SQL Tutorial

www.tsql.info

Select	Join	Create database	Avg
Distinct	Inner Join	Create table	Count
Where	Left Join	Create index	Max
And - OR	Right Join	Create view	Min
Order By	Full Join	Increment	Sum
Group By	Union	Drop	Mid
Having	TOP	Alter table	Len
Like	Wildcard	Add column	Round
Insert	IN	Alter column	Now
Update	Between	Rename column	UCase
Delete	ISNULL	Drop column	LCase

Select

Select Syntax:

SELECT column(s)FROM table ;
SELECT * FROM table ;

Store table:

OBJECT_ID	PRICE
1	100
2	300
3	800
4	300

Example 1:

SELECT * FROM store;

OBJECT_ID	PRICE
1	100
2	300
3	800
4	300

Example 2:

SELECT price FROM store;

PRICE
100
300
800
300

Distinct

Distinct Syntax:

SELECT DISTINCT column_name FROM table_name ;

Store table:

OBJECT_ID	PRICE
1	100
2	400
3	800
4	400

Example:

SELECT DISTINCT price FROM store;

PRICE
100
400
800

Where

Where Syntax:

SELECT column_name(s) FROM table_name WHERE condition ;

Store table:

OBJECT_ID	PRICE
1	200
2	500
3	900
4	500

Example 1:

SELECT * FROM store WHERE price = 500;

OBJECT_ID	PRICE
2	500

4	500

Example 2:

SELECT * FROM store WHERE price > 500;

OBJECT_ID	PRICE
3	900

And & OR

Store table:

OBJECT_ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

Example 1:

SELECT * FROM store WHERE name='B' **AND** price = 500;

OBJECT_ID	PRICE	NAME
2	500	B

Example 2:

SELECT * FROM store WHERE name='B' **OR** price = 500;

OBJECT_ID	PRICE	NAME
2	500	B
4	500	D

Example 3:

SELECT * FROM store WHERE price = 900 **AND** (name='A' **OR** name='C');

OBJECT_ID	PRICE	NAME

3	900	C

Order By

Order By Syntax:

SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC

Store table:

OBJECT_ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

Example 1:

SELECT * FROM store ORDER BY price, name;

OBJECT_ID	PRICE	NAME
1	200	A
2	500	B
4	500	D
3	900	C

Example 2:

SELECT * FROM store ORDER BY name DESC;

OBJECT_ID	PRICE	NAME
4	500	D
3	900	C
2	500	B
1	200	A

Group By

Group By Syntax:

SELECT column_name1, aggregate_function(column_name2)

FROM table GROUP BY column_name1

Store table:

OBJECT_ID	PRICE	TYPE
1	200	LOW
2	500	MEDIUM
3	900	HIGH
4	500	MEDIUM

Example:

SELECT type, SUM(price) FROM store GROUP BY type;

TYPE	PRICE
LOW	200
MEDIUM	1000
HIGH	900

Having

Having Syntax:

SELECT column_name(s), aggregate_function(column_name)
FROM my_table
WHERE condition {optional}
GROUP BY column_name(s)
HAVING (aggregate_function condition)

Having Example:

Sales table:

ID	PRICE	CUSTOMER
1	200	David
2	500	Linda
3	900	Tom
4	500	David
5	1200	Ellen
6	1200	Linda

```
SELECT customer, SUM(price)
FROM sales
GROUP BY customer
HAVING SUM(price) > 1000
```

Having Result:

customer	SUM(price)
Linda	1700
Ellen	1200

Like

Like Syntax:

SELECT column(s) FROM table WHERE column LIKE pattern

Employee table:

EMPLOYEE_ID	NAME	DEP_ID
1	John	21
2	Samantha	22

3	Tom	23
4	James	24
5	Sandra	24

Like Example 1:

Find the employee names that contain letters: am.
SELECT * FROM employee WHERE name LIKE '%am%';

Like Result:

EMPLOYEE_ID	NAME	DEP_ID
2	Samantha	22
4	James	24

Like Example 2:

Find the employee names that begin with: J.
SELECT * FROM employee WHERE name LIKE 'J%';

Like Result:

EMPLOYEE_ID	NAME	DEP_ID
1	John	21
4	James	24

Like Example 3:

Find the employee names that end with: a.
SELECT * FROM employee WHERE name LIKE '%a';

Like Result:

EMPLOYEE_ID	NAME	DEP_ID
2	Samantha	22
5	Sandra	24

Insert into

Insert into Syntax:

INSERT INTO table_name VALUES (value1, value2, ...)

OR

INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...)

Store table:

	OBJECT_ID	PRICE	NAME
1		200	A
2		500	B
3		900	C
4		500	D

Example 1:

INSERT INTO store VALUES (5, 600, 'E');

Example 2:

INSERT INTO store(object_id, price, name) VALUES (6, 400, 'F');

	OBJECT_ID	PRICE	NAME
1		200	A
2		500	B
3		900	C
4		500	D
5		600	E
6		400	F

Update

Update Syntax:

UPDATE table_name SET column1 = new_value1, column2 = new_value2,... WHERE {condition}

IF you don't put the {condition} then all records on the updated column will be chenged.

Store table:

OBJECT_ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

Example 1:

```
UPDATE store SET price = 300 WHERE object_id=1 AND name='A';

SELECT * FROM store WHERE object_id=1 AND name='A';
```

OBJECT_ID	PRICE	NAME
1	300	A

Example 2:

```
UPDATE store SET price = 1000, name = 'Y' WHERE object_id=3;

SELECT * FROM store WHERE object_id=3;
```

OBJECT_ID	PRICE	NAME
3	1000	Y

Delete

Delete Syntax:

```
DELETE FROM table_name WHERE {condition}
```

IF you don't put the {condition} then all the records from the table will be erased.

Store table:

OBJECT_ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

Example:

DELETE FROM store WHERE price=500;

SELECT * FROM store;

OBJECT_ID	PRICE	NAME
1	200	A
3	900	C

Join

Join Example:

```
SELECT s.student_id, s.name, b.book_id, b.price
FROM students s, books b
WHERE s.student_id = b.student_id
AND b.price > 90 ;
```

Students table:

STUDENT_ID	NAME	YEAR
1	STUDENT_1	I
2	STUDENT_2	II
3	STUDENT_3	III
4	STUDENT_4	IV

Books table:

BOOK_ID	STUDENT_ID	PRICE
1	1	40
2	2	50
3	3	70
4	1	100
5	2	120
6	4	90
7	3	200
8	2	150

Join Result:

STUDENT_ID	NAME	BOOK_ID	PRICE
1	STUDENT_1	4	100
2	STUDENT_2	5	120
3	STUDENT_3	7	200
2	STUDENT_2	8	150

Inner Join

Inner Join Example:

```
SELECT s.student_id, s.name, SUM(b.price)
FROM students s INNER JOIN books b
ON s.student_id = b.student_id
GROUP BY b.price ;
```

Students table:

STUDENT_ID	NAME	YEAR
1	STUDENT_1	I
2	STUDENT_2	II
3	STUDENT_3	III
4	STUDENT_4	IV

Books table:

BOOK_ID	STUDENT_ID	PRICE
1	1	40
2	2	50
3	3	70
4	1	100
5	2	120
6	4	90
7	3	200
8	2	150

Inner Join Result:

STUDENT_ID	NAME	PRICE
1	STUDENT_1	140
2	STUDENT_2	320
3	STUDENT_3	270
4	STUDENT_4	90

Left Join

Left Join Example:

```
SELECT s.student_id, s.name, b.price
FROM students s LEFT JOIN books b
ON s.student_id = b.student_id
ORDER BY s.student_id ;
```

Students table:

STUDENT_ID	NAME	YEAR
1	STUDENT_1	I
2	STUDENT_2	II
3	STUDENT_3	III
4	STUDENT_4	IV
5	STUDENT_5	I
6	STUDENT_6	IV

Books table:

BOOK_ID	STUDENT_ID	PRICE
1	1	40
2	2	50
3	3	70
4	1	100
5	2	120
6	4	90
7	3	200
8	2	150

Left Join Result:

STUDENT_ID	NAME	PRICE
1	STUDENT_1	40
1	STUDENT_1	100
2	STUDENT_2	50
2	STUDENT_2	120
2	STUDENT_2	150
3	STUDENT_3	70
3	STUDENT_3	200
4	STUDENT_4	90
5	STUDENT_5	
6	STUDENT_6	

Right Join

Right Join Example:

```
SELECT * FROM employee e RIGHT JOIN department d
ON e.DEP_ID = d.DEP_ID
ORDER BY d.DEP_ID ;
```

Employee table:

EMPLOYEE_ID	NAME	DEP_ID
1	EMPLOYEE_1	21
2	EMPLOYEE_2	22
3	EMPLOYEE_3	23
4	EMPLOYEE_4	24

5	EMPLOYEE_5

Department table:

DEP_ID	DEP_NAME
21	DEP_21
22	DEP_22
23	DEP_23
24	DEP_24
25	DEP_25

Right Join Result:

EMPLOYEE_ID	NAME	DEP_ID	DEP_ID	DEP_NAME
1	EMPLOYEE_1	21	21	DEP_21
2	EMPLOYEE_2	22	22	DEP_22
3	EMPLOYEE_3	23	23	DEP_23
4	EMPLOYEE_4	24	24	DEP_24
			25	DEP_25

Full Join

Full Join Example:

```
SELECT * FROM employee e FULL JOIN department d
ON e.DEP_ID = d.DEP_ID
ORDER BY e.employee_id ;
```

Employee table:

EMPLOYEE_ID	NAME	DEP_ID
1	EMPLOYEE_1	21
2	EMPLOYEE_2	22
3	EMPLOYEE_3	23
4	EMPLOYEE_4	24
5	EMPLOYEE_5	

Department table:

DEP_ID	DEP_NAME
21	DEP_21
22	DEP_22
23	DEP_23
24	DEP_24
25	DEP_25

Full Join Result:

EMPLOYEE_ID	NAME	DEP_ID	DEP_ID	DEP_NAME
1	EMPLOYEE_1	21	21	DEP_21
2	EMPLOYEE_2	22	22	DEP_22
3	EMPLOYEE_3	23	23	DEP_23
4	EMPLOYEE_4	24	24	DEP_24
5	EMPLOYEE_5			
			25	DEP_25

Union

Union Syntax:

```
SELECT column_name(s) FROM table_name_a
UNION
SELECT column_name(s) FROM table_name_b
```

Union All Syntax:

```
SELECT column_name(s) FROM table_name_a
UNION ALL
SELECT column_name(s) FROM table_name_b
```

Employee_a		Employee_b	
id	**name**	**id**	**name**
1	Martin	1	David
2	Carol	2	Barbara
3	Davis	3	Carol
4	Sandra	4	Sandra

UNION Example:

SELECT * FROM employee_a UNION SELECT * FROM employee_b;

UNION Result:

1 Martin
2 Carol
3 Davis
4 Sandra
1 David
2 Barbara
3 Carol

UNION ALL Example:

SELECT * FROM employee_a UNION ALL SELECT * FROM employee_b;

UNION ALL Result:

1 Martin
2 Carol
3 Davis
4 Sandra
1 David
2 Barbara
3 Carol
4 Sandra

TOP

TOP Syntax:

SELECT **TOP number** column_name(s) FROM table_name
SELECT **TOP percent** column_name(s) FROM table_name

Employee table:

EMPLOYEE_ID	NAME	DEP_ID
1	EMPLOYEE_1	21

2	EMPLOYEE_2	22
3	EMPLOYEE_3	23
4	EMPLOYEE_4	24

TOP number Example:

SELECT TOP 3 * FROM employee;

TOP Result:

EMPLOYEE_ID	NAME	DEP_ID
1	EMPLOYEE_1	21
2	EMPLOYEE_2	22
3	EMPLOYEE_3	23

TOP percent Example:

SELECT TOP 50 PERCENT * FROM employee;

TOP Result:

EMPLOYEE_ID	NAME	DEP_ID
1	EMPLOYEE_1	21
2	EMPLOYEE_2	22

Wildcard

Wildcard	Definition
%	Represents zero or more characters
_	Represents exactly one character
[char list]	Represents any single character in charlist
[^char list] or [!char list]	Represents any single character not in charlist

Students table:

ID	NAME	STATE

1	Tom	Arizona
2	Martin	Texas
3	Helen	Florida
4	Tania	California
5	Harry	Colorado

_ Wildcard Example:

Select the student with a name that starts with any character, followed by "ar".
SELECT * FROM students WHERE name LIKE '_ar';

_ Wildcard Result:

ID	NAME	STATE
2	Martin	Texas
5	Harry	Colorado

[char list] Wildcard Example:

Select the student with a name that starts with any character from char list.
SELECT * FROM students WHERE name LIKE '[tma]%';

[char list] Wildcard Result:

1	Tom	Arizona
2	Martin	Texas
4	Tania	California

[!char list] Wildcard Example:

Select the student with a name that do not starts with any character from char list.
SELECT * FROM students WHERE name LIKE '[!tma]%';

[!char list] Wildcard Result:

3	Helen	Florida
5	Harry	Colorado

In

In Syntax:

SELECT column_name(s) FROM table_name WHERE column_name IN (value1,value2,value3,...)

Employee table:

EMPLOYEE_ID	NAME	DEP_ID
1	John	33
2	Samantha	34
3	Bill	35
4	James	36
5	Sandra	37

In Example:

SELECT * FROM employee WHERE name IN ('Samantha', 'Bill', 'Sandra');

In Result:

2	Samantha	34
3	Bill	35
5	Sandra	37

Between

Between Syntax:

SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value_a AND value_b

Employee table:

EMPLOYEE_ID	NAME	DEP_ID
1	John	33
2	Samantha	34
3	Bill	35
4	James	36
5	Sandra	37

Between Example:

```
SELECT * FROM employee WHERE dep_id BETWEEN 34 AND 36;
```

Between Result:

2	Samantha	34
3	Bill	35
4	James	36

ISNULL

ISNULL Syntax:

```
SELECT ISNULL(column_name,0) FROM table_name
```

Sales table:

ID	PRICE	NAME
1	100	A
2		B
3	600	C
4		D

ISNULL Example:

```
SELECT id, ISNULL(price,0), name FROM store;
```

ISNULL Result:

ID	PRICE	NAME
1	100	A
2	0	B
3	600	C
4	0	D

Create Table

Create Database Syntax:

CREATE DATABASE database_name

Create Database Example:

CREATE DATABASE new_dba;

Create Table

Create Table Syntax:

```
CREATE TABLE new_table
(
column_name_1 datatype,
column_name_2 datatype,
....
column_name_n datatype
);
```

Create Table Example:

```
CREATE TABLE sales
(
id int,
price int,
name varchar(50)
);
```

Create Index

Create Index Syntax:

```
CREATE INDEX my_index
ON my_table (column_name)
```

Create Unique Index Syntax:

```
CREATE UNIQUE INDEX my_index
ON my_table (column_name)
```

Create View

Create View Syntax:

```
CREATE VIEW my_view_name AS
SELECT column_name(s)
FROM my_table_name
WHERE condition
```

Create View Example:

Sales table:

ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

```
CREATE VIEW sales_view AS
SELECT id, price, name
FROM sales
WHERE price=500;
```

Create View Result:

ID	PRICE	NAME
2	500	B
4	500	D

Increment

Identity Syntax:

```
CREATE TABLE new_table
(
column_name_1 PRIMARY KEY IDENTITY,
column_name_2 datatype,
....
column_name_n datatype
);
```

Identity Example:

```
CREATE TABLE sales
(
id int PRIMARY KEY IDENTITY,
price int,
name varchar(50)
);
```

Drop

Drop Table Syntax:

```
DROP TABLE table_name;
```

Drop Database Syntax:

```
DROP DATABASE database_name;
```

Drop Index Syntax:

DROP INDEX table_name.index_name;

Truncate Table Syntax:

TRUNCATE TABLE table_name;

Alter Table

Alter Table
Add Column
Alter Column
Rename Column
Drop Column

Add Column

Add Column Syntax:

ALTER TABLE table_name
ADD column_name data_type

Employee table:

Column name	Data_type
id	int
name	varchar(250)

Add Column Example:

ALTER TABLE employee ADD (dep_id int, address varchar(100));

Add Column Result:

Column name	Data_type
id	int
name	varchar(250)
dep_id	int
address	varchar(100)

Alter Column

Alter Column Syntax:

ALTER TABLE table_name
ALTER COLUMN column_name data_type

Employee table:

Column name	Data_type
id	int
name	varchar(250)
dep_id	int
address	varchar(100)

Alter Column Example:

ALTER TABLE employee ALTER COLUMN address varchar(400);

Alter Column Result:

Column name	Data_type
id	int
name	varchar(250)
dep_id	int
address	varchar(400)

Rename Column

Rename Column Syntax:

EXEC sp_rename 'Table.Old_Column', 'New_Column', 'COLUMN'

Employee table:

Column name	Data_type
id	int
name	varchar(250)
dep_id	int
address	varchar(400)

Rename Column Example:

EXEC sp_rename 'employee.address', 'new_address', 'COLUMN' ;

Rename Column Result:

Column name	Data_type
id	int
name	varchar(250)
dep_id	int
new_address	varchar(400)

Drop Column

Drop Column Syntax:

ALTER TABLE table_name
DROP COLUMN column_name

Employee table:

Column name	Data_type
id	int
name	varchar(250)
dep_id	int
address	varchar(400)

Drop Column Example:

ALTER TABLE employee DROP COLUMN address;

Drop Column Result:

Column name	Data_type
id	int
name	varchar(250)
dep_id	int

AVG

AVG Syntax:

SELECT AVG(column_name) FROM table_name

Sales table:

ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

AVG Example:

SELECT AVG(price) FROM store;

AVG Result:

525

Count

Count Syntax:

SELECT COUNT(column_name) FROM table_name
SELECT COUNT(*) FROM table_name

Sales table:

ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

Count Example 1:

SELECT COUNT(id) FROM store WHERE price=500;

Count Result: 2

Count Example 2:

SELECT COUNT(*) FROM store;

Count Result: 4

Max

Max Syntax:

SELECT MAX(column_name) FROM table_name

Sales table:

ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

Max Example:

SELECT MAX(price) FROM store;

Max Result: 900

Min

Min Syntax:

SELECT MIN(column_name) FROM table_name

Sales table:

ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

Min Example:

SELECT MIN(price) FROM store;

Min Result: 200

Sum

Sum Syntax:

SELECT SUM(column_name) FROM table_name

Sales table:

ID	PRICE	NAME
1	200	A
2	500	B
3	900	C
4	500	D

Sum Example:

SELECT SUM(price) FROM store;

Sum Result: 2100

Mid

Mid Syntax:

SELECT MID(column_name,start[,length]) FROM table_name

Students table:

ID	NAME	State
1	Tom	Arizona
2	Linda	Texas
3	Helen	Florida
4	Robert	California

Mid Example:

SELECT state, MID(state,1,4) FROM students;

Mid Result:

State	MID(state,1,3)
Arizona	Ari
Texas	Tex
Florida	Flo
California	Cal

Len

Len Syntax:

SELECT LEN(column_name) FROM table_name

Students table:

ID	NAME	State
1	Tom	Arizona
2	Linda	Texas
3	Helen	Florida
4	Robert	California

Len Example:

SELECT state, LEN(state) FROM students;

Len Result:

State	LEN(state)
Arizona	7
Texas	5
Florida	7
California	10

Round

Round Syntax:

SELECT ROUND(column_name,decimal precision) FROM table_name

Sales table:

ID	PRICE	NAME
1	25.845	A
2	26.97	B

3	27.9	C
4	28.34	D

Round Example 1:

SELECT id, ROUND(price,1) FROM store;

Round Result:

ID	PRICE
1	25.8
2	26.9
3	27.9
4	28.3

Round Example 2:

SELECT id, ROUND(price,0) FROM store;

Round Result:

ID	PRICE
1	26
2	27
3	28
4	28

Now

Now Syntax:

SELECT NOW() FROM my_table

Sales table:

ID	PRICE	NAME
1	25.845	A
2	26.97	B

3	27.9	C
4	28.34	D

Now Example:

SELECT id, price, NOW() as PriceDate FROM store;

Now Result:

Id	Price	PriceDate
1	25.845	12/9/2012 15:30:23 PM
2	26.97	12/9/2012 15:30:23 PM
3	27.9	12/9/2012 15:30:23 PM
4	28.34	12/9/2012 15:30:23 PM

UCase

UCase Syntax:

SELECT UCASE(column_name) FROM table_name

Students table:

ID	NAME	State
1	Tom	Arizona
2	Linda	Texas
3	Helen	Florida
4	Robert	California

UCase Example:

SELECT name, UCASE(name) FROM students;

UCase Result:

Name	UCASE(name)
Tom	TOM
Linda	LINDA

Helen	HELEN
Robert	ROBERT

LCase

LCase Syntax:

SELECT LCASE(column_name) FROM table_name

Students table:

ID	NAME	State
1	Tom	Arizona
2	Linda	Texas
3	Helen	Florida
4	Robert	California

LCase Example:

SELECT name, LCASE(name) FROM students;

LCase Result:

Name	LCASE(name)
Tom	tom
Linda	linda
Helen	helen
Robert	robert

Resources:

www.tsql.info/sql.php