

# T SQL

## DATA TYPES AND SELECT

# Типи даних (числові)

<b>BIGINT</b>	ціле зі знаком в діапазоні від - 9223372036854775808 до 9223372036854775807.
<b>INT</b>	ціле зі знаком в діапазоні від - 2147483648 до 2147483647.
<b>SMALLINT</b>	ціле зі знаком в діапазоні від - 32768 до 32767.
<b>TINYINT</b>	ціле без знаку в діапазоні від 0 до 255.
<b>BIT</b>	ціле, замінює логічний тип даних , може набувати значень: або 0, або 1.
<b>DECIMAL ЧИ NUMERIC</b>	Дійсне число з фіксованою комою. Синтаксис визначення decimal(p, s), де p - точність (кількість десяткових розрядів числа) від 1 до 38, s - масштаб (кількість десяткових розрядів після коми) від 0 до p.
<b>FLOAT</b>	Дійсне число з плаваючою комою. Синтаксис визначення : float[(p)], де p - точність (число знаків основи) від 1 до 53. За умовчанням p = 53.

# Типи даних (символьні або текст)

<b>CHAR</b>	Синтаксис визначення : <code>char[(n)]</code> , де <code>n</code> - число символів від 1 до 8000. Якщо <code>n</code> не задається у визначенні даних або в інструкції оголошення змінної, довжина за умовчанням дорівнює 1. Якщо при використанні функцій <code>CAST</code> і <code>CONVERT</code> <code>n</code> не задається, довжина за умовчанням дорівнює 30.
<b>VARCHAR</b>	Аналогічний <code>char</code> , але при зберіганні рядків пам'яті відводиться по довжині рядка, а не завжди <code>n</code> байт, але не більше <code>n</code> . Синтаксис визначення : <code>varchar[(n)]</code> , де <code>n</code> - число символів від 1 до 8000. Якщо <code>n</code> не задається у визначенні даних або в інструкції оголошення змінної, довжина за умовчанням дорівнює 1. Якщо при використанні функцій <code>CAST</code> і <code>CONVERT</code> <code>n</code> не задається, довжина за умовчанням дорівнює 30.
<b>TEXT</b>	Масив 8-розрядних символів розміром до 2 Гб.
<b>NCHAR</b>	Символи зберігаються в кодуванні <code>UNICODE UCS - 2</code> . Синтаксис визначення : <code>nchar[(n)]</code> , де <code>n</code> - число символів від 1 до 4000. Якщо <code>n</code> не задається у визначенні даних або в інструкції оголошення змінної, довжина за умовчанням дорівнює 1. Якщо при використанні функцій <code>CAST</code> і <code>CONVERT</code> <code>n</code> не задається, довжина за умовчанням дорівнює 30.
<b>NVARCHAR</b>	Аналогічний <code>nchar</code> за винятком того, що при зберіганні рядків пам'яті відводиться по довжині рядка, а не завжди <code>n</code> байт, але не більше <code>n</code> . Синтаксис визначення : <code>nvarchar[(n)]</code> , де <code>n</code> - число символів від 1 до 4000. Якщо <code>n</code> не задається у визначенні даних або в інструкції оголошення змінної, довжина за умовчанням дорівнює 1. Якщо при використанні функцій <code>CAST</code> і <code>CONVERT</code> <code>n</code> не задається, довжина за умовчанням дорівнює 30.
<b>NTEXT</b>	Масив 16-иразрядних символів розміром до 2 Гб. Символи зберігаються в кодуванні <code>UNICODE UCS - 2</code> .

# Типи даних (дата та час)

<b>DATE</b>	Зберігає дати від 0001-01-01 (1 січня 0001 року) до 9999-12-31 (31 грудня 9999 року).
<b>TIME</b>	Зберігає час в діапазоні від 00: 00: 00.0000000 до 23: 59: 59.9999999. Може мати форму TIME (n), де n представляє кількість цифр від 0 до 7 в дробовій частини секунд.
<b>DATETIME</b>	Зберігає дати і час від 01/01/1753 до 31/12/9999.
<b>SMALLDATETIME</b>	Зберігає дати і час в діапазоні від 01/01/1900 до 06/06/2079, тобто найближчі дати.

# SELECT

```
SELECT  
FROM table_references  
[WHERE where_condition]  
[GROUP BY]  
[HAVING where_condition]  
[ORDER BY]
```

**ПОРЯДОК ВАЖЛИВИЙ !**

# SELECT

Варто відмітити порядок опрацювання виразу SELECT:

1. FROM
2. ON
3. JOIN
4. WHERE
5. GROUP BY
6. HAVING
7. SELECT
8. DISTINCT
9. ORDER BY
10. TOP

# SELECT FROM

## **SELECT**

column\_name,  
column\_name2

**FROM** table\_name;

## **SELECT**

\*

**FROM** table\_name;

## **SELECT DISTINCT**

column\_name,  
column\_name2

**FROM** table\_name;

# SELECT FROM WHERE

## SELECT

column\_name,column\_name

**FROM** table\_name

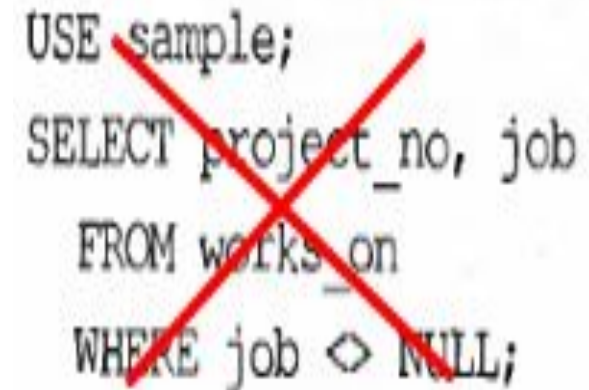
**WHERE** column\_name *operator* value;

- **SELECT** \*
- **FROM** tblRoutes
- **WHERE** Route\_id = 1000100016 **AND** City\_id = 204015
  
- **SELECT** \*
- **FROM** tblRoutes
- **WHERE** Route\_id = 1000100016 **OR** City\_id = 111024



# WHERE

Оператор	Призначення
=	Одинакові
<> Або !=	Не однакові
>	Більше чим
<	Менше чим
>=	Більше чим або рівно з умовою
<=	Менше чим або рівно з умовою
BETWEEN	Між вказаними значеннями
LIKE	шукати за зразком
IN	стовпець специфічних можливих значень



```
USE sample;
SELECT project_no, job
FROM works_on
WHERE job <> NULL;
```

**IS [NOT] NULL**

А також функція: **ISNULL**(column,'Name instead of NULL')

# ORDER BY

**SELECT**

column\_name,  
column\_name2

**FROM** table\_name

**ORDER BY** column\_name,column\_name2

**ASC | DESC;**

# Функції SQL

Елемент SQL	Призначення
<u>AVG()</u>	Функція розрахунку середнього значення <code>SELECT AVG(column_name) FROM table_name</code>
<u>COUNT()</u>	Функція розрахунку кількості записів рядків <code>SELECT COUNT(column_name) FROM table_name;</code>
<u>MAX()</u>	Функція розрахунку найбільшого значення вибраної стовпця <code>SELECT MAX(column_name) FROM table_name;</code>
<u>MIN()</u>	Функція розрахунку найменшого значення вибраної стовпця <code>SELECT MIN(column_name) FROM table_name;</code>
<u>SUM()</u>	Функція розрахунку суми елементів стовпця Баних Даних <code>SELECT SUM(column_name) FROM table_name;</code>

# SQL – GROUP BY

**SELECT**

food,

**SUM(sold) AS totalSold**

**FROM** FoodCart

**GROUP BY** food;

date	food	sold
02/25/08	pizza	349
02/26/08	hotdog	500
02/26/08	pizza	70

food	totalSold
hotdog	500
pizza	419

# SQL – HAVING

```
SELECT  
    food,  
    SUM(sold) AS totalSold  
FROM FoodCart  
GROUP BY food  
HAVING SUM(sold) > 450;
```

date	food	sold
02/25/08	pizza	349
02/26/08	hotdog	500
02/26/08	pizza	70

food	totalSold
hotdog	500

# Оператори IN та BETWEEN

**WHERE** condition **IN** (x1, x2,...)

or

**WHERE** condition **NOT IN** (x1, x2)

**WHERE** condition **BETWEEN** x1 AND x2

or

**WHERE** condition **NOT BETWEEN** x1 AND x2

# LIKE

column [NOT] **LIKE** 'pattern'

де pattern – рядок або дата/число.

**%** – задає будь-яку послідовність символів, від нуля і більше;

**\_** – заміняє один символ;

**[abc]** або **[a-c]** – заміняє собою перелік або діапазон символів;

**^** – заперечення переліку або діапазону.

Всі з шаблонних символів, охоплених **[ ]** є звичайними символами.

# Q&A