

# Мова SQL (частина 2)

# Оператор **LIKE**

- Оператор **LIKE** застосовний тільки до полів типу **CHAR** або **VARCHAR**, з якими він використовується, щоб знаходити підрядки.
- Є два типи групових символів використуваних з **LIKE**:
  - *символ підкреслення* ( **\_** ) заміщає будь-який одиночний символ. Наприклад, 'b\_t' буде відповідати словам 'bat' або 'bit', але не буде відповідати 'brat'.
  - *знак відсотка* ( **%** ) заміщає послідовність будь-якого числа символів (включаючи символи нуля). Наприклад '%p%t' буде відповідати словам 'put', 'posit', або 'opt', але не 'spite'.

Приклад № 11 Виведемо інформацію про службовців, що проживають у містах, що починаються на букву М.

```
SELECT *  
FROM S  
WHERE City LIKE "M%";
```

SNum	SName	City	Age	Comm
104	Рома	Москва	20	0,15
112	Андрей	Минск		0,11

Потрібно вивести інформацію про службовців,  
в іменах яких зустрічається буква «р».

```
SELECT *  
FROM S  
WHERE Sname LIKE 'P_%' OR Sname LIKE '_%p%';
```

Якщо необхідно визначити записи, що містить символ підкреслення або відсотка, то в LIKE предикаті, можна визначити будь-який одиночний символ як символ ESC

Наприклад, ми могли б знайти Sname стовпець, де є присутнім підкреслення, у такий спосіб:

```
SELECT *  
FROM S  
WHERE Sname LIKE '%/_%'ESCAPE'/';
```

Часто, в таблиці можуть бути зписи, що в деяких ячейках будуть «порожні», які не мають ніяких значень для деяких полів, наприклад тому що інформація не завершена, або тому що це поле просто не заповнювалося.

SQL враховує такий варіант, дозволяючи вводити значення **NULL** (ПОРОЖНІЙ) у поле, замість значення. Коли значення поля рівно NULL, це означає, що програма бази даних спеціально промаркувала це поле

- Оскільки NULL вказує на відсутність значення, тому не можна знати який буде результат будь-якого порівняння з використанням NULL.
- Коли NULL порівнюється з будь-яким значенням, навіть із іншим таким же NULL, результат буде ні вірним ні невірним, він - невідомий.
- Отже, вирази типу 'city=NULL' або 'city IN (NULL)' будуть невідомі, незалежно від значення city.

- Приклад № 12 Вивести інформацію про службовців, для яких не відомий вік

```
SELECT *  
FROM S  
WHERE Age IS NULL;
```

	SNum	SName	City	Age	Comm
▶	111	Сергей	Киев		0,1
	112	Андрей	Минск		0,11



# **ФОРМУВАННЯ ВИВОДІВ ЗАПИТІВ**

- Наприклад, можна побажати, представити комісійні службовця у відсотковому відношенні, а не як десяткові числа:

```
SELECT Sname, City, Comm*100  
FROM S;
```

SName	City	Expr1002
Петя	Харьков	12
Саша	Киев	13
Рома	Москва	15
Андрей	Киев	10
Миша	Харьков	11
Миша	Харьков	14
Саша	Бабаи	14
Сергей	Киев	10
Андрей	Минск	11

- Наприклад, вивести інформацію про службовців з поясненнями.

```
SELECT "Службовець ", Sname, " проживає в місті ", City  
FROM S;
```

Expr1000	Sname	Expr1002	City
Служащий	Петя	проживает в городе	Харьков
Служащий	Саша	проживает в городе	Киев
Служащий	Миша	проживает в городе	Харьков
Служащий	Рома	проживает в городе	Москва

# ORDER BY

**ORDER BY** - впорядковує вивід запиту згідно зі значеннями в тій або іншій кількості обраних стовпців.

Декілька стовпців впорядковуються один усередині іншого, і можна визначати зростання (**ASC**) або убубання (**DESC**) для кожного стовпця. За замовчуванням встановлене - зростання.

Виведемо інформацію про службовців,  
упорядкувавши її по зростанню комісійних.

```
SELECT *  
FROM S  
ORDER BY Comm ASC;
```

SNum	SName	City	Age	Comm
111	Сергей	Киев		0,1
107	Андрей	Киев	25	0,1
112	Андрей	Минск		0,11
103	Миша	Харьков	25	0,11
101	Петя	Харьков	28	0,12
102	Саша	Киев	24	0,13
109	Саша	Бабаи	20	0,14
108	Миша	Харьков	22	0,14
104	Рома	Москва	20	0,15

Упорядковувати таблицю наприклад за допомогою поля Comm, усередині впорядкування поля Age

```
SELECT *  
FROM S  
ORDER BY Age ASC, Comm ASC;
```

	SNum	SName	City	Age	Comm
▶	111	Сергей	Киев		0,1
	112	Андрей	Минск		0,11
	109	Саша	Бабаи	20	0,14
	104	Рома	Москва	20	0,15
	108	Миша	Харьков	22	0,14
	102	Саша	Киев	24	0,13
	107	Андрей	Киев	25	0,1
	103	Миша	Харьков	25	0,11
	101	Петя	Харьков	28	0,12

Вивести інформацію про службовців з міста Київ у порядку убубання віку:

```
SELECT *  
FROM S  
WHERE City="Київ"  
ORDER BY Age DESC;
```

```
SELECT *  
FROM S  
WHERE City="Київ"  
ORDER BY 4 DESC;
```

	SNum	SName	City	Age	Comm
	107	Андрей	Київ	25	0,1
	102	Саша	Київ	24	0,13
	111	Сергей	Київ		0,1

# Запити на декількох таблицях

- Використовуючи об'єднання, ми безпосередньо зв'язуємо інформацію з будь-яким номером таблиці, і в такий спосіб здатні створювати зв'язки між порівняльними фрагментами даних.
- При об'єднанні, таблиці представлені списком у пропозиції FROM запити, відділяються комами. Предикат запити може посилатися до будь-якого стовпця будь-якої зв'язаної таблиці й, отже, може використовуватися для зв'язку між ними. Звичайно, предикат порівнює значення в стовпцях різних таблиць, щоб визначити, чи задовольняє WHERE установленій умові.



- Повне ім'я стовпця таблиці фактично складається з імені таблиці, супроводжуваного крапкою й потім іменем стовпця. Є кілька прикладів імен:

S.Snum

S.City

Припустимо що потрібно поставити у відповідність службовцеві клієнтів у місті, у якому вони живуть, тобто побачити всі комбінації службовців і клієнтів для цього міста.

```
SELECT S.City, S.SName, C.CName  
FROM S, C  
WHERE S.City=C.City;
```

	City	SName	CName
▶	Харьков	Миша	Наташа
	Харьков	Миша	Наташа
	Харьков	Петя	Наташа
	Харьков	Миша	Рома
	Харьков	Миша	Рома
	Харьков	Петя	Рома
	Киев	Сергей	Миша
	Киев	Андрей	Миша
	Киев	Саша	Миша
	Харьков	Миша	Клим

# Таблиця О (Операції)



	ONum	ODate	CNum	Summa
▶	1	10.12.2002	201	150
	2	12.12.2002	206	250
	3	05.07.2002	211	400
	4	05.07.2002	203	350
	5	10.10.2002	207	550
	6	15.10.2002	208	100
	7	01.03.2002	208	300
	8	10.03.2002	203	200
	9	05.04.2002	212	650

Запись: 1 из 9

- Onum - номер операції (первинний ключ);
- Odate - дата здійснення операції;
- Cnum - номер клієнта, що брало участь в операції (зовнішній ключ);
- Summa - сума, на яку була укладена операція.

- Розглянемо запит на об'єднання трьох таблиць: вивести імена службовців, клієнтів і суми операцій, які були здійснені.

The image shows a database application with three windows. The top-left window, titled 'S : таблиця', displays a table of employees. The top-right window, titled 'C : таблиця', displays a table of clients. The bottom window shows the result of a join query between the two tables. Arrows indicate the mapping of columns from the source tables to the result table.

SNum	SName	City	Age	Comm
101	Петя	Харьков	28	0,12
102	Саша	Киев	24	0,13
103	Миша	Харьков	25	0,11
104	Рома	Москва	20	0,15
107	Андрей	Киев	25	0,1
108	Миша	Харьков	22	0,14
109	Саша	Бабаи	20	0,14
111	Сергей	Киев		
112	Андрей	Минск		

  

CNum	CName	City	Rating	SNum
201	Наташа	Харьков	100	101
202	Жора	Осло	200	108
203	Рома	Париж	150	111
204	Рома	Харьков	200	102
206	Миша	Киев	250	101
207	Клим	Харьков	200	103
208	Саша	Рим	300	105
211	Петя	Киев	300	101
212	Ваня	Харьков	200	109

  

SName	CName
Петя	Наташа
Петя	Миша
Петя	Петя
Саша	Рома
Миша	Клим
Миша	Жора
Саша	Ваня
Сергей	Рома

```
SELECT S.Sname, C.Cname, Summa  
FROM S,C,O  
WHERE S.Snum=C.Snum AND C. Cnum=O.Cnum;
```

	SName	CName	Summa
	Петя	Наташа	150
	Сергей	Рома	350
	Сергей	Рома	200
	Петя	Миша	250
	Миша	Клим	550
	Рома	Саша	100
	Рома	Саша	300
▶	Петя	Петя	400
	Саша	Ваня	650

- Дякую за увагу!