARTITION BY** addr ()
VALUE **ORDER BY** height DESC)) AS ' ' FROM Tbl1;
OVER

CUME_DIST SELECT addr, name,height, (**CUME_DIST()** OVER (**PARTITION BY** addr **ORDER BY** height
DESC)) * 100 AS ' ' FROM Tbl1;

PERCENTIL- SELECT **DISTINCT** addr, **PERCENTILE_CONT(0.5)** WITHIN GROUP (**ORDER BY** height) WTHIN GROUP
E_CONT **OVER (PARTITION BY** addr) AS ' ' FROM Tbl1;

PERCENTIL-
E_DISC

PIVOT SELECT * FROM pivotTest **PIVOT** (SUM(amount) **FOR** season **IN**(' ', ' ', ' ', ' ')) AS resultPivot;

C

By **nollae93**
cheatography.com/nollae93/

Not published yet.
Last updated 7th September, 2020.
Page 26 of 33.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

JOIN

< INNER JOIN >

```
INNER JOIN          SELECT B.uesrID, U.name, B.prodName, U.addr FROM buyTbl B
                   .....INNER JOIN userTbl U ON B.userID = U.userID WHERE B.userid = 'JYP';           U.
-- DISTINCT        SELECT DISRINCT U.uesrID, U.name, U.addr, FROM userTbl U
                   .....INNER JOIN buyTbl B ON U.userID = B.userID ORDER BY U.userid;
-- WHERE EXISTS    SELECT U.userID, U.addr FROM userTbl U
                   .....WHERE EXISTS (SELECT * FROM buyTbl B WHERE U.userID = B.userID);
```

< OUTER JOIN >

```
-- LEFT OUTER JOIN SELECT U.uesrID, U.name, U.prodName, U.addr FROM userTbl U
                   .....LEFT OUTER JOIN buyTbl B ON U.userID = B.userID ORDER BY U.userID;           userTbl .
                   .....WHERE B.pridName = NULL;
-- RIGHT OUTER JOIN .....FROM buyTbl B RIGHT OUTER JOIN userTbl U ON...
                   .....RIGHT OUTER JOIN buyTbl B ON U.userID = B.userID ORDER BY U.userID;
-- FULL OUTER JOIN
```

< >

```
-- INNER JOIN      SELECT S.stdName, S.addr, C.clubName, C.roomNo FROM stdTbl S
                   .....INNER JOIN stdclubTbl SC ON S.stdname = SC.stdName           S SC JOIN
                   .....INNER JOIN clubTbl C ON SC.clubName = C.clubName           C JOIN
-- LEFT OUTER JOIN SELECT S.stdName, S.addr, C.clubName, C.roomNo FROM stdTbl S
                   ..... LEFT OUTERJOIN stdclubTbl SC ON S.stdname = SC.stdName
                   ..... LEFT OUTERJOIN clubTbl C ON SC.clubName = C.clubName
-- RIGHT OUTER JOIN SELECT S.stdName, S.addr, C.clubName, C.roomNo FROM stdTbl S
```



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 27 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

JOIN (cont)

..... **LEFT OUTERJOIN** stdclubTbl **SC ON** S.stdname = SC.stdName

..... **RIGHT OUTERJOIN** clubTbl **C ON** SC.clubName = C.clubName

-- FULL
OUTER JOIN **SELECT S.stdName, S.addr, C.clubName, C.roomNo FROM stdTbl S**

..... **FULL T OUTERJOIN** stdclubTbl **SC ON** S.stdname = SC.stdName

..... **FULL OUTERJOIN** clubTbl **C ON** SC.clubName = C.clubName

CROSS JOIN **SELECT * FROM buyTbl CROSS JOIN userTbl;**

-- 'ON'

SELF JOIN **SELECT A.emp AS ' ', B.emp AS ' ', B.department AS ' ' FROM empTbl A INNER JOIN empTbl B ON A.manager = B.emp WHERE A.emp = ' ';**

< / >

UNION (ALL) **SELECT stdName, addr FROM stdTbl UNION SELECT clubName, roomNo FROM clubTbl;**

-- , UNION

EXCEPT **SELECT name, mobile1+mobile2 AS ' ' FROM userTbl EXCEPT SELECT name, mobile1+mobile2 FROM userTbl WHERE mobile1 IS NULL;**

INTERSECT **SELECT name, mobile1+mobile2 AS ' ' FROM userTbl INTERSECT SELECT name, mobile1+mobile2 FROM userTbl WHERE mobile1 IS NULL;**

```
CASE WHEN - THEN      DECLARE @credit CHAR(1), @point INT = 77          ELSE . END .
                        SET @credit =
                        CASE
                          WHEN (@point >= 90) THEN 'A'
                          WHEN (@point >=80) THEN 'B'
                          ELSE 'C'
                        END
                        PRINT N' : ' + @credit
```

(cont)

-- CASE	<pre> DECLARE @credit CHAR(1), @point INT = 77 SET @credit = CASE @point WHEN 90 THEN 'A' WHEN 80 THEN 'B' ELSE 'C' END PRINT N' : ' + @credit </pre>	
	<pre> SELECT SUM(CASE WHEN classify = ' ' THEN price ELSE 0) AS sum_price_close, SUM(CASE WHEN classify = ' ' THEN price ELSE 0) AS sum_price_kitchen SUM(CASE WHEN classify = ' ' THEN price ELSE 0) AS sum_price_office FROM Goods; </pre>	
IF - ELSE	<pre> IF @var1 = 100 PRINT '100' ELSE PRINT 'Not 100' </pre>	begin - end
WHILE	<pre> DECLARE @i INT = 1, @hap BIGINT = 0 WHILE (@i < 100) BEGIN IF (@i%7 = 0) BEGIN SET @i += 1 CONTINUE END SET @hap += @i IF (@hap > 1000) BREAK SET @i += 1 END PRINT N' : ' + CAST(@hap AS NCHAR(10)) </pre>	
-- CONTINUE / BREAK		
GOTO	<pre> IF (@hap > 1000) GOTO endpoint endpoint: PRINT N' = ' + CAST(@hap AS NCHAR(10)) </pre>	+
WAIT FOR DELAY	WAIT FOR DELAY '00:00:05'	5
WAIT FOR TIME	WAIT FOR TIME '23:59'	23:55



(cont)

```

TRY / CATCH      BEGIN TRY  SQL  END TRY
                 BEGIN CATCH  SQL  END CATCH

-- ERROR_LINE    BEGIN TRY INSERT INTO userTbl VLAUES('LSG', ' ') END TRY
                 BEGIN CATCH PRINT ERROR_LINE() END CATCH

-- ERROR_MESSAGE() / ERROR_NUM() / ERROR_PROCEDURE()

RAISERROR        RAISERROR(N'  !!', 5, 1);                5, 1 Error message

THROW            THORW 50000, N'  !!', 1                Error 50000, 1 Error .50000

EXEC()           DECLARE @curDATE DATE, @curMonth VCHAR(2), @curDay VCHAR(2), @sql VCHAR(100)
                 SET @curDATE = GERDTE()
                 SET @curMonth = MONTH(@curDATE)
                 SET @curDay = DAY(@curDATE)
                 SET @sql = 'CREATE TABLE myTbl' + @curMonth + '_' + @curDay
                 SET @sql += '(id INT, name NCHAR(10))'
                 EXEC(@sql)
    
```

Procedure

Syntex

Python

SQL

BEGIN - END

```

CREATE PROCEDURE    CREATE PROCEDURE usp_user1 @userBirth INT, @userHeight INT = 178 AS
                   SELECT * FROM userTbl WHERE birthYear > @userBirth AND height > @userHeight;
                   EXEC usp_user2 1970, 178; EXEC usp_user2 @userHeight=178, @userBirth=1970;
                   EXEC usp_user1 1960, 170;
                   EXEC usp_user1 1680;

                   CREATE #usp_temp AS SELECT * FROM userTbl; --
    
```

C

By [nollae93](#)
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 30 of 33.

Sponsored by [CrosswordCheats.com](#)
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Procedure (cont)

```
Output CREATE PROCEDURE usp_user2 @textValue NCHAR(10), @outValue INT OUTPUT AS
        INSERT INTO testTbl VALUES (@textValue);
        SELECT @outValue = IDENT_CURRENT('testTbl');
CREATE TABLE testTbl (id INT IDENTITY, txt NCHAR(10));
DECLARE @myValue INT;
EXEC usp_user2 ' 1', @myValue OUTPUT;
PRINT ' id ==> ' + CAST(@myValue AS CHAR(5));
```

```
RETURN CREATE PROCEDURE usp_return @userName NVARCHAR(10) AS
        DECLARE @userID CHAR(8);
        SELECT @userID = userID FROM userTbl WHERE name = @userName;
        IF @userID <> RETURN 0;
        ELSE RETURN -1;
DECLARE @retVal INT
EXEC @retVal = usp_return ' ';
SELECT @retVal;
```

```
Table Type CREATE TYPE userTblType AS TABLE (userID CHAR(8), name NVARCHAR(10), birthYear INT, addr NCHAR(2));
CREATE PROCEDURE usp_tableTypeParam @tblParam userTblType READONLY AS -- READONLU
        SELECT * FROM @tblParam WHERE birthYear < 1970;
DECLARE @tblVar userTblType;
INSERT INTO @tblVar SELECT userID, name, birthYear, addr FROM userTbl;
EXEC usp_tableTypeParam @tblVar;

EXEC sp_helptext usp_user1;
```

```
SELECT o.name, m.definition FROM sys.sql_modules m JOIN sys.objects o ON m.object_id = o.object_id AND o.TYPE = 'P';
```

Function

```
Syntax TRY - CATCH / CREATE / ALTER / DROP
```

```
BEGIN - END
```

```
CREATE FUNCTION CREATE FUNCTION ufn_getAge(@bYear INT)
        RETURNS INT AS -- RETURNS 'S'
BEGIN
        DECLARE @age INT;
        SET @age = YEAR(GETDATE()) - @bYear;
        RETURN @age;
END
```



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 31 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Function (cont)

```
SELECT dbo.ufn_getAge(1979);-- : Schema .
```

```
SELECT userID, name, dbo.ufn_getAge(birthYear) AS [ ] FROM userTbl;
```

ALTER FUNCTION

```
ALTER FUNCTION ufn_getAge(@bYear INT)
    RETURNS INT AS
BEGIN
    DECLARE @age INT;
    SET @age = YEAR(GETDATE()) - @bYear + 1;
    RETURN @age;
END
```

DROP FUNCTION

```
DROP FUNCTION ufn_getAge;
```

WITH SCHEMABINDING

```
-- .
```

Inline Table

```
CREATE FUNCTION ufn_getUser(@ht INT)
    RETURNS TABLE
    WITH SCHEMABINDING -- Table .
AS
    RETURN (SELECT userID, name, height FROM dbo.userTbl WHERE height > @ht)
-- SCHEMABINDIND Table Schema .
SELECT * FROM dbo.ufn_getUser(177);
```



By **nollae93**
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 32 of 33.

Sponsored by **CrosswordCheats.com**
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Function (cont)

```

CREATE FUNCTION ufn_userGrade(@bYear INT)
  RETURNS @retTable TABLE (userID CHAR(8), name NCHAR(10), grade NCHAR(5)) AS
BEGIN
  DECLARE @rowCnt INT;
  SELECT @rowCnt = COUNT(*) FROM userTbl WHERE birthYear >= @bYear;

  --      ' '      .
  IF @rowCnt <= 0
  BEGIN
    INSERT INTO @retTable VALUES (' ', ' ', ' ');
    RETURN;
  END;

  -- 1
  INSERT INTO @retTable
    SELECT U.userID, U.name,
      CASE
        WHEN (SUM(price*amount) >= 1500) THEN ' '
        WHEN (SUM(price*amount) >= 1000) THEN ' '
        WHEN (SUM(price*amount) > 1) THEN ' '
        ELSE ' '
      END
    FROM buyTbl B RIGHT OUTER JOIN userTbl U ON B.userID = U.userID
    WHERE birthYear >= @bYear
    GROUP BY U.userID, U.name;
  RETURN;
END;
SELECT * FROM dbo.ufn_userGrade(1970);

```

Cursor

(DECLARE) - (OPEN) - (FETCH) - - (CLOSE) - (DEALLOCATE)



By [nollae93](#)
cheatography.com/nollae93/

Not published yet.
 Last updated 7th September, 2020.
 Page 33 of 33.

Sponsored by [CrosswordCheats.com](#)
 Learn to solve cryptic crosswords!
<http://crosswordcheats.com>