

Конспект з “Теорії баз даних”

Зміст

| | |
|-------------------------------|--------|
| Теорія загальних понять | 2 ст. |
| Оператор SELECT | 3 ст. |
| Коментарі | 5 ст. |
| Псевдонім AS (alias) | 6 ст. |
| Ключове слово WHERE | 7 ст. |
| Оператори AND, OR і NOT | 11 ст. |
| Функція CONCAT | 15 ст. |
| Арифметичні дії | 16 ст. |
| Значення NULL | 17 ст. |
| Оператор LIKE | 19 ст. |
| Оператором BETWEEN | 22 ст. |
| Оператор IN | 24 ст. |
| Оператора ORDER BY | 26 ст. |
| Оператор TOP | 28 ст. |
| Оператор DISTINCT | 30 ст. |
| Функція COUNT | 31 ст. |
| Функції MIN, MAX | 33 ст. |
| Функція SUM | 34 ст. |
| Функція AVG | 34 ст. |
| Оператор GROUP BY | 35 ст. |
| Оператор INSERT | 36 ст. |
| Оператор UPDATE | 38 ст. |
| Оператор DELETE | 39 ст. |
| Оператора JOIN | 40 ст. |



База даних - це організована структура, призначена для зберігання, зміни й обробки взаємопов'язаної інформації. Вона може включати структуровані, напів структуровані або неструктуровані дані, такі як текст, числа, зображення, відео тощо.

База даних дозволяє зберігати, оновлювати, керувати та отримувати доступ до великої кількості даних. Вона може бути використана для зберігання інформації про різні сутності, такі як клієнти, товариші, транзакції, студенти, співробітники тощо.

СУБД (система управління базами даних) - це програмне забезпечення, яке дозволяє створювати, керувати та управляти базами даних. Вона забезпечує зручний інтерфейс для взаємодії з базою даних, включаючи створення, оновлення та видалення даних, запити та аналіз даних.

СУБД дозволяє організувати структуру бази даних, визначати типи даних, встановлювати взаємозв'язки між даними, застосовувати обмеження та правила до даних. Вона також надає механізми для забезпечення безпеки даних, резервного копіювання, відновлення та інших операцій для керування базою даних.

Популярними СУБД є **MySQL, Oracle Database, Microsoft SQL Server, PostgreSQL, MongoDB** та багато інших, які надають різні можливості та функціонал для різних потреб користувачів.

Основні моделі баз даних, які використовують для організації та управління даними, включають наступні:

Реляційна модель бази даних (модель реляційної бази даних): це найпоширеніша модель, в якій дані представлені у вигляді таблиці з рядками (кортежами) та стовпцями (атрибутами). Реляційна модель використовує ключі для інсталяції взаємозв'язків між таблицями та забезпечує структуроване зберігання даних.

Мережна модель бази даних (Network Database Model): ця модель використовує систему зв'язків для організації даних. Вона передбачає використання записів (сегментів) та взаємних зв'язків між ними. Мережна модель була популярна в ранніх системах управління базами даних, але зараз втратила свою популярність.

Ієрархічна модель бази даних (Hierarchical Database Model): Ця модель організовує дані в ряді ієрархії батьківських та дочірніх записів. Кожен запис може мати лише одного батька, але кілька дочірніх записів.

Ієрархічна модель також була розширена в ранніх системах баз даних, особливо в галузі обробки даних з великими ієрархічними структурами.

Об'єктно-орієнтована модель бази даних (Object-Oriented Database Model):

Ця модель базується на об'єктно-орієнтованому програмуванні та дозволяє зберегти об'єкти, їх властивості та взаємозв'язки між ними.

Об'єктно-орієнтована модель надає більшу гнучку структуру для зберігання складських даних, таких як зображення, звуки, відео та інші мультимедійні об'єкти.

Кожна модель бази даних має свої переваги та застосування в залежності від потреб користувача та характеристик даних. У багатьох випадках використовуються реляційні бази даних через їх широкую підтримку та надійність, але інші моделі також мають свою важливу роль у конкретних сферах.

SQL (Structured Query Language - мова структурованих запитів) є мовою програмування та запитів, яка використовується для взаємодії з реляційними базами даних. Вона використовується для створення, модифікації та керування даними у базі даних.

Під час роботи з базами даних можна виконувати 4 базові операції **CRUD**:

- **create** (створювати);
- **read** (читати);
- **update** (оновлювати);
- **delete** (видаляти).

Оператор **SELECT** є одним з основних операторів у SQL і використовується для вибору даних з бази даних. Він дозволяє виконувати запити та отримувати результати у вигляді набору записів, які відповідають вказаним умовам.

Ключове слово **FROM** використовується для вказівки джерела отримання даних, після нього прописується назву таблиці або список таблиць. Основний синтаксис оператора SELECT виглядає наступним чином:

```
SELECT column_1, column_2, ...  
FROM table_name;
```

SELECT вказує на те, що ми хочемо вибрати певні стовпці з бази даних. column1, column2, ... вказують на назви стовпців, які ми хочемо вибрати. Якщо необхідно вибрати всі стовпці, можна використовувати *.

FROM table_name вказує на таблицю, з якої ми хочемо вибрати дані.

Приклад. Нам потрібно вибрати всю інформацію з таблиці Team. Для цього використовуємо *

```
SELECT *  
FROM team;
```

| | id | name | surname | gender | position | salary | birth_date | city | school |
|----|----|-----------|------------|--------|----------------------------|---------|------------|-----------------|--------------------------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000.00 | 1985-02-01 | Kyiv | IT Step Academy |
| 2 | 2 | Anna | Shevchenko | female | Software Developer | 2000.00 | 1999-02-28 | Lviv | NULL |
| 3 | 3 | Olena | Bondarenko | female | System Administrator | 1700.00 | 1983-08-19 | Kyiv | Lviv IT School |
| 4 | 4 | Petro | Tkachenko | male | Network Engineer | 1700.00 | 1990-06-02 | Dnipro | UNIT Factory |
| 5 | 5 | Artem | Kovalchuk | male | Web Developer | 2100.00 | 1988-12-29 | Odesa | NULL |
| 6 | 6 | Denis | Kravchenko | male | Data Analyst | 2500.00 | 1994-01-14 | Zhytomyr | Data Science UA |
| 7 | 7 | Anastasia | Lysenko | female | Project Manager | 2400.00 | 1996-10-07 | Kyiv | GoIT |
| 8 | 8 | Julia | Romanenko | female | Cybersecurity Specialist | 3000.00 | 1985-03-22 | Ivano-Frankivsk | SoftServe IT Academy |
| 9 | 9 | Alina | Miller | female | Database Administrator | 2200.00 | 1982-05-31 | Sumy | Beetroot Academy |
| 10 | 10 | Katerina | Petrenko | female | IT Consultant | 2800.00 | 1986-09-10 | Vinnytsia | SkillUp |
| 11 | 11 | Darina | Ivanenko | female | Cloud Computing Specialist | 3000.00 | 1997-11-26 | Zaporizhzhia | IT Step Academy |
| 12 | 12 | Mark | Griorenko | male | AI Engineer | 3200.00 | 1992-08-03 | Kharkiv | Cisco Networking Academy |

Приклад. Щоб вибрати всю інформацію з таблиці Team, нам потрібно вказати назви стовпців, які ми хочемо вибрати:

```
SELECT id, name, surname, gender, position, salary, birth_date, city, school  
FROM team;
```

| | id | name | surname | gender | position | salary | birth_date | city | school |
|----|----|-----------|------------|--------|----------------------------|---------|------------|-----------------|--------------------------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000.00 | 1985-02-01 | Kyiv | IT Step Academy |
| 2 | 2 | Anna | Shevchenko | female | Software Developer | 2000.00 | 1999-02-28 | Lviv | NULL |
| 3 | 3 | Olena | Bondarenko | female | System Administrator | 1700.00 | 1983-08-19 | Kyiv | Lviv IT School |
| 4 | 4 | Petro | Tkachenko | male | Network Engineer | 1700.00 | 1990-06-02 | Dnipro | UNIT Factory |
| 5 | 5 | Artem | Kovalchuk | male | Web Developer | 2100.00 | 1988-12-29 | Odesa | NULL |
| 6 | 6 | Denis | Kravchenko | male | Data Analyst | 2500.00 | 1994-01-14 | Zhytomyr | Data Science UA |
| 7 | 7 | Anastasia | Lysenko | female | Project Manager | 2400.00 | 1996-10-07 | Kyiv | GoIT |
| 8 | 8 | Julia | Romanenko | female | Cybersecurity Specialist | 3000.00 | 1985-03-22 | Ivano-Frankivsk | SoftServe IT Academy |
| 9 | 9 | Alina | Miller | female | Database Administrator | 2200.00 | 1982-05-31 | Sumy | Beetroot Academy |
| 10 | 10 | Katerina | Petrenko | female | IT Consultant | 2800.00 | 1986-09-10 | Vinnytsia | SkillUp |
| 11 | 11 | Darina | Ivanenko | female | Cloud Computing Specialist | 3000.00 | 1997-11-26 | Zaporizhzhia | IT Step Academy |
| 12 | 12 | Mark | Griorenko | male | AI Engineer | 3200.00 | 1992-08-03 | Kharkiv | Cisco Networking Academy |

У SQL **коментарі** використовуються для додавання пояснень або інформації, які не впливають на виконання запиту або інших операцій. Коментарі не виконуються як частина запиту і ігноруються системою управління базами даних.

SQL підтримує два типи коментарів:

Однорядкові коментарі: Використовуються для додавання коментарів, які охоплюють лише один рядок. Однорядкові коментарі починаються з подвійного дефісу (--) і продовжуються до кінця рядка. Все, що знаходиться після коментарів у рядку, не виконується.

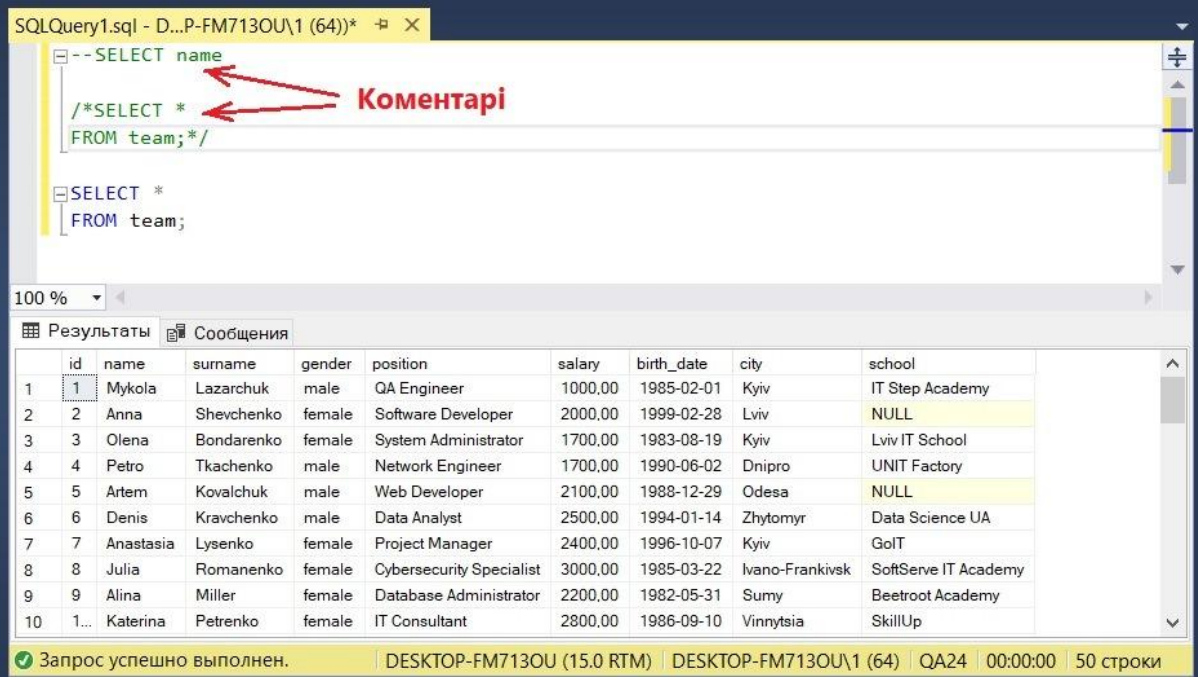
Багаторядкові коментарі: Використовуються для додавання коментарів, які можуть охоплювати кілька рядків. Багаторядкові коментарі починаються з символів /* і закінчуються символами */ (**/*коментар*/**). Все, що знаходиться між цими символами, не виконується.

Приклад:

```
--SELECT name
```

```
/*SELECT *  
FROM team;*/
```

```
SELECT *  
FROM team;
```



The screenshot shows a SQL query editor window titled "SQLQuery1.sql - D:\P-FM713OU\1 (64)*". The editor contains three SQL queries. The first two are comments: `--SELECT name` and `/*SELECT * FROM team;*/`. Red arrows point from the word "Коментарі" to these two lines. The third query is `SELECT * FROM team;`. Below the editor, the "Результаты" (Results) tab is active, displaying a table with 10 rows and 10 columns: id, name, surname, gender, position, salary, birth_date, city, and school. The first row is highlighted with a dashed border. The status bar at the bottom indicates "Запрос успешно выполнен." (Query successfully executed).

| | id | name | surname | gender | position | salary | birth_date | city | school |
|----|----|-----------|------------|--------|--------------------------|---------|------------|-----------------|----------------------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000.00 | 1985-02-01 | Kyiv | IT Step Academy |
| 2 | 2 | Anna | Shevchenko | female | Software Developer | 2000.00 | 1999-02-28 | Lviv | NULL |
| 3 | 3 | Olena | Bondarenko | female | System Administrator | 1700.00 | 1983-08-19 | Kyiv | Lviv IT School |
| 4 | 4 | Petro | Tkachenko | male | Network Engineer | 1700.00 | 1990-06-02 | Dnipro | UNIT Factory |
| 5 | 5 | Artem | Kovalchuk | male | Web Developer | 2100.00 | 1988-12-29 | Odesa | NULL |
| 6 | 6 | Denis | Kravchenko | male | Data Analyst | 2500.00 | 1994-01-14 | Zhytomyr | Data Science UA |
| 7 | 7 | Anastasia | Lysenko | female | Project Manager | 2400.00 | 1996-10-07 | Kyiv | GoIT |
| 8 | 8 | Julia | Romanenko | female | Cybersecurity Specialist | 3000.00 | 1985-03-22 | Ivano-Frankivsk | SoftServe IT Academy |
| 9 | 9 | Alina | Miller | female | Database Administrator | 2200.00 | 1982-05-31 | Sumy | Beetroot Academy |
| 10 | 10 | Katerina | Petrenko | female | IT Consultant | 2800.00 | 1986-09-10 | Vinnitsia | SkillUp |

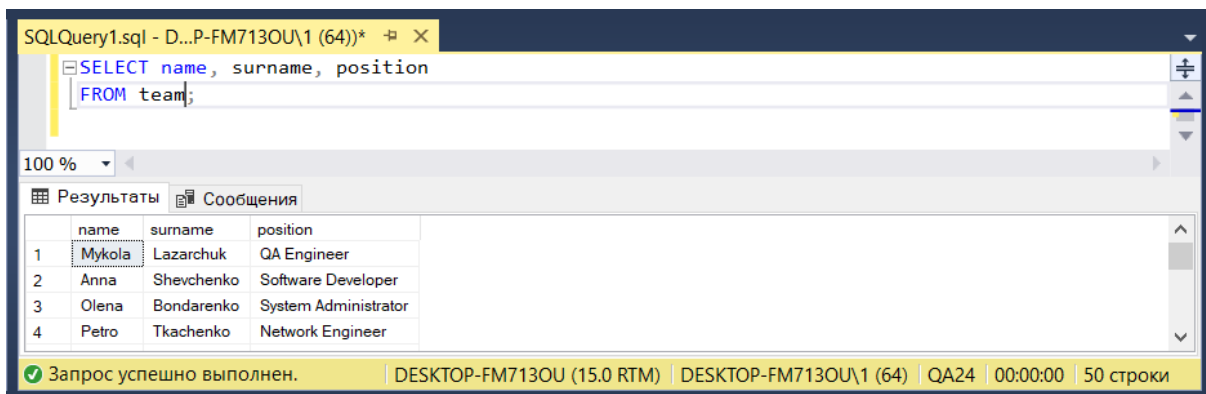
Псевдонім (alias) — це тимчасова назва таблиці чи стовпця. З його допомогою можна скоротити кількість слів, які потрібні для виконання запиту.

Зазвичай, псевдоніми створюють для складних запитів, щоб їх було легше читати. Щоб створити псевдонім, потрібно написати ключове слово **AS**, а далі вказати назву псевдоніму.

Зверни увагу: псевдонім існує тільки на час виконання запиту.

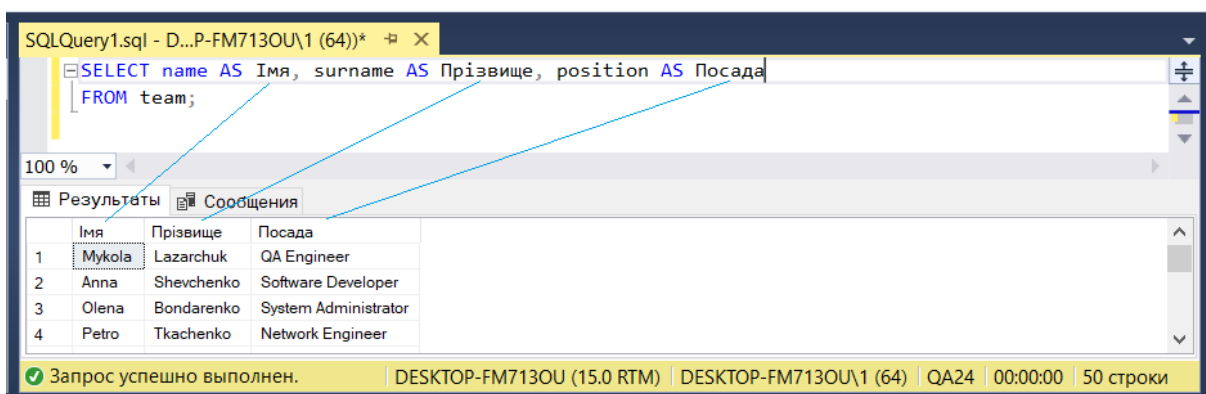
Приклад. Напишемо запит, який поверне ім'я, прізвище та посаду:

```
SELECT name, surname, position  
FROM team;
```



А тепер задамо псевдоніми для name, surname, position.

```
SELECT name AS Імя, surname AS Прізвище, position AS Посада  
FROM team;
```



Ключове слово **WHERE** використовується для фільтрації рядків, які повертаються в результаті запиту SELECT. Воно дозволяє встановити умову, яку повинні задовольняти дані рядки, щоб бути включеними у вибірку.

Синтаксис операції SELECT з ключовим словом WHERE виглядає наступним чином:

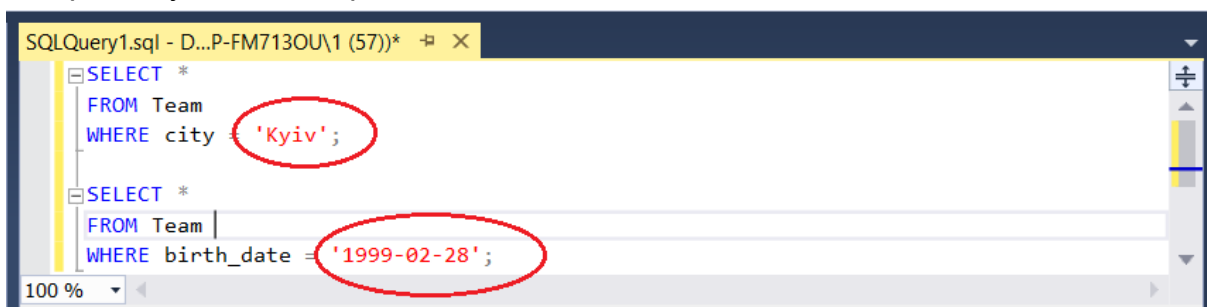
```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

condition - представляє логічний вираз, який описує умову фільтрації.

Для того, щоб в SQL-запиті вказати необхідні обмеження, у мові T-SQL використовуються наступні оператори порівняння:

- = — дорівнює;
- != — не дорівнює;
- <> — не дорівнює;
- > — більше, ніж;
- < — менше, ніж;
- >= — більше або дорівнює;
- <= — менше або дорівнює.

Коли ми працюємо з рядковими типами або типами дати та часу, потрібно використовувати одинарні лапки — ' '.



Приклад. Вивести імена, прізвища та посаду всіх тестувальників (QA engineer).

```
SELECT name, surname, position  
FROM team  
WHERE position = 'qa engineer';
```

SQLQuery1.sql - D...P-FM713OU\1 (57))* ✕

```
SELECT name, surname, position
FROM team
WHERE position = 'qa engineer';
```

100 %

Результаты Сообщения

| | name | surname | position |
|---|---------|-----------|-------------|
| 1 | Mykola | Lazarchuk | QA Engineer |
| 2 | Evgenia | Kravtsova | QA Engineer |
| 3 | Olga | Kuzmenko | QA Engineer |

Запрос у... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 3 строки

Знак 1 BCT

Приклад. Потрібно вивести імена, прізвища та міста співробітників, які не мешкають в місті Києві.

```
SELECT name, surname, city
FROM team
WHERE city != 'kyiv';
```

SQLQuery1.sql - D...P-FM713OU\1 (57))* ✕

```
SELECT name, surname, city
FROM team
WHERE city != 'kyiv';
```

100 %

Результаты Сообщения

| | name | surname | city |
|---|-------|------------|-----------------|
| 1 | Anna | Shevchenko | Lviv |
| 2 | Petro | Tkachenko | Dnipro |
| 3 | Artem | Kovalchuk | Odesa |
| 4 | Denis | Kravchenko | Zhytomyr |
| 5 | Julia | Romanenko | Ivano-Frankivsk |
| 6 | Alina | Miller | Sumy |

Запрос у... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 36 строки

Знак 22 BCT

```
SELECT name, surname, city
FROM team
WHERE city <> 'kyiv';
```


SQLQuery1.sql - D...P-FM713OU\1 (57))*

```

SELECT name, surname, city
FROM team
WHERE city <> 'kyiv';

```

100 %

Результаты Сообщения

| | name | surname | city |
|---|-------|------------|-----------------|
| 1 | Anna | Shevchenko | Lviv |
| 2 | Petro | Tkachenko | Dnipro |
| 3 | Artem | Kovalchuk | Odesa |
| 4 | Denis | Kravchenko | Zhytomyr |
| 5 | Julia | Romanenko | Ivano-Frankivsk |
| 6 | Alina | Miller | Sumy |

Запрос у... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 36 строки

Знак 1 BCT

Приклад. Потрібно вивести імена, прізвища, посаду та зарплату співробітників, у яких зарплата більше 2000.

```

SELECT name, surname, city, salary
FROM team
WHERE salary > 2000;

```

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```

SELECT name, surname, city, salary
FROM team
WHERE salary > 2000;

```

100 %

Результаты Сообщения

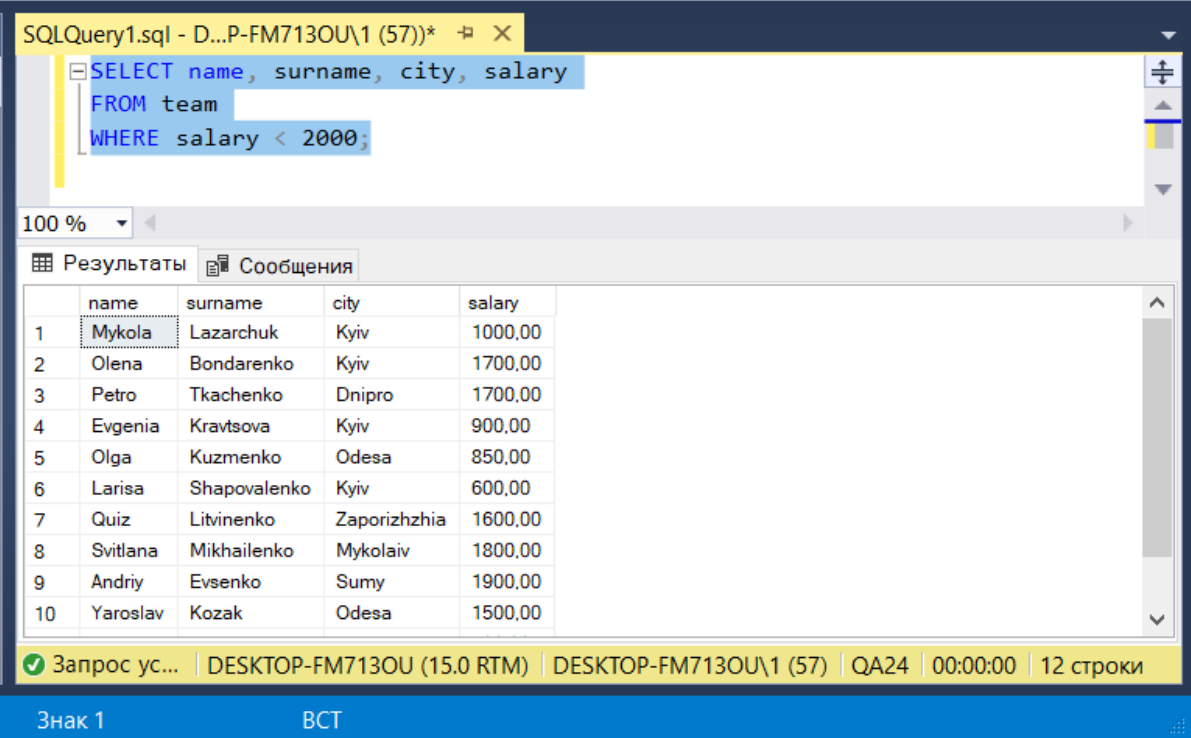
| | name | surname | city | salary |
|----|-----------|------------|-----------------|---------|
| 1 | Artem | Kovalchuk | Odesa | 2100.00 |
| 2 | Denis | Kravchenko | Zhytomyr | 2500.00 |
| 3 | Anastasia | Lysenko | Kyiv | 2400.00 |
| 4 | Julia | Romanenko | Ivano-Frankivsk | 3000.00 |
| 5 | Alina | Miller | Sumy | 2200.00 |
| 6 | Katerina | Petrenko | Vinnytsia | 2800.00 |
| 7 | Darina | Ivanenko | Zaporizhzhia | 3000.00 |
| 8 | Mark | Grigorenko | Kharkiv | 3200.00 |
| 9 | Anton | Sidorenko | Donetsk | 2700.00 |
| 10 | Alla | Fedorenko | Kyiv | 3300.00 |

Запрос ус... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 33 строки

Знак 20 BCT

Приклад. Потрібно вивести імена, прізвища, посаду та зарплату співробітників, у яких зарплата менше 2000.

```
SELECT name, surname, city, salary
FROM team
WHERE salary < 2000;
```



The screenshot shows a SQL query window titled 'SQLQuery1.sql - D...P-FM713OU\1 (57))' containing the following query:

```
SELECT name, surname, city, salary
FROM team
WHERE salary < 2000;
```

Below the query window, the 'Results' tab is active, displaying a table with 10 rows of data. The table has columns: name, surname, city, and salary. The data is as follows:

| | name | surname | city | salary |
|----|----------|--------------|--------------|---------|
| 1 | Mykola | Lazarchuk | Kyiv | 1000.00 |
| 2 | Olena | Bondarenko | Kyiv | 1700.00 |
| 3 | Petro | Tkachenko | Dnipro | 1700.00 |
| 4 | Evgenia | Kravtsova | Kyiv | 900.00 |
| 5 | Olga | Kuzmenko | Odesa | 850.00 |
| 6 | Larisa | Shapovalenko | Kyiv | 600.00 |
| 7 | Quiz | Litvinenko | Zaporizhzhia | 1600.00 |
| 8 | Svitlana | Mikhailenko | Mykolaiv | 1800.00 |
| 9 | Andriy | Evsenko | Sumy | 1900.00 |
| 10 | Yaroslav | Kozak | Odesa | 1500.00 |

At the bottom of the window, a status bar shows: 'Знак 1' (Sign 1), 'ВСТ' (VST), and '12 строки' (12 rows).

Приклад. Потрібно вивести імена, прізвища, посаду та зарплату співробітників, у яких зарплата більше або дорівнює 2000.

```
SELECT name, surname, city, salary
FROM team
WHERE salary >= 2000;
```

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```

SELECT name, surname, city, salary
FROM team
WHERE salary >= 2000;

```

100 %

Результаты Сообщения

| | name | surname | city | salary |
|----|-----------|------------|-----------------|---------|
| 1 | Anna | Shevchenko | Lviv | 2000,00 |
| 2 | Artem | Kovalchuk | Odesa | 2100,00 |
| 3 | Denis | Kravchenko | Zhytomyr | 2500,00 |
| 4 | Anastasia | Lysenko | Kyiv | 2400,00 |
| 5 | Julia | Romanenko | Ivano-Frankivsk | 3000,00 |
| 6 | Alina | Miller | Sumy | 2200,00 |
| 7 | Katerina | Petrenko | Vinnytsia | 2800,00 |
| 8 | Darina | Ivanenko | Zaporizhzhia | 3000,00 |
| 9 | Mark | Grigorenko | Kharkiv | 3200,00 |
| 10 | Anton | Sidorenko | Donetsk | 2700,00 |

Запрос ус... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 38 строки

Знак 16 ВСТ

Оператори **AND**, **OR** і **NOT** є логічними операторами, які використовуються в SQL та інших мовах програмування для поєднання та зміни логічних умов.

Оператор **AND** використовується для поєднання двох або більше умов, і результат буде істинним (true) лише тоді, коли всі умови, які його складають, є істинними. Наприклад, якщо ми маємо умови A і B, то умова A AND B буде істинною лише тоді, коли обидві умови A і B є істинними.

Оператор **OR** використовується для поєднання двох або більше умов, і результат буде істинним (true), якщо хоча б одна з умов, які його складають, є істинною. Наприклад, якщо ми маємо умови A і B, то умова A OR B буде істинною, якщо хоча б одна з умов A або B є істинною.

Оператор **NOT** використовується для зміни логічного значення умови. Він повертає істину, якщо умова є хибною (false), і хибу, якщо умова є істинною (true). Оператор NOT можна використовувати для заперечення умови.

Приклад. Вивести прізвища співробітників, місто проживання та навчальний заклад, які проживають в місті Києві і які навчались в IT Step Academy.

```

SELECT surname, city, school
FROM team
WHERE city = 'Kyiv' AND school = 'IT Step Academy';

```

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```

SELECT surname, city, school
FROM team
WHERE city = 'Kyiv' AND school = 'IT Step Academy';

```

100 %

Результаты Сообщения

| | surname | city | school |
|---|------------|------|-----------------|
| 1 | Lazarchuk | Kyiv | IT Step Academy |
| 2 | Marchuk | Kyiv | IT Step Academy |
| 3 | Koshelenko | Kyiv | IT Step Academy |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 3 строки

Приклад. Вивести прізвища співробітників, місто проживання та навчальний заклад, які проживають в місті Києві, або які навчались в IT Step Academy.

```

SELECT surname, city, school
FROM team
WHERE city = 'Kyiv' OR school = 'IT Step Academy';

```

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```

SELECT surname, city, school
FROM team
WHERE city = 'Kyiv' OR school = 'IT Step Academy';

```

100 %

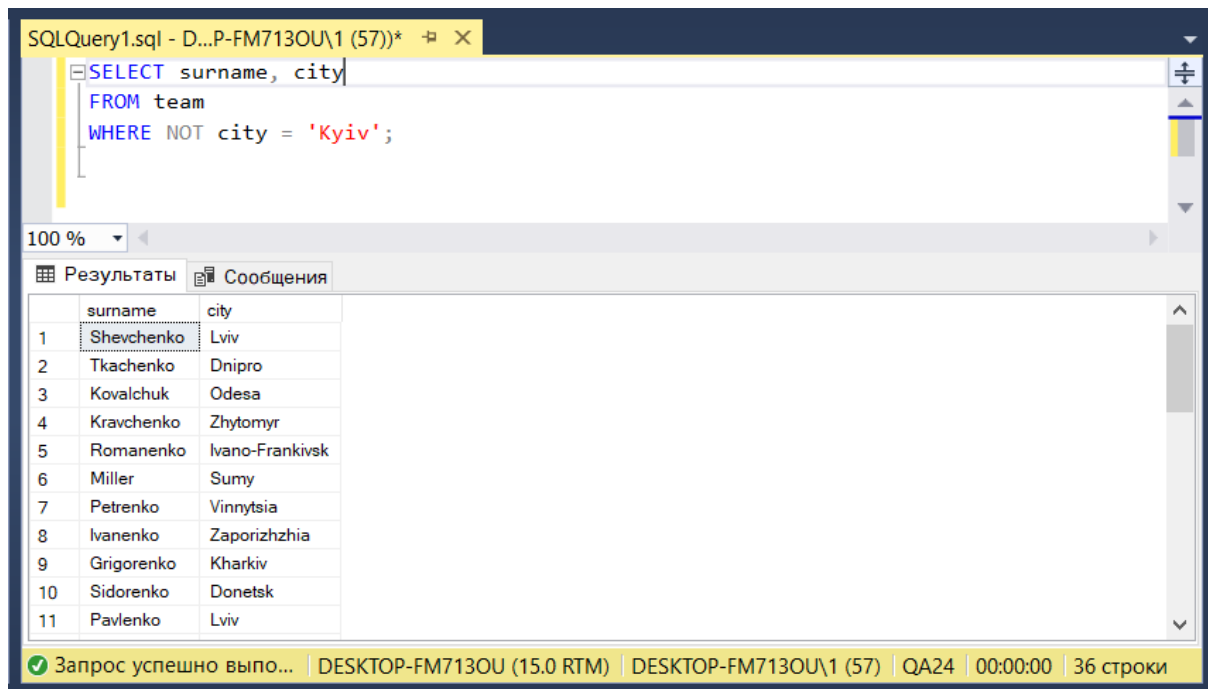
Результаты Сообщения

| | surname | city | school |
|----|--------------|--------------|-----------------------|
| 1 | Lazarchuk | Kyiv | IT Step Academy |
| 2 | Bondarenko | Kyiv | Lviv IT School |
| 3 | Lysenko | Kyiv | GoIT |
| 4 | Ivanenko | Zaporizhzhia | IT Step Academy |
| 5 | Fedorenko | Kyiv | Binary Studio Academy |
| 6 | Kravtsova | Kyiv | IT Education Academy |
| 7 | Shapovalenko | Kyiv | Best IT Center |
| 8 | Gerasimenko | Kherson | IT Step Academy |
| 9 | Gordienko | Kyiv | NULL |
| 10 | Karpenko | Kyiv | GoIT |
| 11 | Levchenko | Kyiv | Lviv IT School |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 17 строки

Приклад. Вивести прізвища, та міста співробітників, які мешкають не в місті Києві.

```
SELECT surname, city  
FROM team  
WHERE NOT city = 'Kyiv';
```



Мова SQL не чутлива до регістру. Але ключові слова прийнято писати великими літерами.

Поганий приклад:

```
select *  
from Team  
where gender = 'male';
```

Гарний приклад:

```
SELECT *  
FROM Team  
WHERE gender = 'male';
```

Також, кожну логічну частину твого запиту потрібно писати з нового рядка.

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```
SELECT * FROM Team WHERE gender = 'male';
```

Такий код незрозумілий

100 %

Результаты Сообщения

| | id | name | surname | gender | position | salary | birth_date | city | school |
|----|----|-----------|------------|--------|----------------------------|---------|------------|-----------|--------------------------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000,00 | 1985-02-01 | Kyiv | IT Step Academy |
| 2 | 4 | Petro | Tkachenko | male | Network Engineer | 1700,00 | 1990-06-02 | Dnipro | UNIT Factory |
| 3 | 5 | Artem | Kovalchuk | male | Web Developer | 2100,00 | 1988-12-29 | Odesa | NULL |
| 4 | 6 | Denis | Kravchenko | male | Data Analyst | 2500,00 | 1994-01-14 | Zhytomyr | Data Science UA |
| 5 | 12 | Mark | Grigorenko | male | AI Engineer | 3200,00 | 1992-08-03 | Kharkiv | Cisco Networking Academy |
| 6 | 13 | Anton | Sidorenko | male | UX/UI Designer | 2700,00 | 1984-12-11 | Donetsk | EPAM Systems |
| 7 | 27 | Oleksandr | Karpenko | male | Data Analyst | 2000,00 | 1994-07-16 | Kyiv | GoIT |
| 8 | 28 | Andriy | Evsenko | male | Database Administrator | 1900,00 | 1986-09-09 | Sumy | SkillUp |
| 9 | 29 | Ivan | Kozlovenko | male | UX/UI Designer | 2000,00 | 1982-12-21 | Lutsk | Brain Academy |
| 10 | 30 | Mikhailo | Ivanov | male | Cloud Computing Specialist | 2400,00 | 1985-02-14 | Chernihiv | Data Science UA |
| 11 | 31 | Dmitro | Bondarenko | male | IT Consultant | 2500,00 | 1999-06-30 | Kharkiv | Cisco Networking Academy |
| 12 | 32 | Serhiy | Kurashov | male | UX/UI Designer | 2000,00 | 1984-08-05 | Vinnytsia | IT Step Academy |

Запрос успешно выполнено | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 25 строки

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```
SELECT *  
FROM Team  
WHERE gender = 'female';
```

Такий код зрозумілий

100 %

Результаты Сообщения

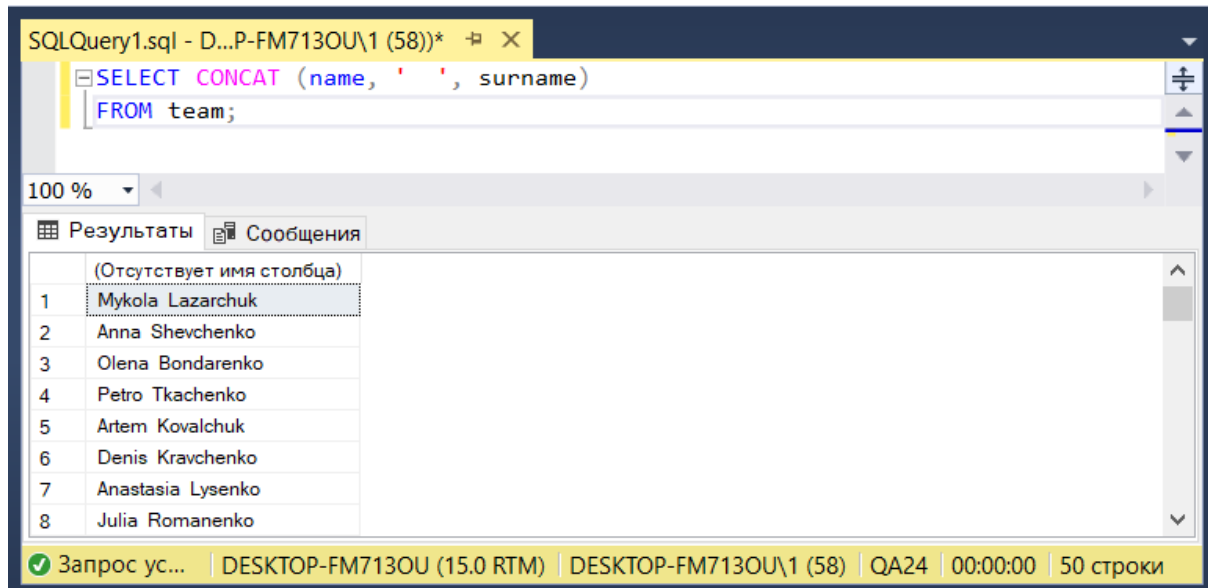
| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|-----------|------------|--------|----------------------------|---------|------------|-----------------|-----------------------|
| 1 | 2 | Anna | Shevchenko | female | Software Developer | 2000,00 | 1999-02-28 | Lviv | NULL |
| 2 | 3 | Olena | Bondarenko | female | System Administrator | 1700,00 | 1983-08-19 | Kyiv | Lviv IT School |
| 3 | 7 | Anastasia | Lysenko | female | Project Manager | 2400,00 | 1996-10-07 | Kyiv | GoIT |
| 4 | 8 | Julia | Romanenko | female | Cybersecurity Specialist | 3000,00 | 1985-03-22 | Ivano-Frankivsk | SoftServe IT Academy |
| 5 | 9 | Alina | Miller | female | Database Administrator | 2200,00 | 1982-05-31 | Sumy | Beetroot Academy |
| 6 | 10 | Katerina | Petrenko | female | IT Consultant | 2800,00 | 1986-09-10 | Vinnytsia | SkillUp |
| 7 | 11 | Darina | Ivanenko | female | Cloud Computing Specialist | 3000,00 | 1997-11-26 | Zaporizhzhia | IT Step Academy |
| 8 | 14 | Alla | Fedorenko | female | DevOps Engineer | 3300,00 | 1981-02-18 | Kyiv | Binary Studio Academy |

Запрос успешно выполнено | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 25 строки

Функція **CONCAT** об'єднує два або більше рядків разом

Приклад. Напишемо запит який об'єднає прізвище та ім'я разом.

```
SELECT CONCAT (name, ' ', surname)
FROM team;
```



Під час написання SQL-запитів можна формувати нові стовпці у віртуальній таблиці, з'єднуючи значення декількох реальних стовпців за допомогою символа +, у цьому випадку заголовок такого стовпця буде (No column name), якщо такий стан речей вас не влаштовує, то ви можете задати псевдонім (alias) цьому стовпцю за допомогою оператора AS. Наприклад, з'єднавши в SQL-запиті ім'я та прізвище студента через пробіл, вкажемо цьому стовпцю псевдонім FullName.

Приклад. Нам потрібно об'єднати стовпці name та surname і вказати нову назву стовпця (FullName). Наступний запит виведе ім'я та прізвище (в одному стовпці) та посаду.

```
SELECT name + ' ' + surname AS FullName, position
FROM team;
```

SQLQuery1.sql - D...P-FM713OU\1 (57))* ✕

```
SELECT name + ' ' + surname AS FullName, position
FROM team;
```

100 %

Результаты Сообщения

| | FullName | position |
|---|-------------------|----------------------|
| 1 | Mykola Lazarchuk | QA Engineer |
| 2 | Anna Shevchenko | Software Developer |
| 3 | Olena Bondarenko | System Administrator |
| 4 | Petro Tkachenko | Network Engineer |
| 5 | Artem Kovalchuk | Web Developer |
| 6 | Denis Kravchenko | Data Analyst |
| 7 | Anastasia Lysenko | Project Manager |

✓ Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 50 строки

При написанні SQL-запиту ви можете змінювати дані у віртуальній таблиці, застосовуючи різні **арифметичні дії**, і це ніяк не позначиться на реальних даних. Наприклад. Нам потрібно дізнатися якою буде зарплата працівників, якщо додати 20%.

Звичайна зарплата

SQLQuery2.sql - D...P-FM713OU\1 (52)) SQLQuery1.sql - D...P-FM713OU\1 (57))* ✕

```
SELECT name, surname, salary
FROM team;
```

100 %

Результаты Сообщения

| | name | surname | salary |
|---|-----------|------------|---------|
| 1 | Mykola | Lazarchuk | 1000.00 |
| 2 | Anna | Shevchenko | 2000.00 |
| 3 | Olena | Bondarenko | 1700.00 |
| 4 | Petro | Tkachenko | 1700.00 |
| 5 | Artem | Kovalchuk | 2100.00 |
| 6 | Denis | Kravchenko | 2500.00 |
| 7 | Anastasia | Lysenko | 2400.00 |

✓ Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 50 строки

Зарплата + 20%

SELECT name, surname, salary*1.2 **AS** [salary plus 20 %]
FROM team;

SQLQuery2.sql - D...P-FM713OU\1 (52)) SQLQuery1.sql - D...P-FM713OU\1 (57))*

```
SELECT name, surname, salary*1.2 AS [salary plus 20 %]
FROM team;
```

100 %

Результаты Сообщения

| | name | surname | salary plus 20 % |
|---|-----------|------------|------------------|
| 1 | Mykola | Lazarchuk | 1200.00000 |
| 2 | Anna | Shevchenko | 2400.00000 |
| 3 | Olena | Bondarenko | 2040.00000 |
| 4 | Petro | Tkachenko | 2040.00000 |
| 5 | Artem | Kovalchuk | 2520.00000 |
| 6 | Denis | Kravchenko | 3000.00000 |
| 7 | Anastasia | Lysenko | 2880.00000 |

Запрос успешно выполне... DESKTOP-FM713OU (15.0 RTM) DESKTOP-FM713OU\1 (57) QA24 00:00:00 50 строки

Значення **NULL** використовується для зазначення того, що значення в базі даних не існує. Поле зі значенням NULL — це поле без значення. NULL це не те саме, що і 0 або поле з пробілом (порожній рядок і 0 представляють фактичні значення).

Приклад значення NULL (невідомо в яких ІТ школах навчались співробітники, або взагалі не навчались в ІТ школах)

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```
SELECT *
FROM team;
```

100 %

Результаты Сообщения

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|-----------|------------|--------|----------------------|---------|------------|----------|-----------------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000,00 | 1985-02-01 | Kyiv | IT Step Academy |
| 2 | 2 | Anna | Shevchenko | female | Software Developer | 2000,00 | 1999-02-28 | Lviv | NULL |
| 3 | 3 | Olena | Bondarenko | female | System Administrator | 1700,00 | 1983-08-19 | Kyiv | Lviv IT School |
| 4 | 4 | Petro | Tkachenko | male | Network Engineer | 1700,00 | 1990-06-02 | Dnipro | UNIT Factory |
| 5 | 5 | Artem | Kovalchuk | male | Web Developer | 2100,00 | 1988-12-29 | Odesa | NULL |
| 6 | 6 | Denis | Kravchenko | male | Data Analyst | 2500,00 | 1994-01-14 | Zhytomyr | Data Science UA |
| 7 | 7 | Anastasia | Lysenko | female | Project Manager | 2400,00 | 1996-10-07 | Kyiv | GoIT |

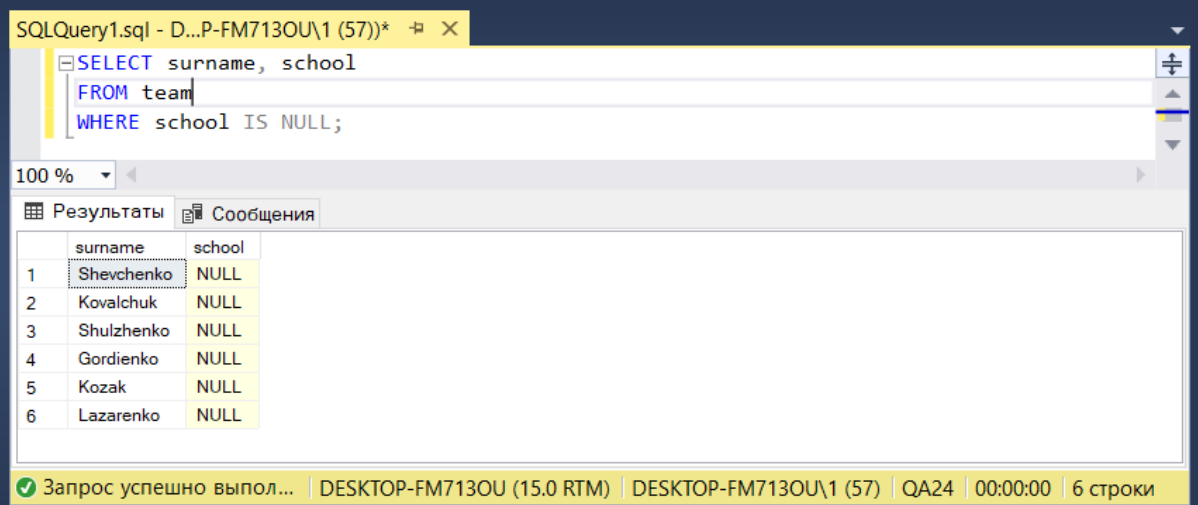
Значення NULL

Запрос успешно выполне... DESKTOP-FM713OU (15.0 RTM) DESKTOP-FM713OU\1 (57) QA24 00:00:00 50 строки

Щоб перевірити значення на NULL, потрібно використати оператори **IS NULL** та **IS NOT NULL**. Зверни увагу: ми не можемо використовувати оператори =, < та > для перевірки значення на NULL.

Приклад. Вивести прізвища співробітників, які не навчались в ІТ школах.

```
SELECT surname, school
FROM team
WHERE school IS NULL;
```



The screenshot shows a SQL Query Editor window with the following query:

```
SELECT surname, school
FROM team
WHERE school IS NULL;
```

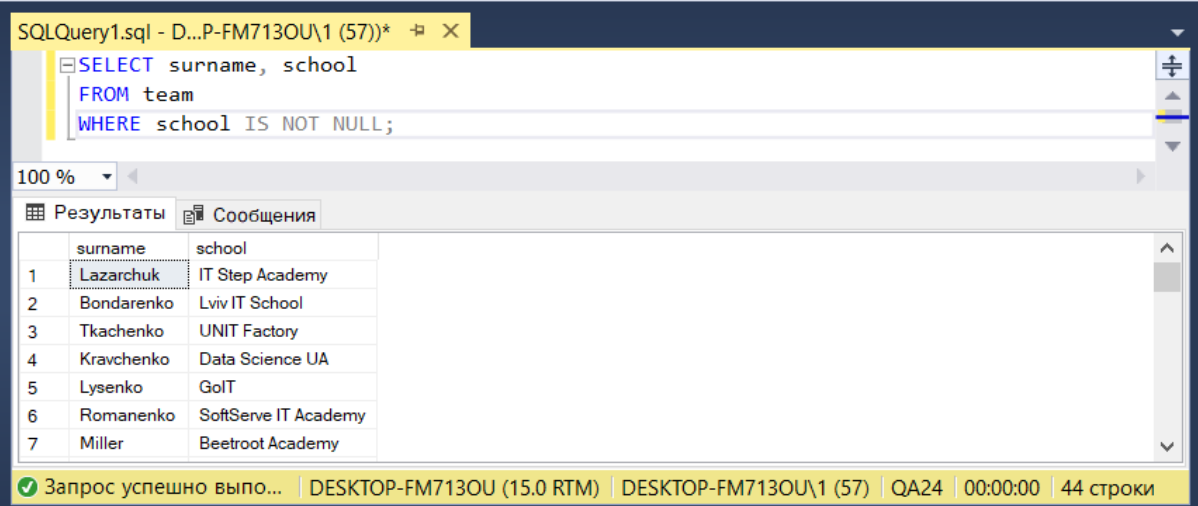
The results are displayed in a table with two columns: surname and school. The table contains 6 rows of data.

| | surname | school |
|---|------------|--------|
| 1 | Shevchenko | NULL |
| 2 | Kovalchuk | NULL |
| 3 | Shulzhenko | NULL |
| 4 | Gordienko | NULL |
| 5 | Kozak | NULL |
| 6 | Lazarenko | NULL |

The status bar at the bottom indicates: Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 6 строки

Приклад. Вивести прізвища співробітників які навчались в ІТ школах.

```
SELECT surname, school
FROM team
WHERE school IS NOT NULL;
```



The screenshot shows a SQL Query Editor window with the following query:

```
SELECT surname, school
FROM team
WHERE school IS NOT NULL;
```

The results are displayed in a table with two columns: surname and school. The table contains 7 rows of data.

| | surname | school |
|---|------------|----------------------|
| 1 | Lazarchuk | IT Step Academy |
| 2 | Bondarenko | Lviv IT School |
| 3 | Tkachenko | UNIT Factory |
| 4 | Kravchenko | Data Science UA |
| 5 | Lysenko | GoIT |
| 6 | Romanenko | SoftServe IT Academy |
| 7 | Miller | Beetroot Academy |

The status bar at the bottom indicates: Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 44 строки

Щоб знайти записи в таблиці, які відповідають певному шаблону, можна скористатися оператором **LIKE**. Наприклад, коли ми маємо лише фрагмент текстового значення.

Оператор LIKE використовується з двома знаками підстановки:

- % — відповідає будь-якій послідовності символів від 0 і більше;
- _ — відповідає будь-якому окремому символу.

| Оператор LIKE | Пояснення | Приклади |
|------------------|--|----------------------|
| LIKE 'a%' | рядок починається на а | apple, apricot |
| LIKE '%a' | рядок закінчується на а | Anna, Maria, Rita |
| LIKE '%ti%' | рядок містить ti у будь-якому місці | motivation |
| LIKE '_p%' | рядок містить р на другій позиції | apple, apricot |
| LIKE 'a_%' | рядок починається на а та має довжину 2 символи або більше | at, atom |

| | | |
|-------------|--|---------------|
| LIKE 'a__%' | рядок починається на а та має довжину 3 символи або більше | atom, all |
| LIKE 'a%e' | рядок починається на а та закінчується на е | apple, advice |

| | | |
|-----------------|---|---------------------|
| LIKE '%e%e%' | рядок містить як мінімум дві літери е | seed, eye, level |
| LIKE '____' | рядок має довжину 4 символи | atom, pear |
| LIKE '____%' | рядок має довжину 4 символи або більше | Emma, Helen |
| LIKE '__nn_' | рядок має довжину 5 символів та містить nn на 3 і 4 позиції | sunny, funny |

Приклад. Потрібно вивести всю інформацію про співробітників, у яких ім'я починається з літери А.

```
SELECT *
FROM team
WHERE name LIKE 'A%';
```

SQLQuery1.sql - D...\P-FM713OU\1 (57))*

```
SELECT *
FROM team
WHERE name LIKE 'A%';
```

100 %

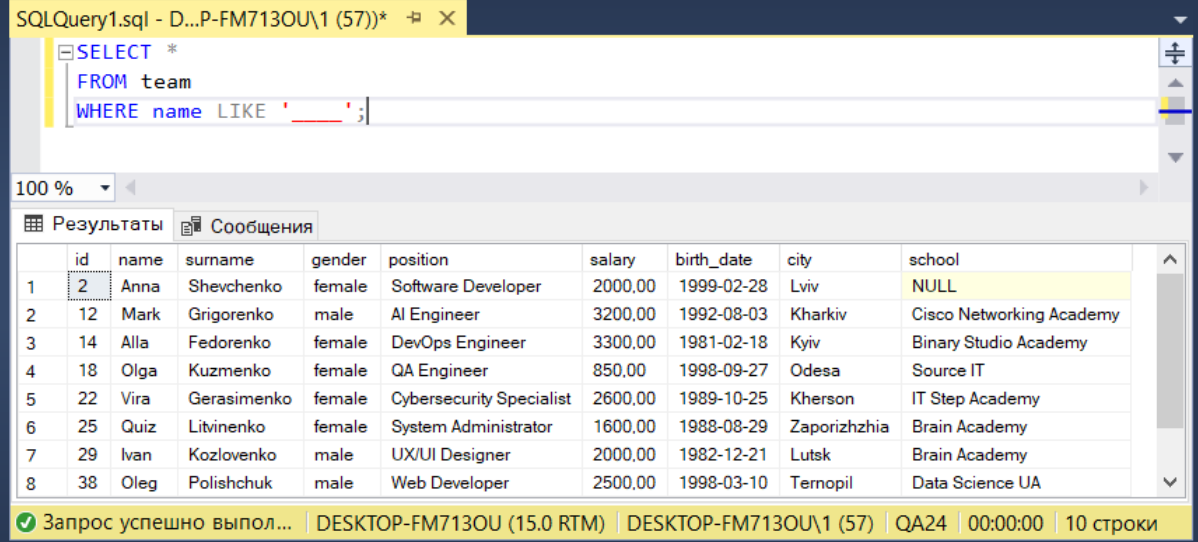
Результаты Сообщения

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|-----------|------------|--------|------------------------|---------|------------|---------|-----------------------|
| 1 | 2 | Anna | Shevchenko | female | Software Developer | 2000,00 | 1999-02-28 | Lviv | NULL |
| 2 | 5 | Artem | Kovalchuk | male | Web Developer | 2100,00 | 1988-12-29 | Odesa | NULL |
| 3 | 7 | Anastasia | Lysenko | female | Project Manager | 2400,00 | 1996-10-07 | Kyiv | GoIT |
| 4 | 9 | Alina | Miller | female | Database Administrator | 2200,00 | 1982-05-31 | Sumy | Beetroot Academy |
| 5 | 13 | Anton | Sidorenko | male | UX/UI Designer | 2700,00 | 1984-12-11 | Donetsk | EPAM Systems |
| 6 | 14 | Alla | Fedorenko | female | DevOps Engineer | 3300,00 | 1981-02-18 | Kyiv | Binary Studio Academy |
| 7 | 24 | Angelina | Gordienko | female | Web Developer | 2800,00 | 1991-05-19 | Kyiv | NULL |
| 8 | 28 | Andriy | Evsenko | male | Database Administrator | 1900,00 | 1986-09-09 | Sumy | SkillUp |

Запрос успешно выполнен... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 8 строки

Приклад. Вивести всю інформацію про співробітників, у яких ім'я має довжину 4 символи (_ _ _ _).

```
SELECT *  
FROM team  
WHERE name LIKE '____';
```



SQLQuery1.sql - D...P-FM713OU\1 (57))

```
SELECT *  
FROM team  
WHERE name LIKE '____';
```

100 %

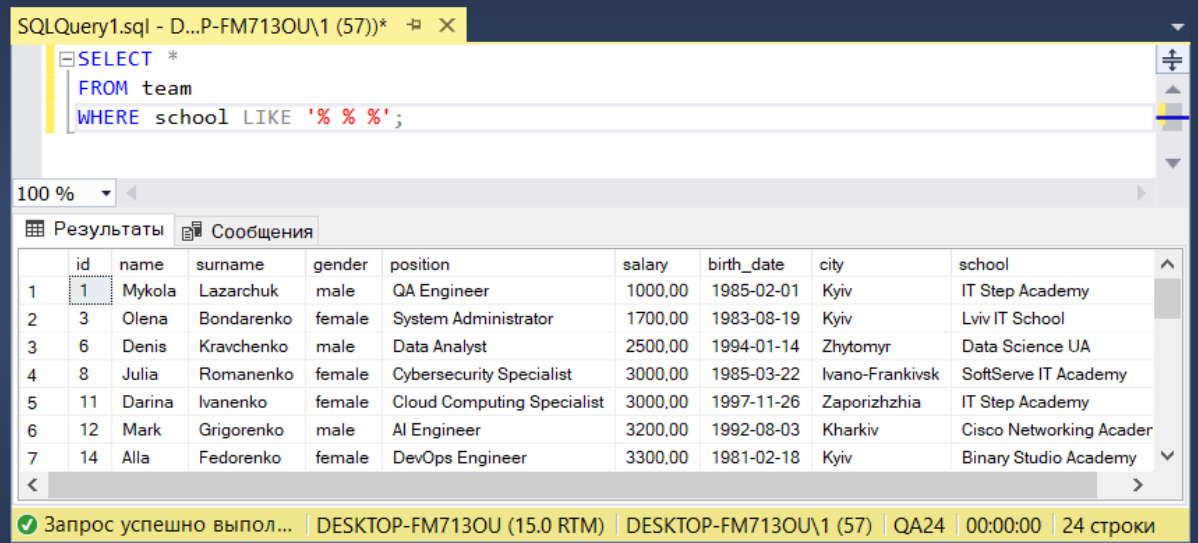
Результаты

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|------|-------------|--------|--------------------------|---------|------------|--------------|--------------------------|
| 1 | 2 | Anna | Shevchenko | female | Software Developer | 2000.00 | 1999-02-28 | Lviv | NULL |
| 2 | 12 | Mark | Grigorenko | male | AI Engineer | 3200.00 | 1992-08-03 | Kharkiv | Cisco Networking Academy |
| 3 | 14 | Alla | Fedorenko | female | DevOps Engineer | 3300.00 | 1981-02-18 | Kyiv | Binary Studio Academy |
| 4 | 18 | Olga | Kuzmenko | female | QA Engineer | 850.00 | 1998-09-27 | Odesa | Source IT |
| 5 | 22 | Vira | Gerasimenko | female | Cybersecurity Specialist | 2600.00 | 1989-10-25 | Kherson | IT Step Academy |
| 6 | 25 | Quiz | Litvinenko | female | System Administrator | 1600.00 | 1988-08-29 | Zaporizhzhia | Brain Academy |
| 7 | 29 | Ivan | Kozlovenko | male | UX/UI Designer | 2000.00 | 1982-12-21 | Lutsk | Brain Academy |
| 8 | 38 | Oleg | Polishchuk | male | Web Developer | 2500.00 | 1998-03-10 | Ternopil | Data Science UA |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 10 строки

Приклад. Вивести всю інформацію про співробітників, у яких назва ІТ школи має три слова (слова ми замінюємо на % і ставим два пробіли).

```
SELECT *  
FROM team  
WHERE school LIKE '% % %';
```



SQLQuery1.sql - D...P-FM713OU\1 (57))

```
SELECT *  
FROM team  
WHERE school LIKE '% % %';
```

100 %

Результаты

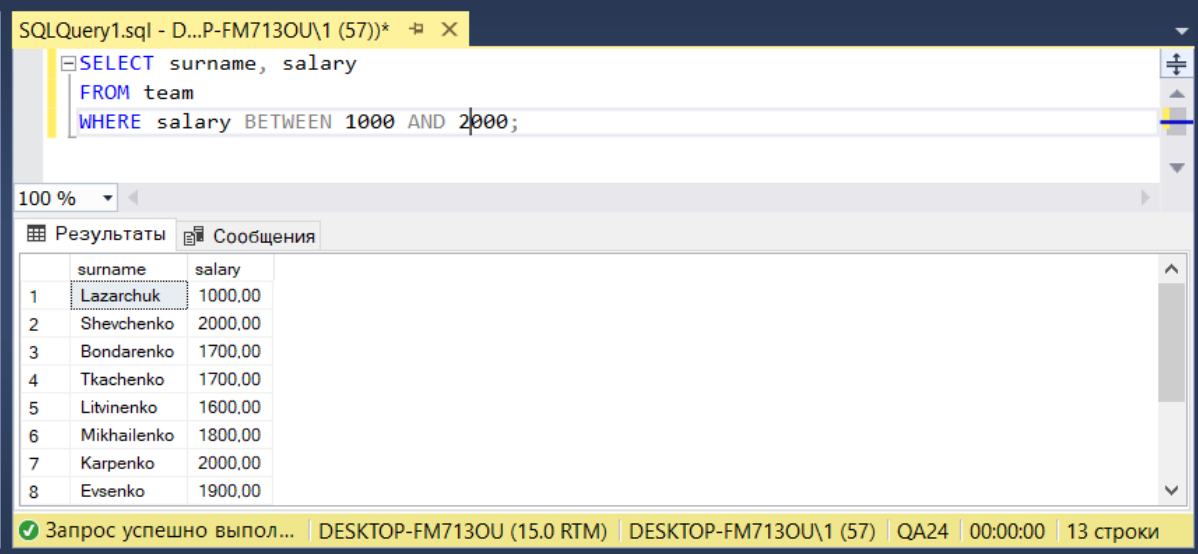
| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|--------|------------|--------|----------------------------|---------|------------|-----------------|-------------------------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000.00 | 1985-02-01 | Kyiv | IT Step Academy |
| 2 | 3 | Olena | Bondarenko | female | System Administrator | 1700.00 | 1983-08-19 | Kyiv | Lviv IT School |
| 3 | 6 | Denis | Kravchenko | male | Data Analyst | 2500.00 | 1994-01-14 | Zhytomyr | Data Science UA |
| 4 | 8 | Julia | Romanenko | female | Cybersecurity Specialist | 3000.00 | 1985-03-22 | Ivano-Frankivsk | SoftServe IT Academy |
| 5 | 11 | Darina | Ivanenko | female | Cloud Computing Specialist | 3000.00 | 1997-11-26 | Zaporizhzhia | IT Step Academy |
| 6 | 12 | Mark | Grigorenko | male | AI Engineer | 3200.00 | 1992-08-03 | Kharkiv | Cisco Networking Acader |
| 7 | 14 | Alla | Fedorenko | female | DevOps Engineer | 3300.00 | 1981-02-18 | Kyiv | Binary Studio Academy |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 24 строки

Щоб знайти записи в таблиці, значення яких перебувають у заданому діапазоні, можна скористатися оператором **BETWEEN**. Значення може бути будь-якого типу даних: числом, рядком і навіть датою.

Приклад. Потрібно вивести прізвища та зарплату співробітників, у яких зарплата між 1000 та 2000.

```
SELECT surname, salary
FROM team
WHERE salary BETWEEN 1000 AND 2000;
```



The screenshot shows a SQL query window with the following SQL code:

```
SELECT surname, salary
FROM team
WHERE salary BETWEEN 1000 AND 2000;
```

Below the query editor, the results are displayed in a table with two columns: 'surname' and 'salary'. The table contains 8 rows of data. The status bar at the bottom indicates 'Запрос успешно выпол...' (Query successfully executed) and '13 строки' (13 rows).

| | surname | salary |
|---|-------------|---------|
| 1 | Lazarchuk | 1000,00 |
| 2 | Shevchenko | 2000,00 |
| 3 | Bondarenko | 1700,00 |
| 4 | Tkachenko | 1700,00 |
| 5 | Litvinenko | 1600,00 |
| 6 | Mikhailenko | 1800,00 |
| 7 | Karpenko | 2000,00 |
| 8 | Evsenko | 1900,00 |

Приклад. Знайдемо імена всіх співробітників, які народилися між 01.01.1985 та 01.01.1990.

```
SELECT name, birth_date
FROM team
WHERE birth_date BETWEEN '1985-01-01' AND '1990-01-01';
```

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```

SELECT name, birth_date
FROM team
WHERE birth_date BETWEEN '1985-01-01' AND '1990-01-01';

```

100 %

Результаты Сообщения

| | name | birth_date |
|---|----------|------------|
| 1 | Mykola | 1985-02-01 |
| 2 | Artem | 1988-12-29 |
| 3 | Julia | 1985-03-22 |
| 4 | Katerina | 1986-09-10 |
| 5 | Larisa | 1987-04-01 |
| 6 | Vira | 1989-10-25 |
| 7 | Quiz | 1988-08-29 |
| 8 | Andriy | 1986-09-09 |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 13 строки

Запит вище можна переписати за допомогою операторів \geq та \leq . Але він став складнішим.

```

SELECT name, birth_date
FROM team
WHERE birth_date >= '1985-01-01' AND birth_date <='1990-01-01';

```

SQLQuery1.sql - D...P-FM713OU\1 (57))*

```

SELECT name, birth_date
FROM team
WHERE birth_date >= '1985-01-01' AND birth_date <='1990-01-01';

```

100 %

Результаты Сообщения

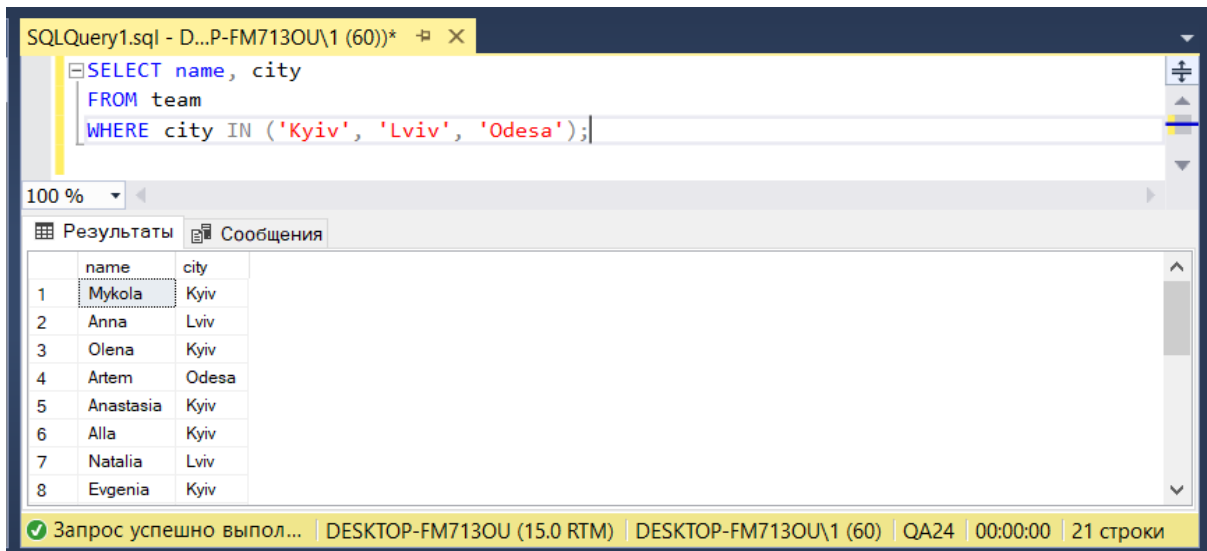
| | name | birth_date |
|---|----------|------------|
| 1 | Mykola | 1985-02-01 |
| 2 | Artem | 1988-12-29 |
| 3 | Julia | 1985-03-22 |
| 4 | Katerina | 1986-09-10 |
| 5 | Larisa | 1987-04-01 |
| 6 | Vira | 1989-10-25 |
| 7 | Quiz | 1988-08-29 |
| 8 | Andriy | 1986-09-09 |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (57) | QA24 | 00:00:00 | 13 строки

За допомогою оператора **IN** можна вказати кілька значень всередині **WHERE**.

Приклад. Знайдемо імена всіх користувачів із міст Lviv, Odesa, Kyiv.

```
SELECT name, city
FROM team
WHERE city IN ('Kyiv', 'Lviv', 'Odesa');
```



SQLQuery1.sql - D...P-FM713OU\1 (60))*

```
SELECT name, city
FROM team
WHERE city IN ('Kyiv', 'Lviv', 'Odesa');
```

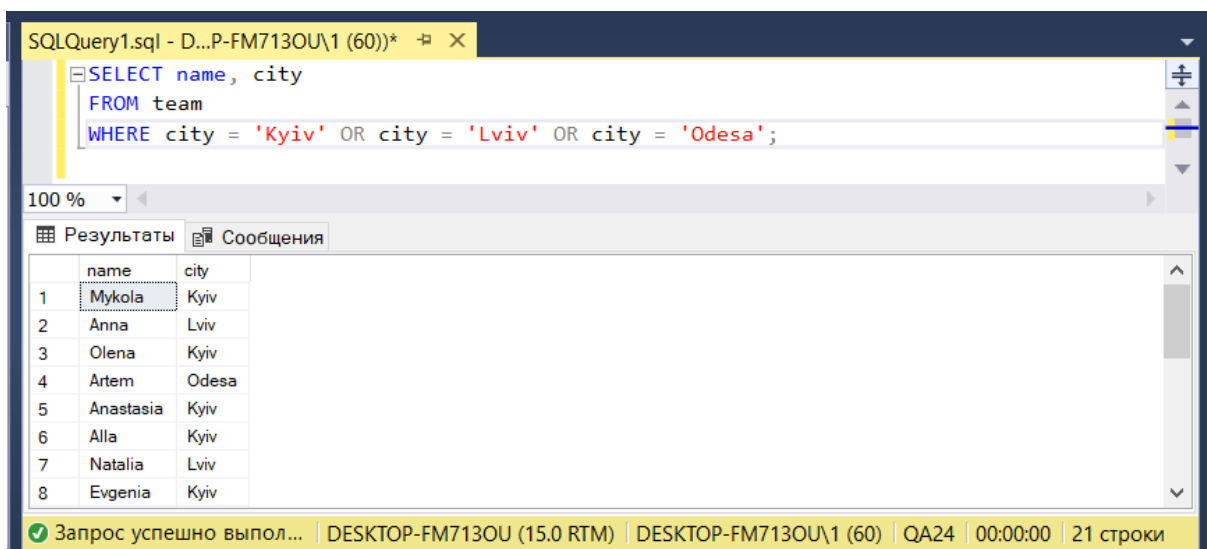
100 %

Результаты Сообщения

| | name | city |
|---|-----------|-------|
| 1 | Mykola | Kyiv |
| 2 | Anna | Lviv |
| 3 | Olena | Kyiv |
| 4 | Artem | Odesa |
| 5 | Anastasia | Kyiv |
| 6 | Alla | Kyiv |
| 7 | Natalia | Lviv |
| 8 | Evgenia | Kyiv |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 21 строки

Оператор **IN** є скороченням для кількох умов **OR**. Запит вище можна записати так:



SQLQuery1.sql - D...P-FM713OU\1 (60))*

```
SELECT name, city
FROM team
WHERE city = 'Kyiv' OR city = 'Lviv' OR city = 'Odesa';
```

100 %

Результаты Сообщения

| | name | city |
|---|-----------|-------|
| 1 | Mykola | Kyiv |
| 2 | Anna | Lviv |
| 3 | Olena | Kyiv |
| 4 | Artem | Odesa |
| 5 | Anastasia | Kyiv |
| 6 | Alla | Kyiv |
| 7 | Natalia | Lviv |
| 8 | Evgenia | Kyiv |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 21 строки

Щоб отримати записи таблиці, які не відповідають умові, можна використовувати оператор **NOT** разом з **LIKE**, **BETWEEN** та **IN**.

Приклад. Напишемо запит, який поверне записи, де name не містить літеру а.


```
SELECT *
FROM team
WHERE name NOT LIKE '%a%' ;
```

SQLQuery1.sql - D...P-FM713OU\1 (60))*

```
SELECT *
FROM team
WHERE name NOT LIKE '%a%' ;
```

100 %

Результаты Сообщения

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|--------|------------|--------|----------------------------|---------|------------|--------------|--------------------------|
| 1 | 4 | Petro | Tkachenko | male | Network Engineer | 1700.00 | 1990-06-02 | Dnipro | UNIT Factory |
| 2 | 6 | Denis | Kravchenko | male | Data Analyst | 2500.00 | 1994-01-14 | Zhytomyr | Data Science UA |
| 3 | 25 | Quiz | Litvinenko | female | System Administrator | 1600.00 | 1988-08-29 | Zaporizhzhia | Brain Academy |
| 4 | 31 | Dmitro | Bondarenko | male | IT Consultant | 2500.00 | 1999-06-30 | Kharkiv | Cisco Networking Academy |
| 5 | 32 | Sergiy | Kuznetsov | male | UX/UI Designer | 2000.00 | 1984-08-05 | Vinnytsia | LaSoft School |
| 6 | 33 | Victor | Levchenko | male | Cloud Computing Specialist | 2400.00 | 1992-10-17 | Kyiv | Lviv IT School |
| 7 | 37 | Novel | Zakharenko | male | Data Scientist | 2500.00 | 1980-06-23 | Poltava | Brain Academy |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 12 строки

Приклад. Напишемо запит який поверне записи, у яких id не потрапляє в діапазон від 4 до 10.

```
SELECT *
FROM team
WHERE id NOT BETWEEN 4 AND 10;
```

SQLQuery1.sql - D...P-FM713OU\1 (60))*

```
SELECT *
FROM team
WHERE id NOT BETWEEN 4 AND 10;
```

100 %

Результаты Сообщения

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|--------|------------|--------|----------------------------|---------|------------|--------------|--------------------------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000.00 | 1985-02-01 | Kyiv | IT Step Academy |
| 2 | 2 | Anna | Shevchenko | female | Software Developer | 2000.00 | 1999-02-28 | Lviv | NULL |
| 3 | 3 | Olena | Bondarenko | female | System Administrator | 1700.00 | 1983-08-19 | Kyiv | Lviv IT School |
| 4 | 11 | Darina | Ivanenko | female | Cloud Computing Specialist | 3000.00 | 1997-11-26 | Zaporizhzhia | IT Step Academy |
| 5 | 12 | Mark | Grigorenko | male | AI Engineer | 3200.00 | 1992-08-03 | Kharkiv | Cisco Networking Academy |
| 6 | 13 | Anton | Sidorenko | male | UX/UI Designer | 2700.00 | 1984-12-11 | Donetsk | EPAM Systems |
| 7 | 14 | Alla | Fedorenko | female | DevOps Engineer | 3300.00 | 1981-02-18 | Kyiv | Binary Studio Academy |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 43 строки

Приклад. Напишемо запит який поверне записи, у яких місто не дорівнює Odesa або Kyiv.

```
SELECT *
FROM team
WHERE city NOT IN ('Odesa', 'Kyiv');
```

The screenshot shows a SQL query window with the following query:

```
SELECT *
FROM team
WHERE city NOT IN ('Odesa', 'Kyiv');
```

The results are displayed in a table with the following columns: id, name, surname, gender, position, salary, birth_date, city, and school. The results are as follows:

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|----------|------------|--------|----------------------------|---------|------------|-----------------|----------------------|
| 1 | 2 | Anna | Shevchenko | female | Software Developer | 2000.00 | 1999-02-28 | Lviv | NULL |
| 2 | 4 | Petro | Tkachenko | male | Network Engineer | 1700.00 | 1990-06-02 | Dnipro | UNIT Factory |
| 3 | 6 | Denis | Kravchenko | male | Data Analyst | 2500.00 | 1994-01-14 | Zhytomyr | Data Science UA |
| 4 | 8 | Julia | Romanenko | female | Cybersecurity Specialist | 3000.00 | 1985-03-22 | Ivano-Frankivsk | SoftServe IT Academy |
| 5 | 9 | Alina | Miller | female | Database Administrator | 2200.00 | 1982-05-31 | Sumy | Beetroot Academy |
| 6 | 10 | Katerina | Petrenko | female | IT Consultant | 2800.00 | 1986-09-10 | Vinnytsia | SkillUp |
| 7 | 11 | Darina | Ivanenko | female | Cloud Computing Specialist | 3000.00 | 1997-11-26 | Zaporizhzhia | IT Step Academy |

The status bar at the bottom indicates: Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 32 строки

За допомогою оператора **ORDER BY** можна сортувати записи за одним або кількома стовпчиками. За замовчуванням записи сортуються в порядку зростання (за алфавітом). Але також можна явно вказати порядок сортування — для цього використай ключове слово **DESC** для порядку спадання, а слово **ASC** — для порядку зростання. Ми можемо використовувати оператор **ORDER BY** з рядками, числами та датою

Приклад. Напишемо запит, який повертає рядки, відсортовані за стовпчиком name у порядку зростання (в алфавітному порядку).

```
SELECT *
FROM team
ORDER BY name;
```

або

```
SELECT *
FROM team
ORDER BY name ASC;
```

SQLQuery1.sql - D...P-FM713OU\1 (60))*

```
SELECT *
FROM team
ORDER BY name;
```

100 %

Результаты

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|-----------|------------|--------|------------------------|---------|------------|---------|-----------------------|
| 1 | 9 | Alina | Miller | female | Database Administrator | 2200,00 | 1982-05-31 | Sumy | Beetroot Academy |
| 2 | 14 | Alla | Fedorenko | female | DevOps Engineer | 3300,00 | 1981-02-18 | Kyiv | Binary Studio Academy |
| 3 | 7 | Anastasia | Lysenko | female | Project Manager | 2400,00 | 1996-10-07 | Kyiv | GolT |
| 4 | 28 | Andriy | Evsenko | male | Database Administrator | 1900,00 | 1986-09-09 | Sumy | SkillUp |
| 5 | 24 | Angelina | Gordienko | female | Web Developer | 2800,00 | 1991-05-19 | Kyiv | NULL |
| 6 | 2 | Anna | Shevchenko | female | Software Developer | 2000,00 | 1999-02-28 | Lviv | NULL |
| 7 | 13 | Anton | Sidorenko | male | UX/UI Designer | 2700,00 | 1984-12-11 | Donetsk | EPAM Systems |
| 8 | 5 | Artem | Kovalchuk | male | Web Developer | 2100,00 | 1988-12-29 | Odesa | NULL |

Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 50 строки

Приклад. Напишемо запит, який повертає рядки, відсортовані за стовпчиком name у порядку спадання.

```
SELECT *
FROM team
ORDER BY name DESC;
```

SQLQuery1.sql - D...P-FM713OU\1 (60))*

```
SELECT *
FROM team
ORDER BY name DESC;
```

100 %

Результаты

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|-----------|-------------|--------|----------------------------|---------|------------|-----------------|---------------------|
| 1 | 44 | Yuri | White | male | Database Administrator | 2200,00 | 1993-11-08 | Zhytomyr | GolT |
| 2 | 35 | Yaroslav | Kozak | male | Junior Data Analyst | 1500,00 | 1983-11-27 | Odesa | NULL |
| 3 | 45 | Volodymyr | Yakovenko | male | IT Consultant | 2800,00 | 1982-04-03 | Kyiv | SkillUp |
| 4 | 34 | Vladislav | Kushnirenko | male | IT Consultant | 2500,00 | 1981-01-01 | Ivano-Frankivsk | IT Step Academy |
| 5 | 46 | Vitaliy | Dmitrenko | male | Cloud Computing Specialist | 3000,00 | 1994-07-22 | Ivano-Frankivsk | Cisco Networking Ac |
| 6 | 22 | Vira | Gerasimenko | female | Cybersecurity Specialist | 2600,00 | 1989-10-25 | Kherson | IT Step Academy |
| 7 | 43 | Victoria | Ponomarenko | female | Cybersecurity Specialist | 2900,00 | 1995-01-18 | Odesa | Lviv IT School |
| 8 | 33 | Victor | Levchenko | male | Cloud Computing Specialist | 2400,00 | 1992-10-17 | Kyiv | Lviv IT School |

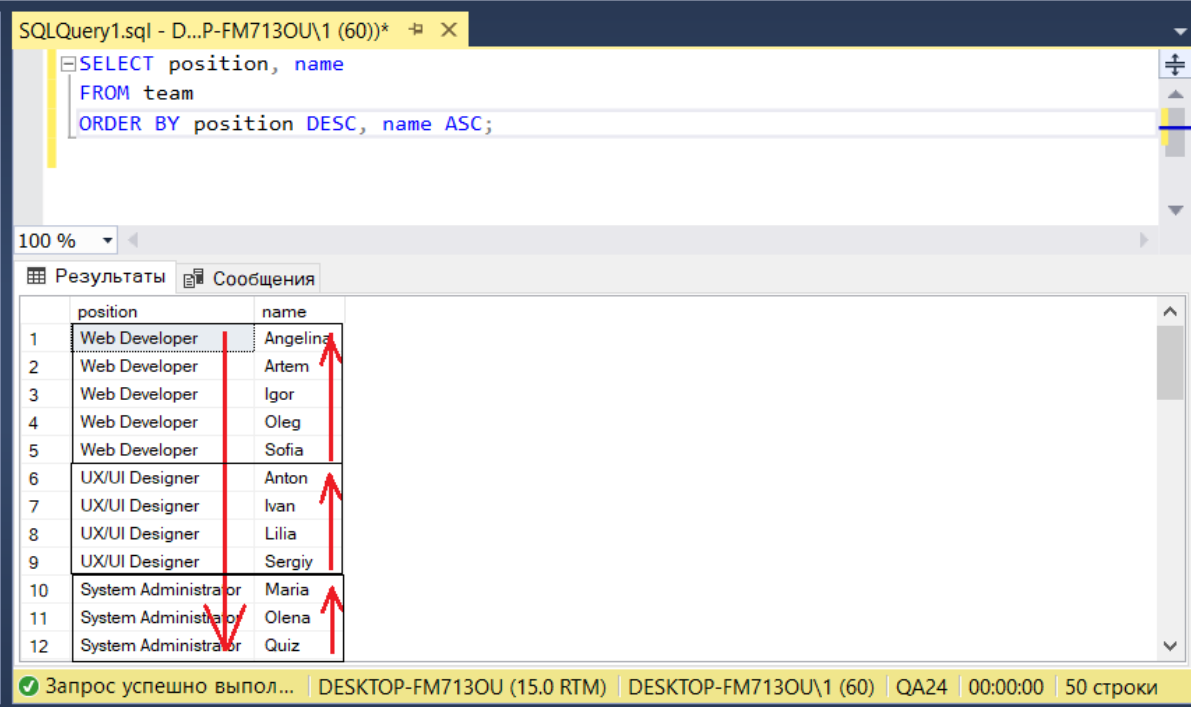
Запрос успешно выпол... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 50 строки

Наступний запит повертає записи з таблиці, відсортовані за двома стовпчиками: position порядку спадання та name у порядку зростання.

```
SELECT position, name
FROM team
ORDER BY position DESC, name ASC;
```

Посади відсортовані в порядку спадання, а імена всередині конкретної посади — в порядку зростання.

Зверни увагу: сортування записів відбувається у тому порядку, який ми вказуємо. У прикладі вище (**ORDER BY** position **DESC**, name **ASC**) записи відсортовані спочатку за посадами, а потім за іменами.



The screenshot shows a SQL Query Editor window with the following query:

```
SELECT position, name
FROM team
ORDER BY position DESC, name ASC;
```

The results are displayed in a table with two columns: 'position' and 'name'. The results are sorted by position in descending order, and then by name in ascending order. Red arrows indicate the sorting order: a vertical arrow pointing down for the 'position' column and a vertical arrow pointing up for the 'name' column.

| | position | name |
|----|----------------------|----------|
| 1 | Web Developer | Angelina |
| 2 | Web Developer | Artem |
| 3 | Web Developer | Igor |
| 4 | Web Developer | Oleg |
| 5 | Web Developer | Sofia |
| 6 | UX/UI Designer | Anton |
| 7 | UX/UI Designer | Ivan |
| 8 | UX/UI Designer | Lilia |
| 9 | UX/UI Designer | Sergiy |
| 10 | System Administrator | Maria |
| 11 | System Administrator | Olena |
| 12 | System Administrator | Quiz |

The status bar at the bottom indicates: Запрос успешно выполнен... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 50 строки

В SQL-запросах, особливо в Microsoft SQL Server, оператор **TOP** використовується для обмеження кількості повернутих рядків в результаті запиту **SELECT**. Оператор **TOP** визначає кількість рядків, які потрібно повернути з верхньої частини результату.

TOP використовується в **Microsoft SQL Server**, тоді як **LIMIT** - в інших популярних СУБД, таких як **MySQL**, **PostgreSQL** і **SQLite**. Однак обидва оператори служать для досягнення подібної функціональності - обмеження кількості повернутих рядків в результаті запиту **SELECT**.

Приклад. Ви хочете отримати перші 5 рядків з таблиці "team", використовуйте такий запит.

```
SELECT TOP 5 *
FROM team;
```

SQLQuery1.sql - D...P-FM713OU\1 (60))* ✕

```
SELECT TOP 5 *
FROM team;
```

100 %

Результаты Сообщения

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|--------|------------|--------|----------------------|---------|------------|--------|-----------------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000,00 | 1985-02-01 | Kyiv | IT Step Academy |
| 2 | 2 | Anna | Shevchenko | female | Software Developer | 2000,00 | 1999-02-28 | Lviv | NULL |
| 3 | 3 | Olena | Bondarenko | female | System Administrator | 1700,00 | 1983-08-19 | Kyiv | Lviv IT School |
| 4 | 4 | Petro | Tkachenko | male | Network Engineer | 1700,00 | 1990-06-02 | Dnipro | UNIT Factory |
| 5 | 5 | Artem | Kovalchuk | male | Web Developer | 2100,00 | 1988-12-29 | Odesa | NULL |

Запрос... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 5 строки

Приклад. Зробимо запит, де результат буде містити 5 рядків з найвищими значеннями зарплати, разом з відповідними прізвищами.

```
SELECT TOP 5 salary, surname
FROM team
ORDER BY salary DESC;
```

SQLQuery1.sql - D...P-FM713OU\1 (60))* ✕

```
SELECT TOP 5 salary, surname
FROM team
ORDER BY salary DESC;
```

100 %

Результаты Сообщения

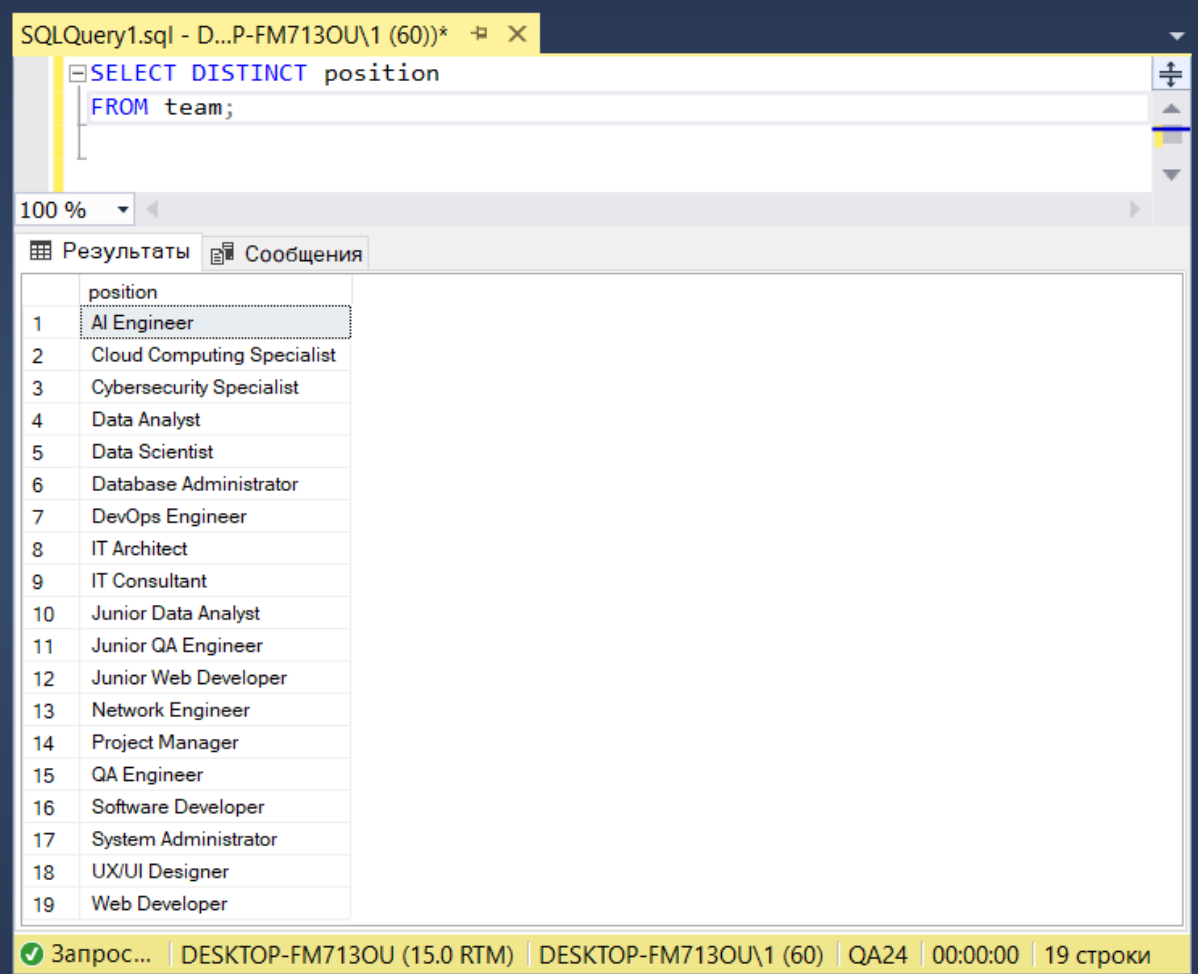
| | salary | surname |
|---|---------|-------------|
| 1 | 3500,00 | Popovich |
| 2 | 3400,00 | Pavlenko |
| 3 | 3300,00 | Fedorenko |
| 4 | 3200,00 | Grigorenko |
| 5 | 3200,00 | Goncharenko |

Запрос... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 5 строки

Оператор **DISTINCT** в SQL використовується для видалення повторюючихся значень з результату запиту SELECT. Він дозволяє отримати унікальні значення з вказаних стовпців або комбінації стовпців. Коли використовується оператор DISTINCT, база даних видаляє будь-які дублікати з результату і повертає лише унікальні значення. Це корисно, коли ви хочете отримати унікальний список значень з певного стовпця або здійснити аналіз даних, видаляючи повтори.

Приклад. Потрібно вивести список посад в компанії.

```
SELECT DISTINCT position  
FROM team;
```



SQLQuery1.sql - D...P-FM713OU\1 (60))*

```
SELECT DISTINCT position  
FROM team;
```

100 %

Результаты Сообщения

| | position |
|----|----------------------------|
| 1 | AI Engineer |
| 2 | Cloud Computing Specialist |
| 3 | Cybersecurity Specialist |
| 4 | Data Analyst |
| 5 | Data Scientist |
| 6 | Database Administrator |
| 7 | DevOps Engineer |
| 8 | IT Architect |
| 9 | IT Consultant |
| 10 | Junior Data Analyst |
| 11 | Junior QA Engineer |
| 12 | Junior Web Developer |
| 13 | Network Engineer |
| 14 | Project Manager |
| 15 | QA Engineer |
| 16 | Software Developer |
| 17 | System Administrator |
| 18 | UX/UI Designer |
| 19 | Web Developer |

Запрос... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (60) | QA24 | 00:00:00 | 19 строки

Агрегатна функція виконує обчислення одного чи кількох значень із стовпців та повертає єдиний результат.

Нижче наведено агрегатні функції, які найчастіше використовуються в SQL:

COUNT — повертає кількість записів таблиці чи стовпця;

MIN — повертає найменше значення вказаного стовпця;

MAX — повертає найбільше значення вказаного стовпця;

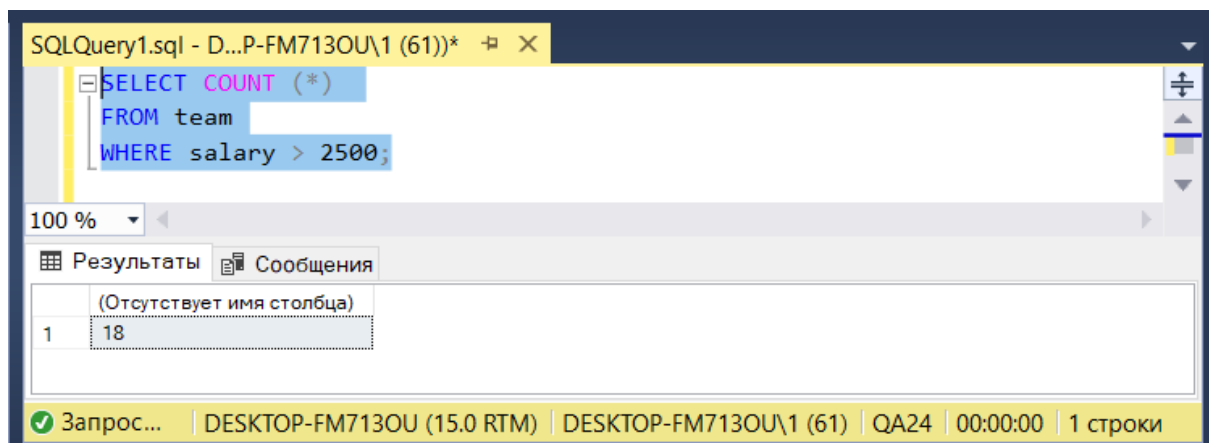
SUM — повертає суму значень у вказаному стовпці;

AVG — повертає середнє значення вказаного стовпця.

Функція **COUNT** повертає кількість записів із таблиці або стовпця, які відповідають умові. Ми можемо використовувати цю функцію з будь-яким типом даних.

Приклад. Напишемо запит, який поверне кількість працівників, у яких зарплата більша 2500 \$.

```
SELECT COUNT (*)  
FROM team  
WHERE salary > 2500;
```



Також можна встановити псевдонім для стовпця результатів

```
SELECT COUNT (*) AS 'К-ть працівників з ЗП > 2500'  
FROM team  
WHERE salary > 2500;
```

SQLQuery1.sql - D...P-FM713OU\1 (61))* ✕

```

SELECT COUNT (*) AS 'К-ть працівників з ЗП > 2500'
FROM team
WHERE salary > 2500;

```

100 %

Результаты Сообщения

| | К-ть працівників з ЗП > 2500 |
|---|------------------------------|
| 1 | 18 |

✓ Запрос... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (61) | QA24 | 00:00:00 | 1 строки

COUNT(*) повертає кількість записів із таблиці, які відповідають умові. При цьому деякі стовпці можуть мати значення NULL;
COUNT(column_name) повертає кількість тільки тих записів, де значення стовпця не дорівнює NULL

Приклад. Наступні два запити показують різницю кількості записів, де стовпці не мають значення NULL та мають NULL (в стовпці school є 6 значень NULL).

SQLQuery1.sql - D...P-FM713OU\1 (61))* ✕

```

SELECT COUNT (name)
FROM team;
SELECT COUNT (school)
FROM team;

```

100 %

Результаты Сообщения

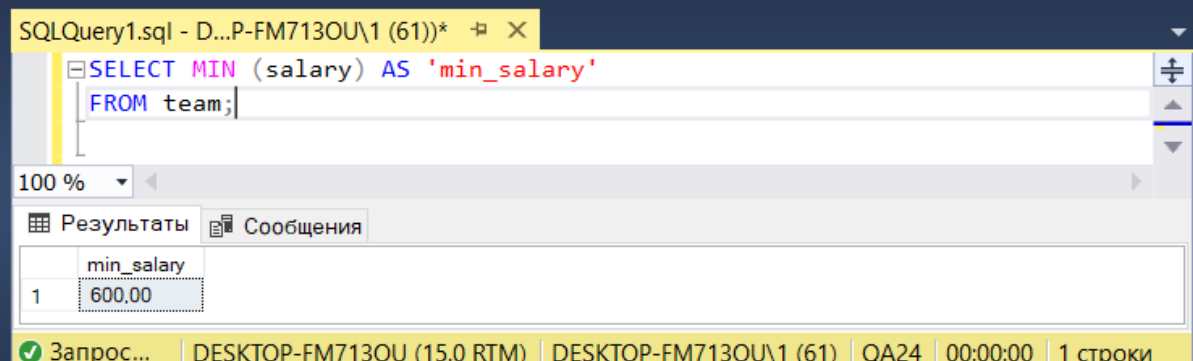
| | | |
|---|---------------------------|----|
| 1 | (Отсутствует имя столбца) | 50 |
| 1 | (Отсутствует имя столбца) | 44 |

✓ Запрос... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (61) | QA24 | 00:00:00 | 2 строки

Функція **MIN** повертає найменше значення вказаного стовпця. Функція **MAX** повертає найбільше значення вказаного стовпця.

Приклад. Визначимо мінімальну зарплату працівників компанії.

```
SELECT MIN (salary) AS 'min_salary'  
FROM team;
```

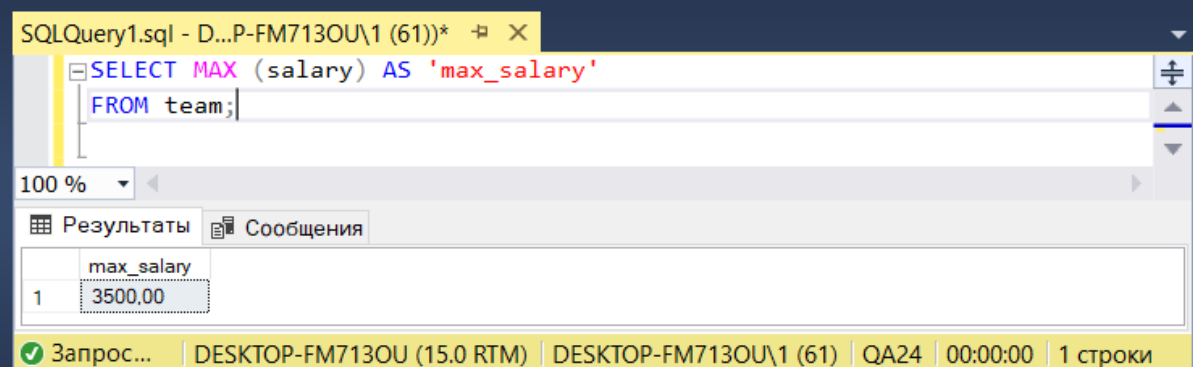


The screenshot shows a SQL query editor window titled "SQLQuery1.sql - D...P-FM713OU\1 (61))*". The query entered is `SELECT MIN (salary) AS 'min_salary' FROM team;`. Below the query editor, the "Результаты" (Results) tab is active, displaying a single row with the column header "min_salary" and the value "600,00". The status bar at the bottom indicates "Запрос..." (Query...), "DESKTOP-FM713OU (15.0 RTM)", "DESKTOP-FM713OU\1 (61)", "QA24", "00:00:00", and "1 строки" (1 rows).

| | min_salary |
|---|------------|
| 1 | 600,00 |

Приклад. Визначимо максимальну зарплату працівників компанії.

```
SELECT MAX (salary) AS 'max_salary'  
FROM team;
```



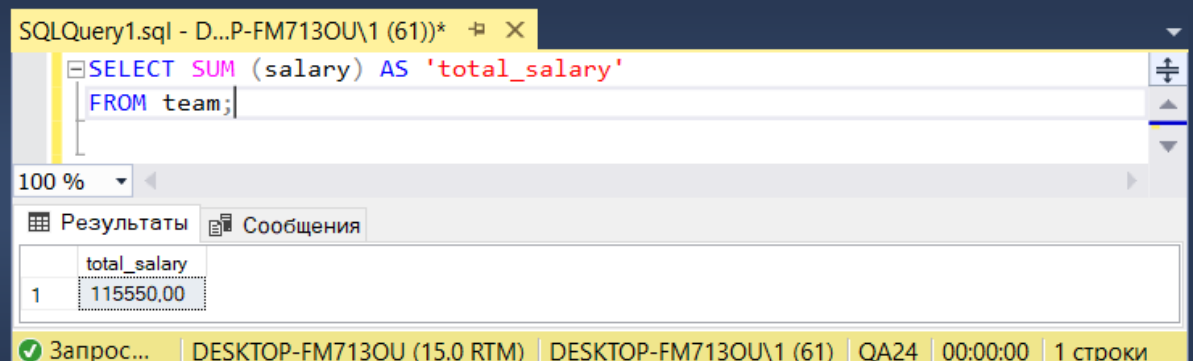
The screenshot shows a SQL query editor window titled "SQLQuery1.sql - D...P-FM713OU\1 (61))*". The query entered is `SELECT MAX (salary) AS 'max_salary' FROM team;`. Below the query editor, the "Результаты" (Results) tab is active, displaying a single row with the column header "max_salary" and the value "3500,00". The status bar at the bottom indicates "Запрос..." (Query...), "DESKTOP-FM713OU (15.0 RTM)", "DESKTOP-FM713OU\1 (61)", "QA24", "00:00:00", and "1 строки" (1 rows).

| | max_salary |
|---|------------|
| 1 | 3500,00 |

Функція **SUM** повертає загальну суму значень числового стовпця. Цю функцію можна використовувати лише з числовими типами даних.

Приклад. Робимо запит, який поверне суму зарплат усіх працівників.

```
SELECT SUM (salary) AS 'total_salary'  
FROM team;
```



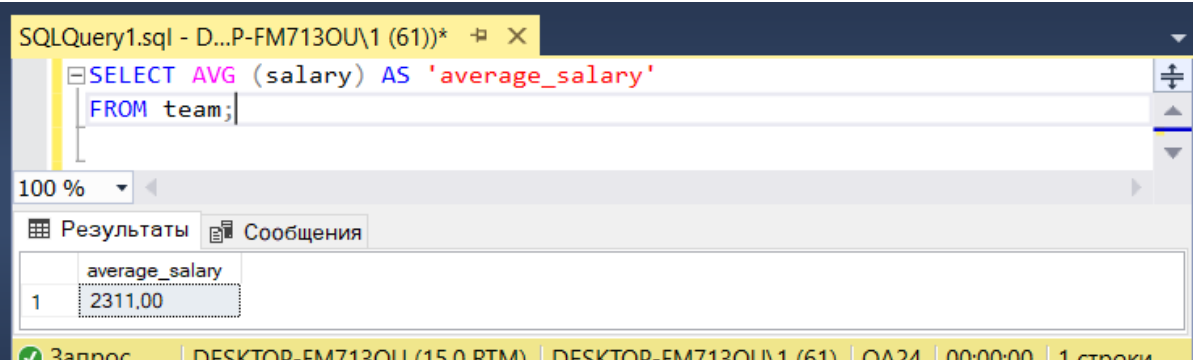
The screenshot shows a SQL query window titled 'SQLQuery1.sql - D...P-FM713OU\1 (61))*'. The query is: `SELECT SUM (salary) AS 'total_salary' FROM team;`. Below the query editor, the 'Результаты' (Results) tab is active, displaying a table with one row and one column. The status bar at the bottom indicates 'Запрос...' (Query...), 'DESKTOP-FM713OU (15.0 RTM)', 'DESKTOP-FM713OU\1 (61)', 'QA24', '00:00:00', and '1 строки' (1 rows).

| | total_salary |
|---|--------------|
| 1 | 115550.00 |

Функція **AVG** повертає середнє значення числового стовпця. Цю функцію можна використовувати лише з числовими типами даних.

Приклад. З'ясуємо середню зарплату працівників компанії.

```
SELECT AVG (salary) AS 'average_salary'  
FROM team;
```



The screenshot shows a SQL query window titled 'SQLQuery1.sql - D...P-FM713OU\1 (61))*'. The query is: `SELECT AVG (salary) AS 'average_salary' FROM team;`. Below the query editor, the 'Результаты' (Results) tab is active, displaying a table with one row and one column. The status bar at the bottom indicates 'Запрос...' (Query...), 'DESKTOP-FM713OU (15.0 RTM)', 'DESKTOP-FM713OU\1 (61)', 'QA24', '00:00:00', and '1 строки' (1 rows).

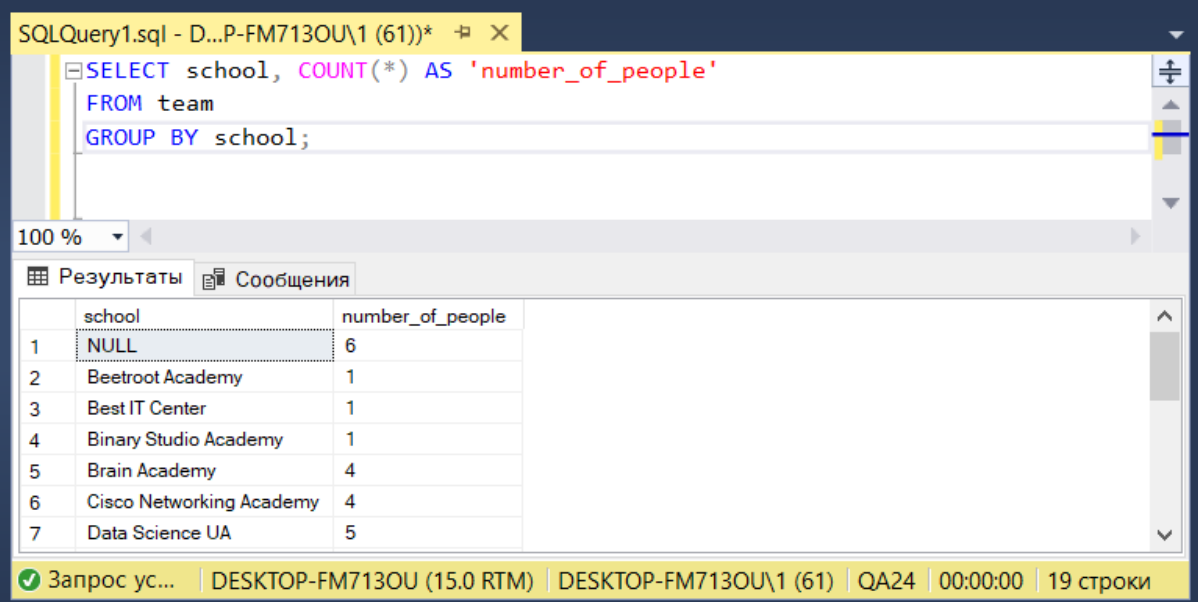
| | average_salary |
|---|----------------|
| 1 | 2311.00 |

Оператор **GROUP BY** групує записи вибірки за групами та повертає один запис для кожної групи. З його допомогою можна групувати записи по одному або кількох стовпцях.

Оператор GROUP BY часто використовується з агрегатними функціями: COUNT, MAX, MIN, SUM та AVG.

Приклад. Потрібно зробити запит, який поверне назви навчальних закладів, та кількість працівників, яка в них вчилася.

```
SELECT school, COUNT(*) AS 'number_of_people'
FROM team
GROUP BY school;
```



The screenshot shows a SQL query window with the following text:

```
SQLQuery1.sql - D...P-FM713OU\1 (61))* X
SELECT school, COUNT(*) AS 'number_of_people'
FROM team
GROUP BY school;
```

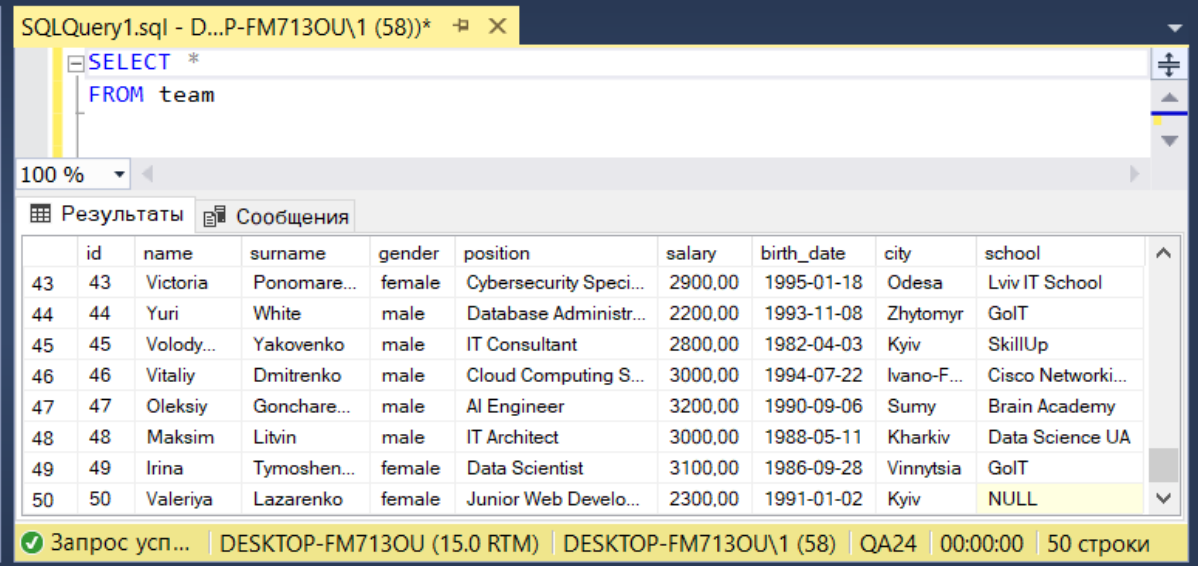
Below the query window, the 'Results' tab is active, displaying a table with two columns: 'school' and 'number_of_people'. The table contains 7 rows of data.

| | school | number_of_people |
|---|--------------------------|------------------|
| 1 | NULL | 6 |
| 2 | Beetroot Academy | 1 |
| 3 | Best IT Center | 1 |
| 4 | Binary Studio Academy | 1 |
| 5 | Brain Academy | 4 |
| 6 | Cisco Networking Academy | 4 |
| 7 | Data Science UA | 5 |

At the bottom of the window, a status bar shows: Запрос ус... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (61) | QA24 | 00:00:00 | 19 строки

Оператор **INSERT** використовується для додавання нових рядків до таблиці в базі даних. Він дозволяє вставити значення в один або кілька стовпців таблиці.

Нам потрібно додати дані нового співробітника. Ми маємо 50 співробітників.



SQLQuery1.sql - D...P-FM713OU\1 (58))*

```
SELECT *  
FROM team
```

100 %

Результаты Сообщения

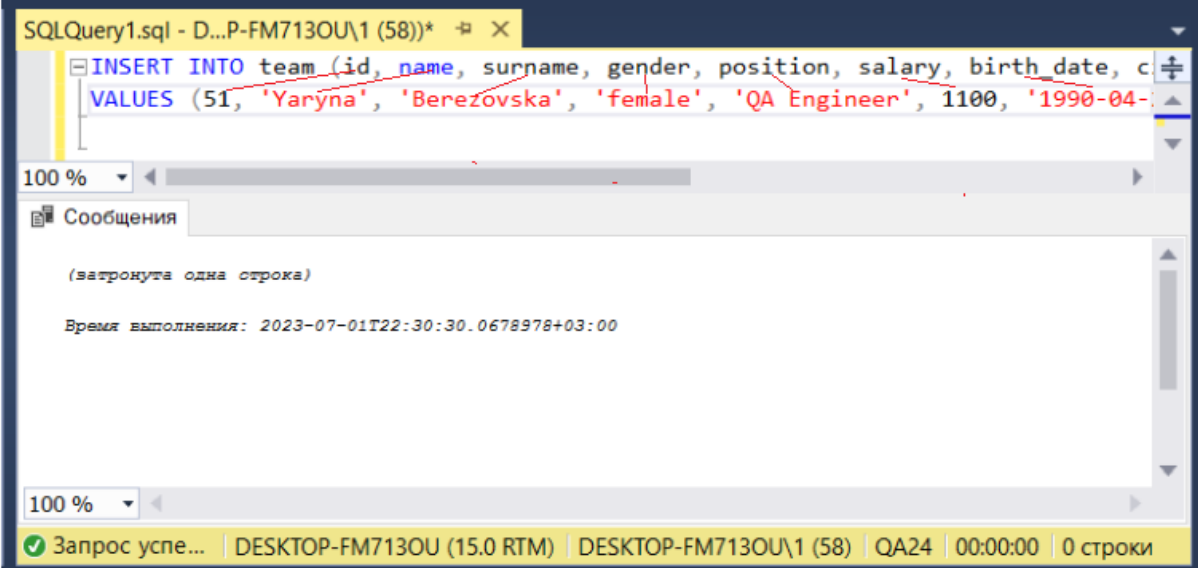
| | id | name | surname | gender | position | salary | birth_date | city | school |
|----|----|-----------|-------------|--------|------------------------|---------|------------|------------|-------------------|
| 43 | 43 | Victoria | Ponomare... | female | Cybersecurity Speci... | 2900,00 | 1995-01-18 | Odesa | Lviv IT School |
| 44 | 44 | Yuri | White | male | Database Administr... | 2200,00 | 1993-11-08 | Zhytomyr | GolT |
| 45 | 45 | Volody... | Yakovenko | male | IT Consultant | 2800,00 | 1982-04-03 | Kyiv | SkillUp |
| 46 | 46 | Vitaliy | Dmitrenko | male | Cloud Computing S... | 3000,00 | 1994-07-22 | Ivano-F... | Cisco Networki... |
| 47 | 47 | Oleksiy | Gonchare... | male | AI Engineer | 3200,00 | 1990-09-06 | Sumy | Brain Academy |
| 48 | 48 | Maksim | Litvin | male | IT Architect | 3000,00 | 1988-05-11 | Kharkiv | Data Science UA |
| 49 | 49 | Irina | Tymoshen... | female | Data Scientist | 3100,00 | 1986-09-28 | Vinnytsia | GolT |
| 50 | 50 | Valeriya | Lazarenko | female | Junior Web Develo... | 2300,00 | 1991-01-02 | Kyiv | NULL |

Запрос успе... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (58) | QA24 | 00:00:00 | 50 строки

Приклад. Наступний запит додати нового співробітника. Ми вказуємо назву стовпця та дані які хочем ввести (id - 51, name - Yaryna і тд).

INSERT INTO team (id, name, surname, gender, position, salary, birth_date, city, school)

VALUES (51, 'Yaryna', 'Berezovska', 'female', 'QA Engineer', 1100, '1990-04-25', 'Dnipro', 'Data Science UA');



SQLQuery1.sql - D...P-FM713OU\1 (58))*

```
INSERT INTO team (id, name, surname, gender, position, salary, birth_date, city, school)  
VALUES (51, 'Yaryna', 'Berezovska', 'female', 'QA Engineer', 1100, '1990-04-25', 'Dnipro', 'Data Science UA');
```

100 %

Сообщения

(затронута одна строка)

Время выполнения: 2023-07-01T22:30:30.0678978+03:00

100 %

Запрос успе... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (58) | QA24 | 00:00:00 | 0 строки

В результаті добавили ще рядок з даними співробітника.

| | id | name | surname | gender | position | salary | birth_date | city | school |
|----|----|----------|-------------|--------|----------------------|---------|------------|------------|--------------------|
| 46 | 46 | Vitaliy | Dmitrenko | male | Cloud Computing S... | 3000.00 | 1994-07-22 | Ivano-F... | Cisco Networking A |
| 47 | 47 | Oleksiy | Gonchare... | male | AI Engineer | 3200.00 | 1990-09-06 | Sumy | Brain Academy |
| 48 | 48 | Maksim | Litvin | male | IT Architect | 3000.00 | 1988-05-11 | Kharkiv | Data Science UA |
| 49 | 49 | Irina | Tymoshen... | female | Data Scientist | 3100.00 | 1986-09-28 | Vinnytsia | GolT |
| 50 | 50 | Valeriya | Lazarenko | female | Junior Web Develo... | 2300.00 | 1991-01-02 | Kyiv | NULL |
| 51 | 51 | Yaryna | Berezovska | female | QA Engineer | 1100.00 | 1990-04-25 | Dnipro | Data Science UA |

Є ще один спосіб, назви стовпців не вказуємо, тільки дані які вводимо, але потрібно ввести стільки даних скільки стовпців (9 стовпців, 9 даних).

INSERT INTO team

VALUES (52, 'Oksana', 'Prudnikova', 'female', 'QA Engineer', 1100, '1988-07-15', 'Kyiv', 'GolT');

| | id | name | surname | gender | position | salary | birth_date | city | school |
|--|----|------|---------|--------|----------|--------|------------|------|--------|
|--|----|------|---------|--------|----------|--------|------------|------|--------|

SQLQuery1.sql - D...P-FM713OU\1 (58))*

```
SELECT *
FROM team;
```

100 %

Результаты

| | id | name | surname | gender | position | salary | birth_date | city | school |
|----|----|----------|-------------|--------|----------------------|---------|------------|-----------|-----------------|
| 47 | 47 | Oleksiy | Gonchare... | male | AI Engineer | 3200,00 | 1990-09-06 | Sumy | Brain Academy |
| 48 | 48 | Maksim | Litvin | male | IT Architect | 3000,00 | 1988-05-11 | Kharkiv | Data Science UA |
| 49 | 49 | Irina | Tymoshen... | female | Data Scientist | 3100,00 | 1986-09-28 | Vinnytsia | GoIT |
| 50 | 50 | Valeriya | Lazarenko | female | Junior Web Develo... | 2300,00 | 1991-01-02 | Kyiv | NULL |
| 51 | 51 | Yaryna | Berezovska | female | QA Engineer | 1100,00 | 1990-04-25 | Dnipro | Data Science UA |
| 52 | 52 | Oksana | Prudnikova | female | QA Engineer | 1100,00 | 1988-07-15 | Kyiv | GoIT |

Запрос усп... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (58) | QA24 | 00:00:00 | 52 строки

Оператор **UPDATE** використовується для зміни даних в таблиці. Цей оператор дозволяє оновити значення стовпців в одному або багатьох рядках таблиці згідно з певними умовами.

Приклад. У співробітника Valeria Lazarenko (id = 50) змінилась зарплата з 2300 до 2400

```
UPDATE team
SET Salary = 2400
WHERE id = 50;
```

SQLQuery1.sql - D...P-FM713OU\1 (58))*

```
UPDATE team
SET Salary = 2400
WHERE id = 50;
```

100 %

Сообщения

(затронута одна строка)

Время выполнения: 2023-07-01T23:08:39.9333923+03:00

100 %

Запрос успе... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (58) | QA24 | 00:00:00 | 0 строки

SQLQuery1.sql - D...P-FM713OU\1 (58))*

```
SELECT *
From team
WHERE id = 50;
```

100 %

Результаты

| | id | name | surname | gender | position | salary | birth_date | city | school |
|---|----|----------|-----------|--------|----------------------|---------|------------|------|--------|
| 1 | 50 | Valeriya | Lazarenko | female | Junior Web Developer | 2400,00 | 1991-01-02 | Kyiv | NULL |

Запрос успе... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (58) | QA24 | 00:00:00 | 1 строки

Оператор **DELETE** використовується для видалення рядків з таблиці бази даних. Він дозволяє видаляти один або багато рядків в таблиці згідно з певними умовами.

Важливо пам'ятати, що оператор DELETE без умови WHERE може видалити всі рядки таблиці і повністю очистити її вміст. Тому будьте обережні при використанні цього оператора і переконайтеся, що ви вказали правильну умову для видалення потрібних рядків.

Приклад. Нам потрібно видалити співробітника з Id - 52

```
DELETE FROM team
WHERE id LIKE 52;
```

SQLQuery1.sql - D...P-FM713OU\1 (58))*

```
DELETE FROM team
WHERE id LIKE 52
```

100 %

Сообщения

(затронуто строк: 1)

Время выполнения: 2023-07-01T23:18:03.3151431+03:00

100 %

Запрос успе... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (58) | QA24 | 00:00:00 | 0 строки

SQLQuery1.sql - D...P-FM713OU\1 (58))*

```
SELECT *
FROM team
```

100 %

Результаты

| | id | name | surname | gender | position | salary | birth_date | city | school |
|----|----|-----------|-------------|--------|----------------------|---------|------------|-------|-----------------|
| 47 | 47 | Oleksiy | Gonchare... | male | AI Engineer | 3200,00 | 1990-09-06 | S... | Brain Academy |
| 48 | 48 | Maks... | Litvin | male | IT Architect | 3000,00 | 1988-05-11 | K... | Data Science UA |
| 49 | 49 | Irina | Tymoshen... | female | Data Scientist | 3100,00 | 1986-09-28 | Vi... | GoIT |
| 50 | 50 | Valeri... | Lazarenko | female | Junior Web Develo... | 2300,00 | 1991-01-02 | Kyiv | NULL |
| 51 | 51 | Yaryna | Berezovska | female | QA Engineer | 1100,00 | 1990-04-25 | D... | Data Science UA |

Запрос успе... | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (58) | QA24 | 00:00:00 | 51 строки

За допомогою оператора **JOIN** можна поєднувати дані з двох або більше таблиць на основі спільного стовпця.

Існують різні типи JOIN:

INNER JOIN — повертає записи, які мають значення, що збігаються в обох таблицях;

LEFT JOIN — повертає всі записи з лівої таблиці та відповідні записи з правої таблиці;

RIGHT JOIN — повертає всі записи з правої таблиці та відповідні записи з лівої таблиці;

FULL OUTER JOIN — повертає всі записи з лівої та правої таблиці.

Для того щоб зробити приклад запиту з двох таблиць зробив нову таблицю Team2. Де стовбці **id** = номери від 1 до 50, **project** = назва проекту, на якому працює співробітник. Також додав в таблицю Team новий стовпчик **project_id**, по якому будемо зв'язувати id з двох таблиць

Нова таблиця Team2

SQLQuery2.sql - D...P-FM713OU\1 (71))* SQLQuery1.sql - D...P-FM713OU\1 (56))*

```
SELECT *
FROM team2;
```

100 %

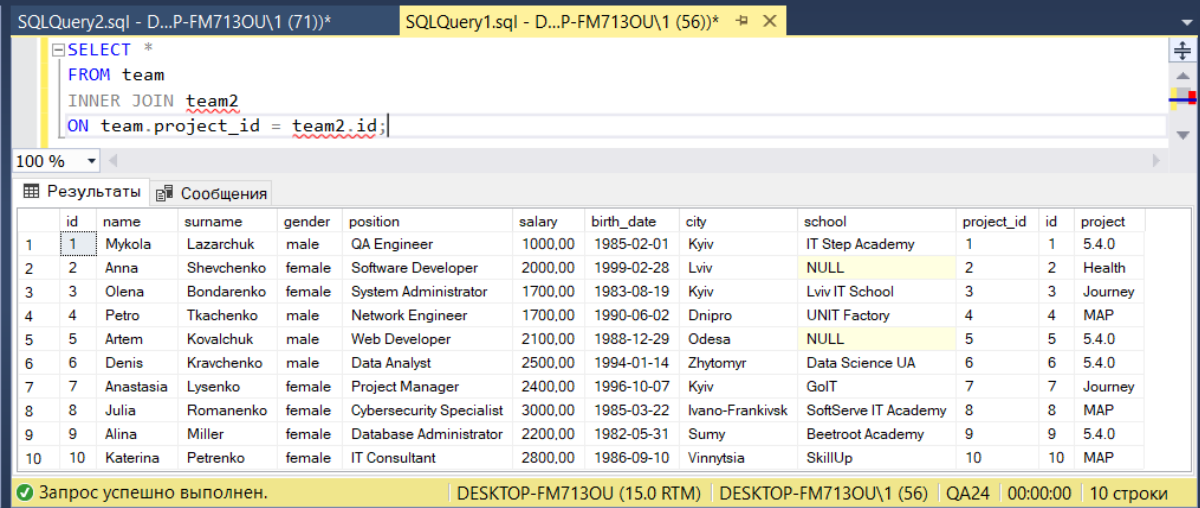
Результаты

| | id | project |
|----|----|---------|
| 1 | 1 | 5.4.0 |
| 2 | 2 | Health |
| 3 | 3 | Journey |
| 4 | 4 | MAP |
| 5 | 5 | 5.4.0 |
| 6 | 6 | 5.4.0 |
| 7 | 7 | Journey |
| 8 | 8 | MAP |
| 9 | 9 | 5.4.0 |
| 10 | 10 | MAP |

Запрос успешно выполнен. | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (71) | QA24 | 00:00:00 | 50 строки

Тепер зробимо запит, який виведе всю інформацію з двох таблиць.

```
SELECT *  
FROM team  
INNER JOIN team2  
ON team.project_id = team2.id;
```



SQLQuery2.sql - D...P-FM713OU\1 (71)*

```
SELECT *  
FROM team  
INNER JOIN team2  
ON team.project_id = team2.id;
```

100 %

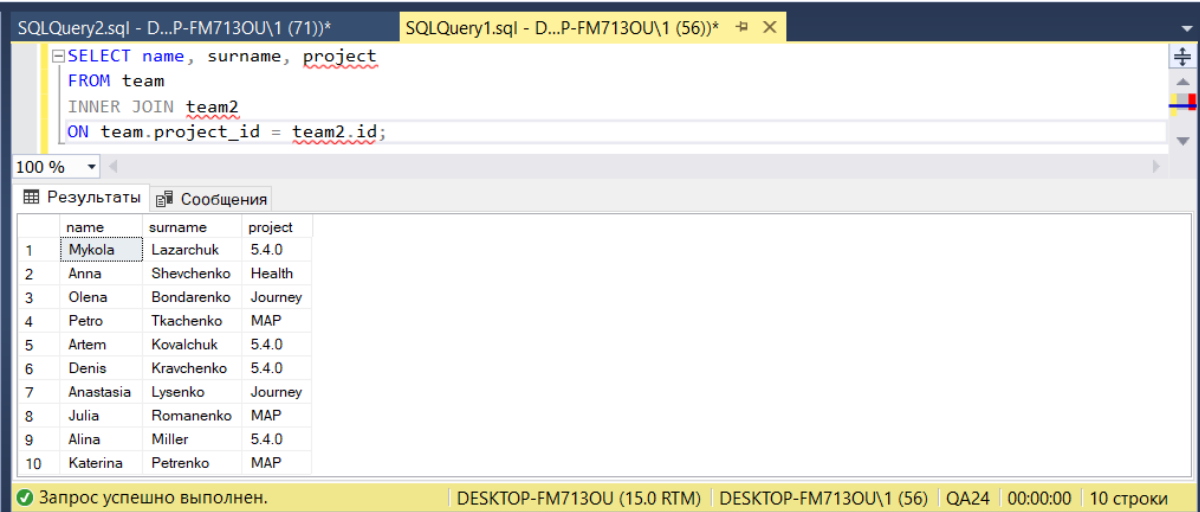
Результаты Сообщения

| | id | name | surname | gender | position | salary | birth_date | city | school | project_id | id | project |
|----|----|-----------|------------|--------|--------------------------|---------|------------|-----------------|----------------------|------------|----|---------|
| 1 | 1 | Mykola | Lazarchuk | male | QA Engineer | 1000.00 | 1985-02-01 | Kyiv | IT Step Academy | 1 | 1 | 5.4.0 |
| 2 | 2 | Anna | Shevchenko | female | Software Developer | 2000.00 | 1999-02-28 | Lviv | NULL | 2 | 2 | Health |
| 3 | 3 | Olena | Bondarenko | female | System Administrator | 1700.00 | 1983-08-19 | Kyiv | Lviv IT School | 3 | 3 | Journey |
| 4 | 4 | Petro | Tkachenko | male | Network Engineer | 1700.00 | 1990-06-02 | Dnipro | UNIT Factory | 4 | 4 | MAP |
| 5 | 5 | Artem | Kovalchuk | male | Web Developer | 2100.00 | 1988-12-29 | Odesa | NULL | 5 | 5 | 5.4.0 |
| 6 | 6 | Denis | Kravchenko | male | Data Analyst | 2500.00 | 1994-01-14 | Zhytomyr | Data Science UA | 6 | 6 | 5.4.0 |
| 7 | 7 | Anastasia | Lysenko | female | Project Manager | 2400.00 | 1996-10-07 | Kyiv | GolT | 7 | 7 | Journey |
| 8 | 8 | Julia | Romanenko | female | Cybersecurity Specialist | 3000.00 | 1985-03-22 | Ivano-Frankivsk | SoftServe IT Academy | 8 | 8 | MAP |
| 9 | 9 | Alina | Miller | female | Database Administrator | 2200.00 | 1982-05-31 | Sumy | Beetroot Academy | 9 | 9 | 5.4.0 |
| 10 | 10 | Katerina | Petrenko | female | IT Consultant | 2800.00 | 1986-09-10 | Vinnitsia | SkillUp | 10 | 10 | MAP |

Запрос успешно выполнен. | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (56) | QA24 | 00:00:00 | 10 строки

Для того щоб побачити хто працює на якому проекті, робим наступний запит.

```
SELECT name, surname, project  
FROM team  
INNER JOIN team2  
ON team.project_id = team2.id;
```



SQLQuery2.sql - D...P-FM713OU\1 (71)*

```
SELECT name, surname, project  
FROM team  
INNER JOIN team2  
ON team.project_id = team2.id;
```

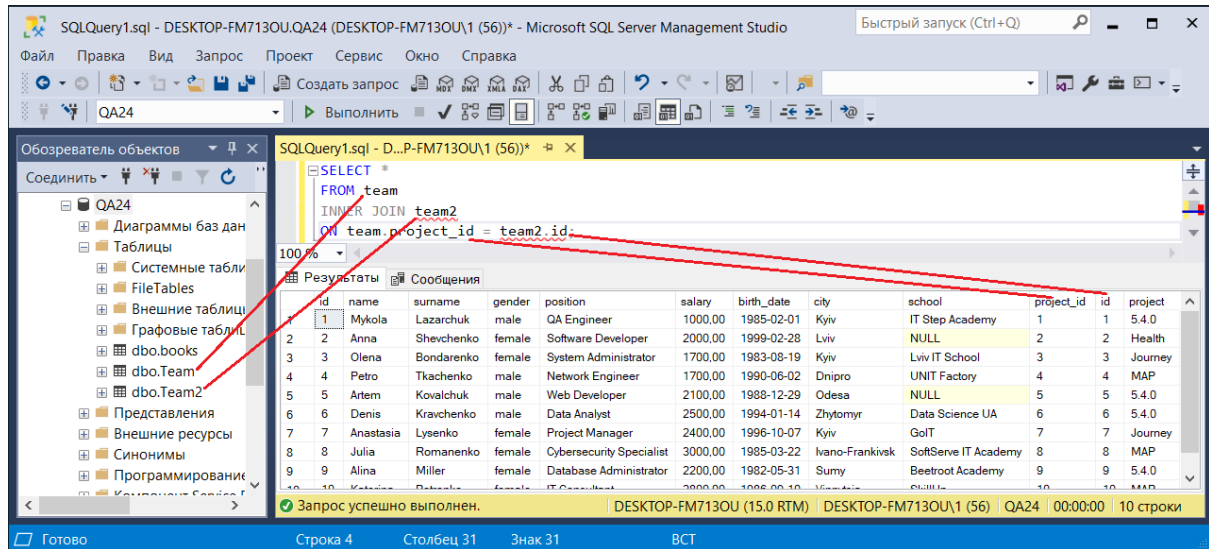
100 %

Результаты Сообщения

| | name | surname | project |
|----|-----------|------------|---------|
| 1 | Mykola | Lazarchuk | 5.4.0 |
| 2 | Anna | Shevchenko | Health |
| 3 | Olena | Bondarenko | Journey |
| 4 | Petro | Tkachenko | MAP |
| 5 | Artem | Kovalchuk | 5.4.0 |
| 6 | Denis | Kravchenko | 5.4.0 |
| 7 | Anastasia | Lysenko | Journey |
| 8 | Julia | Romanenko | MAP |
| 9 | Alina | Miller | 5.4.0 |
| 10 | Katerina | Petrenko | MAP |

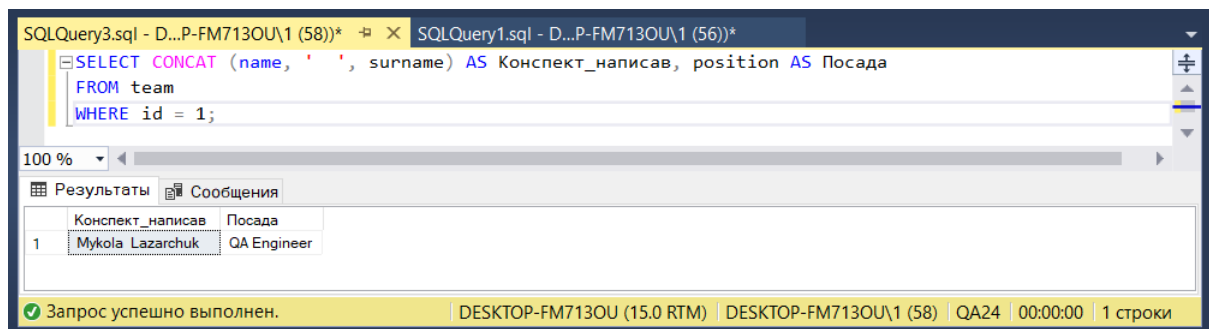
Запрос успешно выполнен. | DESKTOP-FM713OU (15.0 RTM) | DESKTOP-FM713OU\1 (56) | QA24 | 00:00:00 | 10 строки

Для того щоб зрозуміти що таке багатотабличний запит, треба подивитись на наступну картинку



Наступний запит виведе автора конспекта та його посаду.

```
SELECT CONCAT (name, ' ', surname) AS Конспект_написав, position AS Посада
FROM team
WHERE id = 1;
```



**Дякую що прочитали, сподіваюсь мій конспект
допоможе вам в вивченні SQL**

