

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»



**Т. В. Хом'як, К. С. Хабарлак, Д.М. Гаранжа**

---

## **БАЗИ ДАНИХ У ПРОФЕСІЙНИХ ЗАДАЧАХ АНАЛІТИКИ**

Навчальний наочний посібник

Дніпро  
НТУ «ДП»  
2024



УДК 004.65:001.82(075.8)

X76

Рекомендовано

*вченою радою НТУ «Дніпровська політехніка» як навчальний наочний посібник  
для здобувачів ступеня магістра спеціальності 124 Системний аналіз (протокол № 8 від 27.06.2024 року)*

Рецензенти:

Д. В. Широкоград – канд. фіз.-мат. наук, доц. (Національний університет «Запорізька політехніка»);

Д. Г. Зеленцов – д-р техн. наук, проф. (Український державний університет науки і технологій).

**Хом'як Т. В.**

X76 Бази даних у професійних задачах аналітики [Електронний ресурс] : навч. наочн. посіб. / Т. В. Хом'як, К. С. Хабарлак, Д.М. Гаранжа; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 192с.

Викладено основи мови створення запитів до бази даних SQL, розглянуто структуру SQL-запитів, аналітичні функції мови як найбільш популярного інструменту для проведення аналізу інформації, що зберігається в базах даних. З'ясовано застосування віконних функцій у проведенні аналітичних досліджень і встановлено їхню перевагу порівняно із звичайними функціями агрегації, а також описано способи візуалізації розрахованих даних у процесі розв'язування професійних задач аналітики.

Видання включає відомості й приклади розв'язування задач, що дозволяє краще засвоїти матеріал і відобразити результати аналізу у вигляді дашбордів та інтерактивних звітів за допомогою додатка Power BI і будувати OLAP-куби.

Для здобувачів ступеня магістра спеціальності 124 Системний аналіз.

**УДК 004.65:001.82(075.8)**

© Т. В. Хом'як, К. С. Хабарлак, Д.М. Гаранжа, 2024

© НТУ «Дніпровська політехніка», 2024

# Зміст

<u>ВСТУП</u>	4
<u>Тема 1. Огляд провідних інструментів СУБД</u>	5
<u>Тема 2. Створення запитів мовою SQL. Оператор SELECT</u>	32
<u>Тема 3. Використання функцій агрегації та сортування даних у SQL-запитах</u>	76
<u>Тема 4. Реляційна модель даних</u>	104
<u>Тема 5. Використання віконних функцій для написання SQL-запитів</u>	143
<u>Тема 6. Технологія ETL. Візуалізація даних</u>	159
<u>Тема 7. Проектування сховищ даних і створення OLAP-куба для роз'язування професійних аналітичних задач</u>	184



# ВСТУП

Найбільш популярним інструментом у роботі аналітика є база даних, що являє собою фундамент будь-якої системи. Усі бізнес-процеси та функції зводяться до одного завдання – обробки даних. Ефективність прийняття бізнес-рішень залежить від того, наскільки якісно та правильно було проаналізовано дані. Адже саме на основі такого аналізу будуються гіпотези, проводиться тестування та генеруються відповідні рішення. Ключові навички системного аналітика – володіння мовою SQL, розуміння структури SQL-запитів, основних аналітичних функцій мови. До того ж необхідне вміння використовувати віконні функції, складаючи запити.

Мета дисципліни – сформувати в здобувачів навички застосування ефективних інструментів відбору за певними критеріями структурованих і неструктурованих даних для проведення аналізу та візуалізації, що полегшує прийняття необхідних рішень у різних сферах діяльності.

Засвоєння здобувачем розглянутого в навчальному наочному посібнику матеріалу має забезпечити такі результати навчання:

- спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері системного аналізу та інформаційних технологій і є основою для оригінального мислення та проведення досліджень;
- розробляти інтелектуальні системи в умовах слабо структурованих даних різної природи;
- вільно презентувати та обговорювати усно і письмово результати досліджень та інновацій, інші питання професійної діяльності державною та англійською мовами;
- поєднувати технології ETL з можливостями Data Mining і сховищ інформації для швидкої обробки великих масивів багатовимірних даних, виявлення наявних закономірностей та їх візуалізації.

Текст посібника поділено на сім тем відповідно до робочої програми дисципліни. Усі теми включають задачі для самостійного розв’язування, які, без сумніву, будуть сприяти кращому засвоєнню навчального матеріалу. Користуючись цим виданням, здобувач навчиться ефективно аналізувати результати SQL-запитів та правильно їх інтерпретувати для бізнесу, а також виробить вміння представляти результати дослідження у вигляді дашбордів та інтерактивних звітів у додатку Power BI, та будувати OLAP-куби.

Посібник створено з використанням демонстраційного матеріалу лекцій, читаних авторами перед здобувачами НТУ «Дніпровська політехніка» з дисципліни «Бази даних в професійних задачах аналітики». Значну частку прикладів, наведених у цьому виданні, взято з бази даних про співробітників фірми, розміщеної у відкритих джерелах.



# Тема 1

## Огляд провідних інструментів СУБД

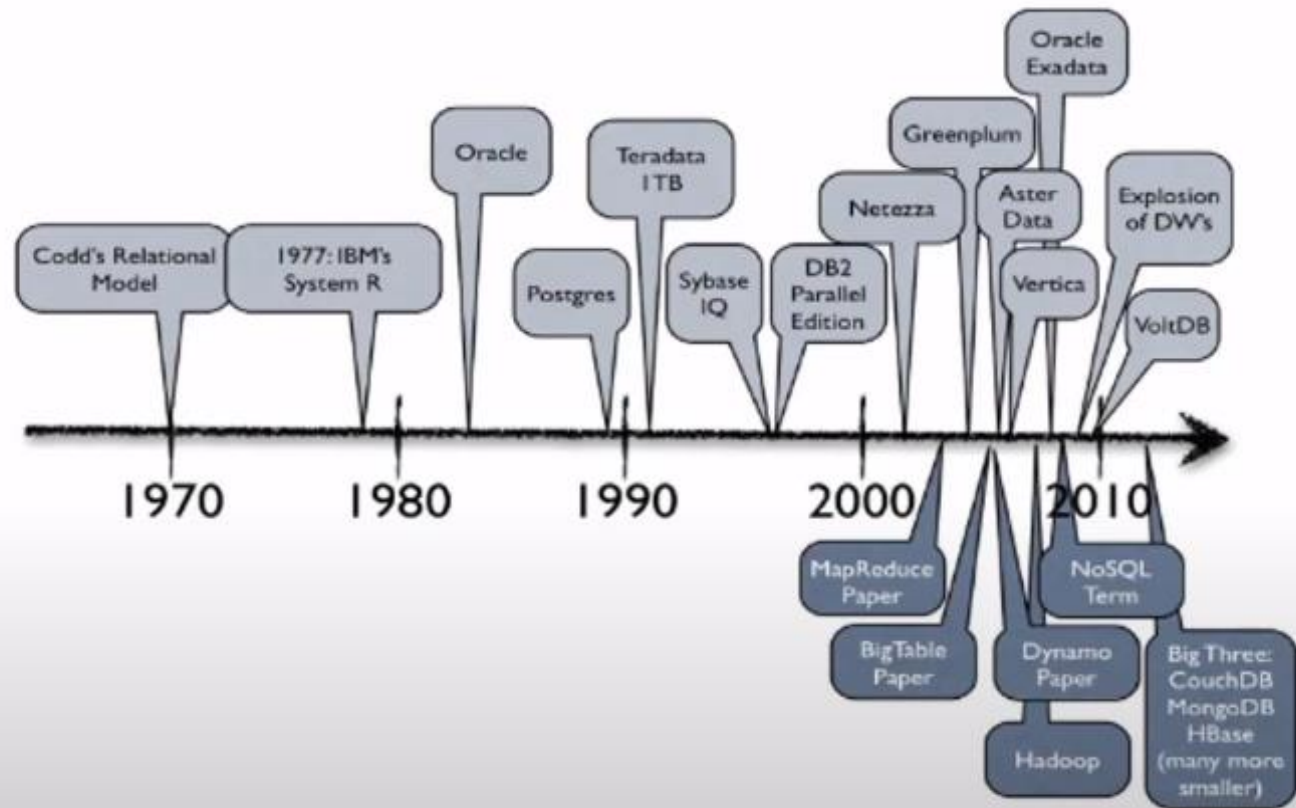
**База даних** - це структура даних, яка зберігає організовану інформацію.

**СУБД** - система управління базами даних: пакет програмного забезпечення з комп'ютерними програмами, що контролює створення, обслуговування та використання бази даних

Бази даних створюються для управління великими обсягами інформації шляхом введення, зберігання, витягування та управління цією інформацією.

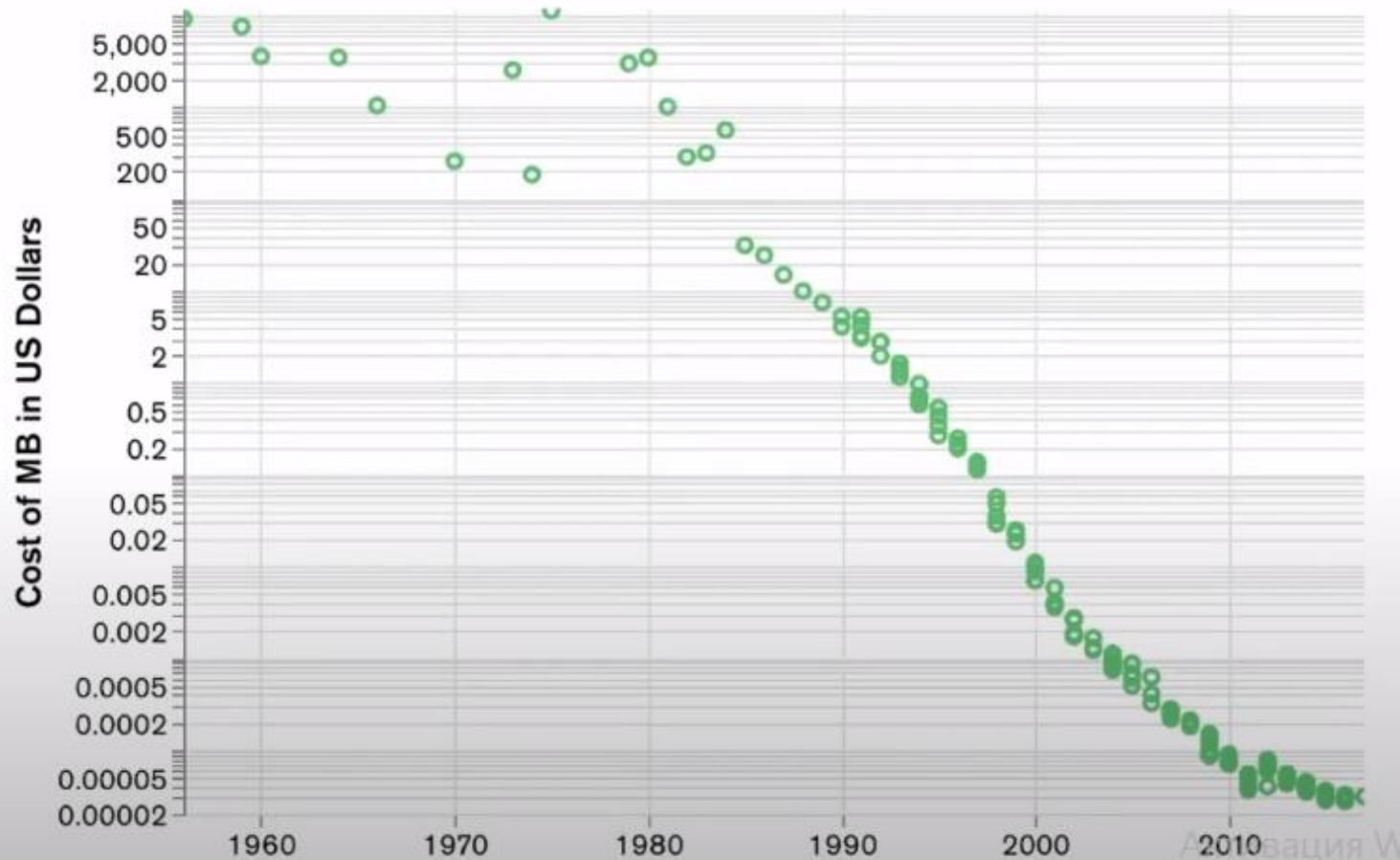


# Історія розвитку



# Історія розвитку

Cost Per MB of Data Over Time (Log Scale)



# Типи Баз Даних

Реляційні  
Мають тенденцію бути  
більшими, монолітними



Нереляційні  
Легко додати  
нові поля, багато  
гравців на ринку

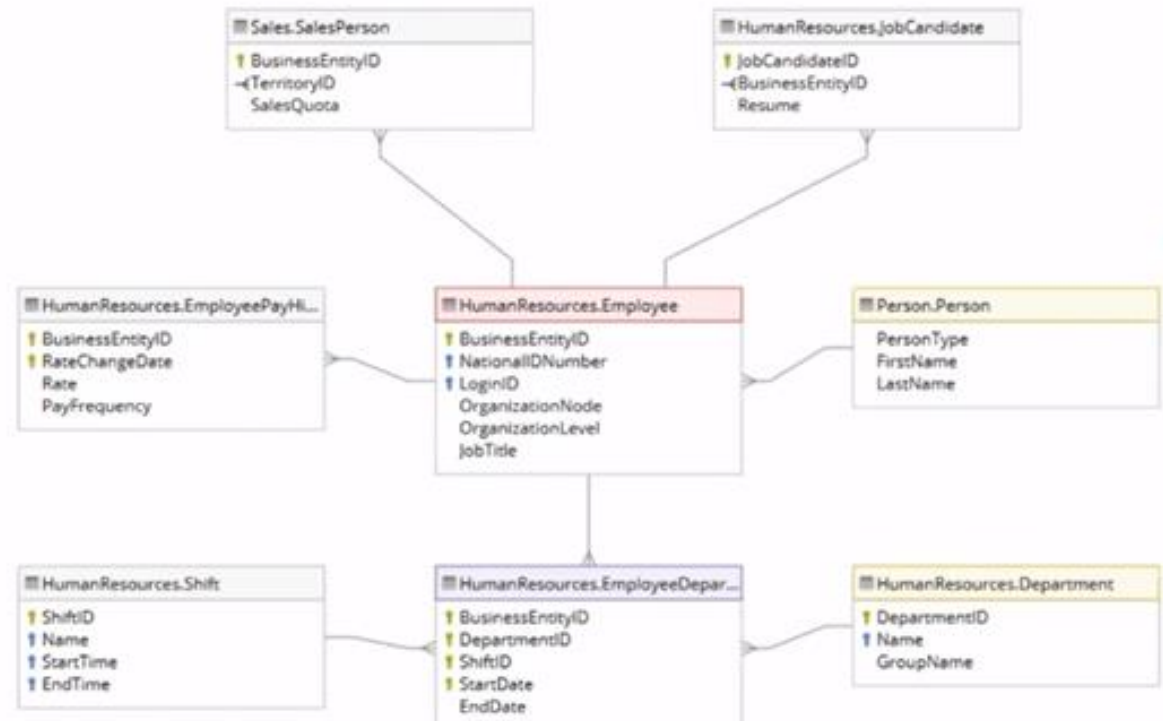




# Типи Баз Даних

**Системи управління реляційними базами даних** (СУБД) підтримують реляційну модель даних. Схема таблиці (схема зв'язків) визначається назвою таблиці та фіксованою кількістю атрибутів із фіксованими типами даних. Запис відповідає рядку в таблиці і складається зі значень кожного атрибута. Таким чином, зв'язок складається з набору єдиних записів. Схеми таблиць формуються шляхом нормалізації в процесі моделювання даних.

## Приклад схеми структурованої БД



# MySQL Server

MySQL Server - одна з найбільш широко використовуваних систем управління реляційними базами даних. За оцінками, понад 100 мільйонів копій сервера MySQL були завантажені по всьому світу приватними особами, корпоративними компаніями та невеликими організаціями.

MySQL - це безкоштовний сервер баз даних з відкритим вихідним кодом, який надає численні розширені функції баз даних. Google, Facebook, Вікіпедія, GitHub, Youtube та багато інших відомих організацій застосували MySQL для обробки даних.



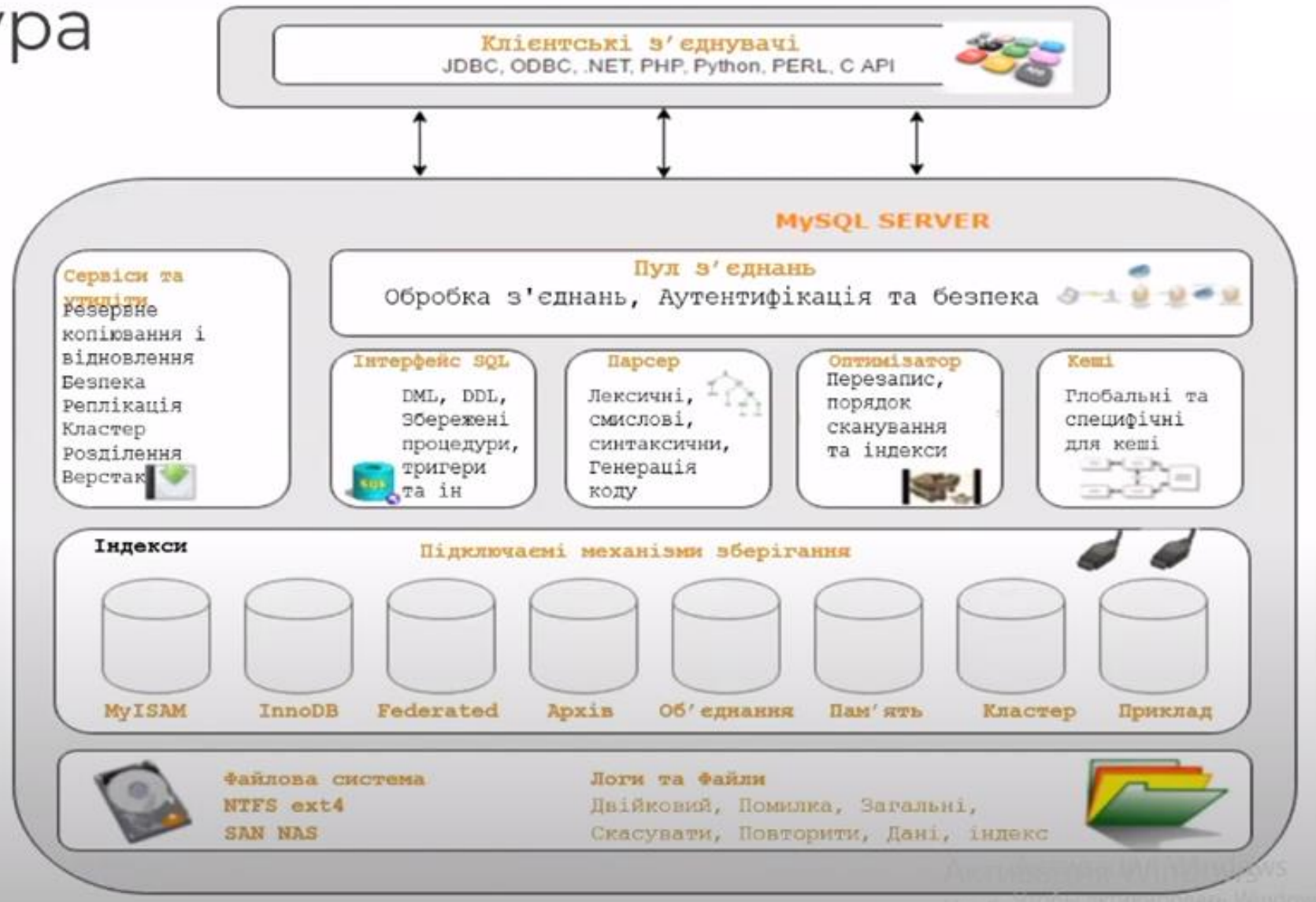
# Історія MySQL

MySQL було створено шведською компанією MySQL AB, заснованою Девідом Аксмарком, Алланом Ларссоном та Майклом "Монті" Віденіусом. Початкова розробка MySQL компаніями Widenius та Axmark розпочалася у 1994 році.

- Перший внутрішній випуск відбувся 23 травня 1995 року
- Sun Microsystems придбала MySQL AB у 2008 році.
- Oracle придбала Sun Microsystems 27 січня 2010 року
- MySQL оголошено СУБД 2019 року з рейтингу DB-Engines



# Архітектура



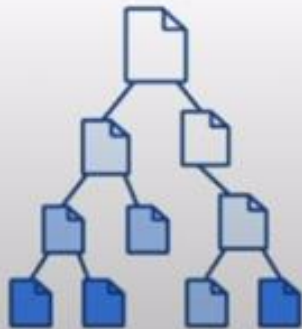
# Типи Баз Даних

Системи баз даних **NoSQL** є альтернативою поширеним реляційним СУБД. Вони не використовують реляційну модель даних і зазвичай не мають інтерфейсу SQL.

Абревіатура NoSQL часто розуміється як "не тільки SQL", що означає, що реляційні системи є перевіреною технологією, але не обов'язково оптимальним вибором для кожного виду передбачуваного використання.

## Типи NoSQL

Документ



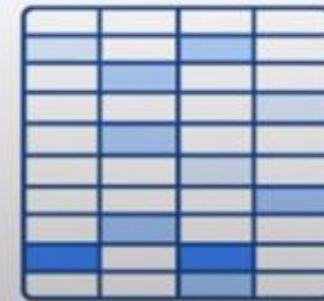
Граф



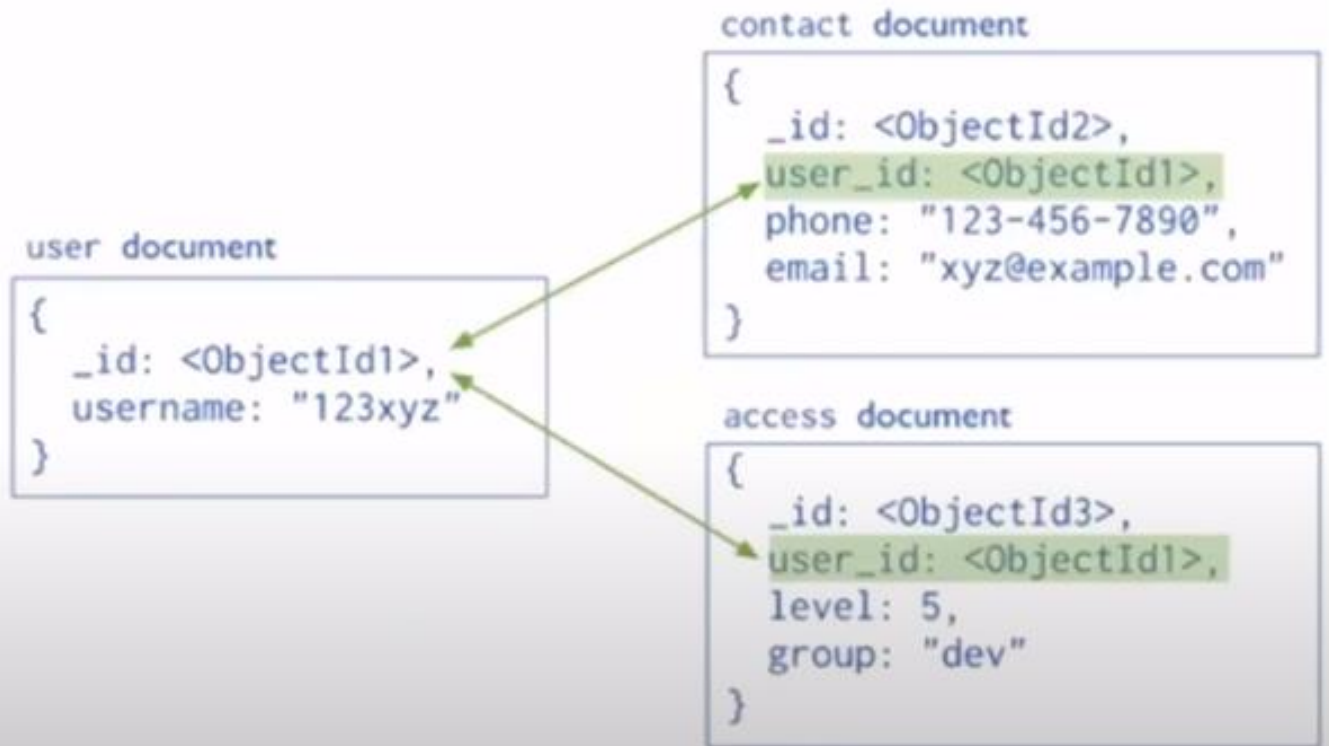
Ключ-Значення



Колоночні



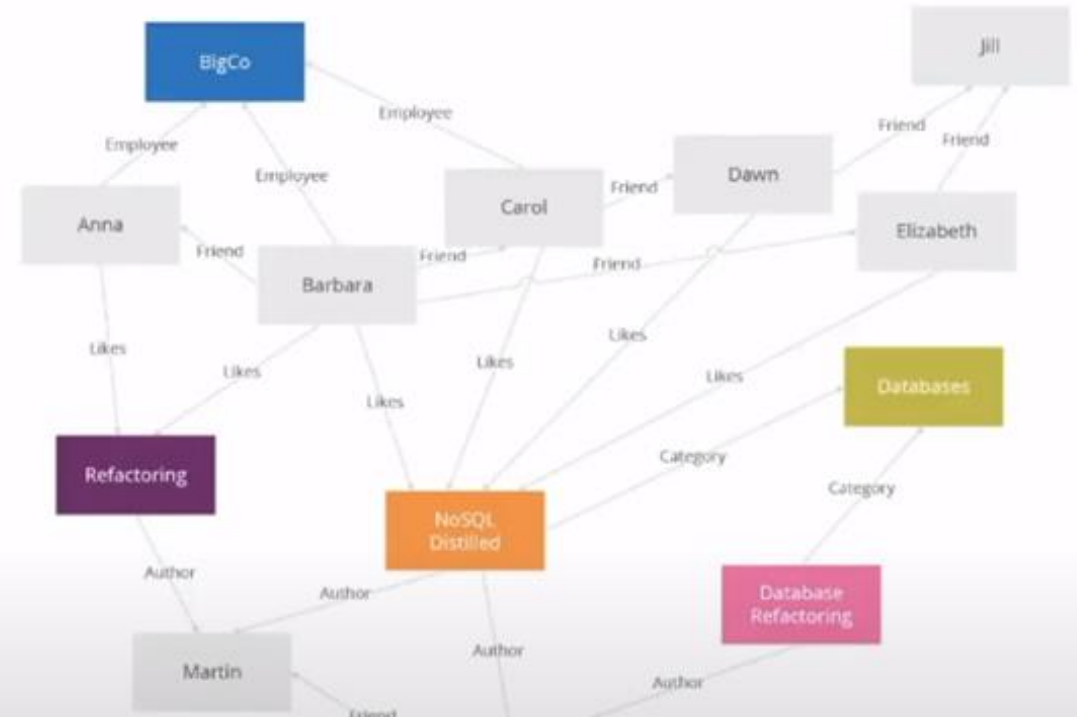
# MongoDB



# Redis



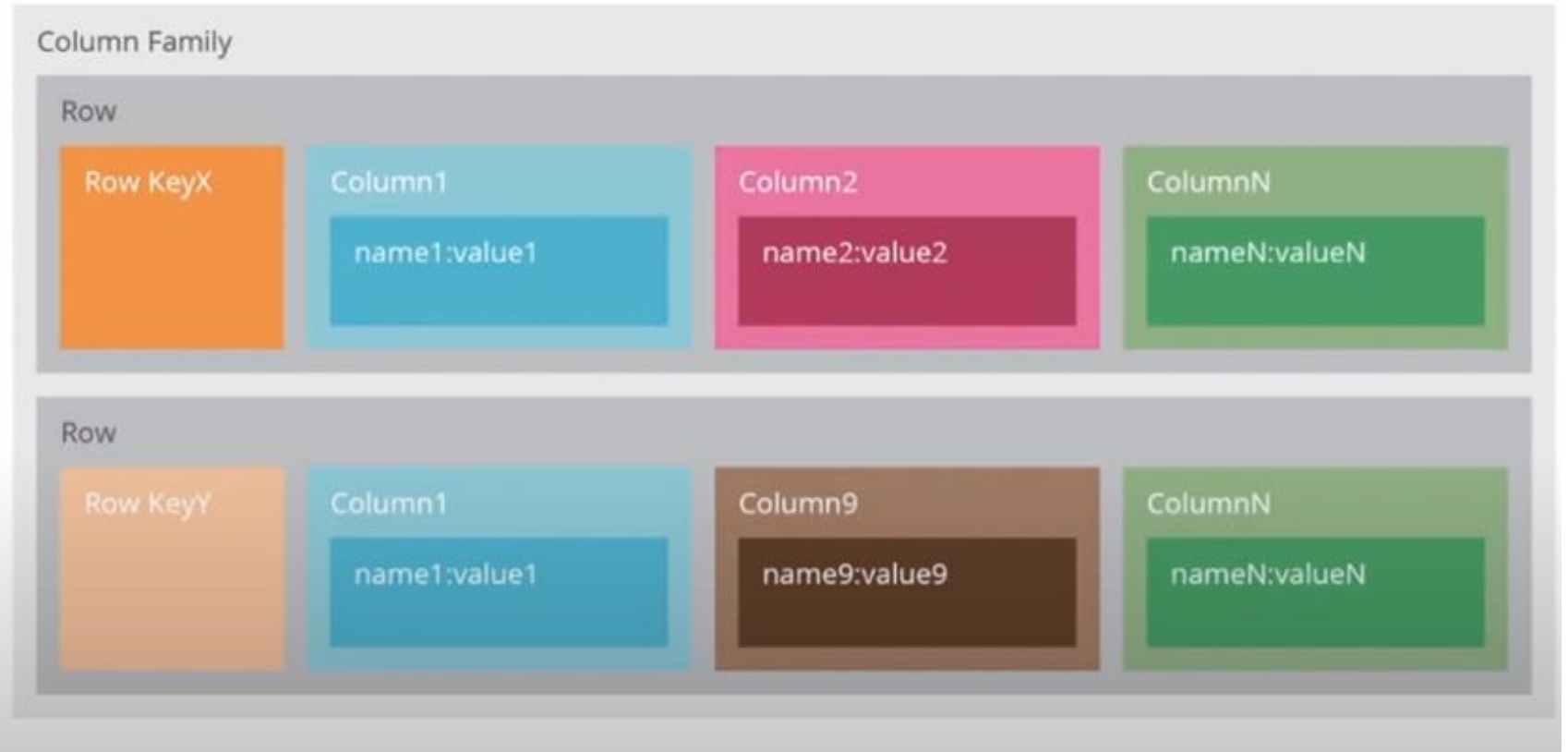
Neo4j





# Колоночні бази

Cassandra  
Vertica  
Clickhouse  
...



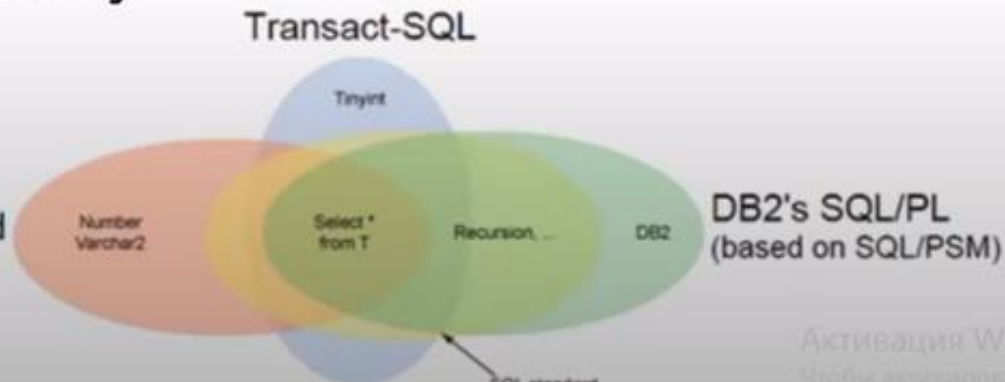
# T-SQL

**T-SQL** (Transact-SQL)- це власне розширення процедурної мови для SQL Server. Він містить інструкції REPL (Read-Eval -Print - Loop - цикл читання-обчислення-друку), які **розширюють набір стандартних інструкцій** SQL для маніпулювання та визначення даних, таких як специфічні параметри SQL Server, управління безпекою та статистикою бази даних, контроль транзакцій, обробка винятків та помилок, обробка рядку та оголошені змінні.

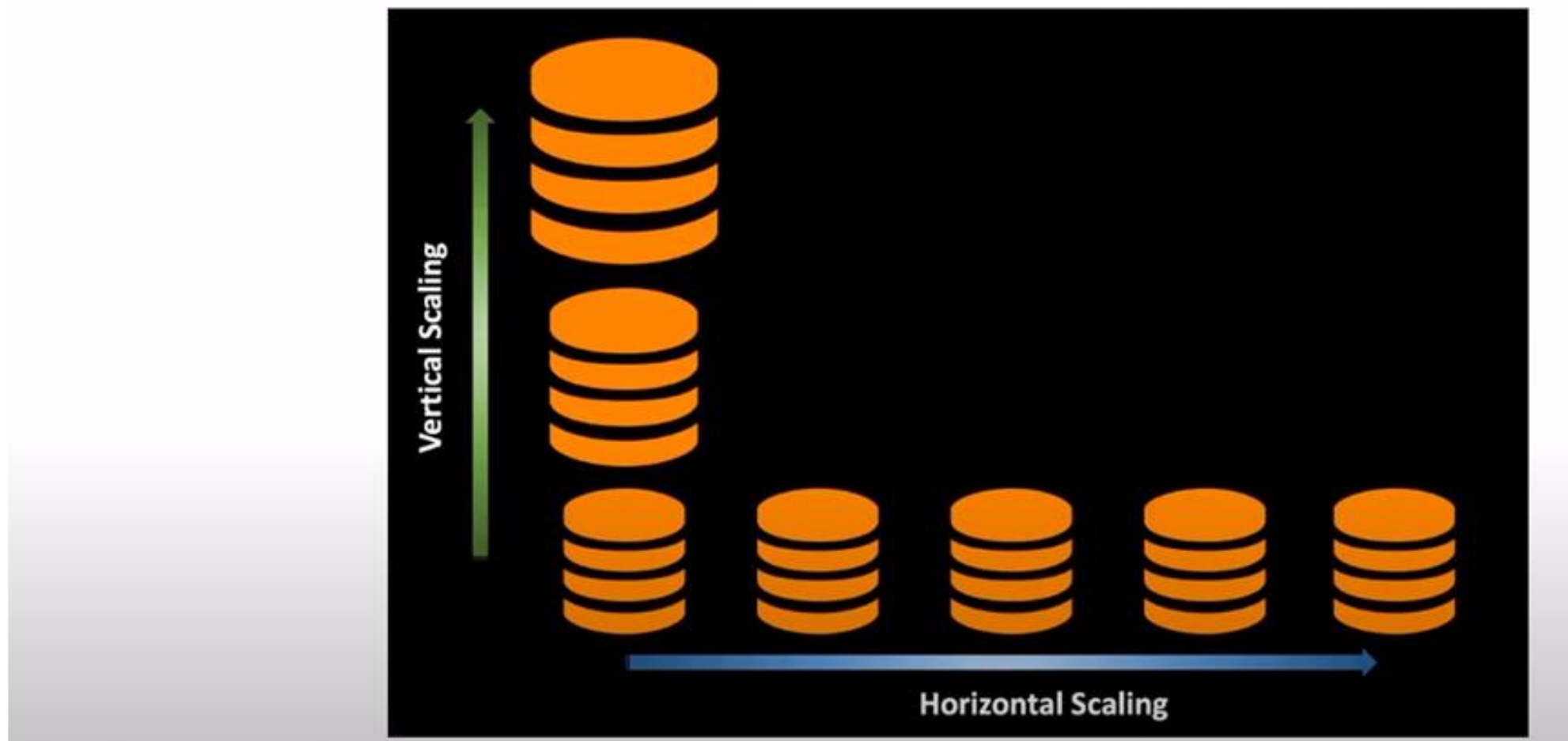
Він надає ключові слова для операцій, які можуть виконуватися на SQL Server, включаючи створення і зміна схем бази даних, введення та редагування даних в базі даних, а також моніторинг і управління самим сервером. **Пов'язані сервери дозволяють обробляти операції, що виконуються на декількох серверах, за допомогою одного запиту.**

Різниця між SQL,  
T-SQL та PL/SQL

Oracle  
SQL and  
PL/SQL



# Масштабування БД



# MySQL Server. Інсталяція та параметри налаштування

<https://dev.mysql.com/downloads/workbench/>

**General Availability (GA) Releases** Archives ⓘ

## MySQL Workbench 8.0.34

Select Operating System:  
Microsoft Windows ▾

**Recommended Download:**

**MySQL Installer**  
for Windows

All MySQL Products. For All Windows Platforms.  
In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

**Windows (x86, 32 & 64-bit), MySQL Installer MSI** [Go to Download Page >](#)

**Other Downloads:**

<b>Windows (x86, 64-bit), MSI Installer</b> <small>(mysql-workbench-community-8.0.34-winx64.msi)</small>	8.0.34	46.4M	<a href="#">Download</a>
	<small>MD5: eF5294cd0807979c060e1808Fc488006   Signature</small>		



MySQL Installer

# MySQL Installer

Adding Community

- Choosing a Setup Type
- Select Products
- Download
- Installation
- Installation Complete

## Choosing a Setup Type

Please select the Setup Type that suits your use case.

- Developer Default  
Installs all products needed for MySQL development purposes.
- Server only  
Installs only the MySQL Server product.
- Client only  
Installs only the MySQL Client products, without a server.
- Full  
Installs all included MySQL products and features.
- Custom  
Manually select the products that should be installed on the system.

Setup Type Description

Allows you to select exactly which products you would like to install. This also allows to pick other server versions and architectures (depending on your OS).

Next

MySQL Installer

# MySQL Installer

Adding Community

- Choosing a Setup Type
- Select Products
- Download
- Installation
- Installation Complete

## Select Products

Please select the products you would like to install on this computer.

Filter: All Software, Current GA, Any Edit

Available Products:

- MySQL Server 8.0
  - MySQL Server 8.0.26 - X64
  - MySQL Server 8.0.25 - X64
  - MySQL Server 8.0.24 - X64
  - MySQL Server 8.0.23 - X64
  - MySQL Server 8.0.22 - X64**
  - MySQL Server 8.0.21 - X64
  - MySQL Server 8.0.20 - X64
  - MySQL Server 8.0.19 - X64
  - MySQL Server 8.0.18 - X64
  - MySQL Server 8.0.17 - X64
  - MySQL Server 8.0.16 - X64
  - MySQL Server 8.0.15 - X64
  - MySQL Server 8.0.14 - X64

Products To Be Installed:

Enable the Select Features page to customize product features



# MySQL. Installer

MySQL Server 8.0.22

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Apply Configuration

## Type and Networking

### Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type: 

### Connectivity

Use the following controls to select how you would like to connect to this server.

- TCP/IP Port:  X Protocol Port:
- Open Windows Firewall ports for network access
- Named Pipe Pipe Name:
- Shared Memory Memory Name:

### Advanced Configuration

Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.

- Show Advanced and Logging Options

Next &gt;

# MySQL. Installer

MySQL Server 8.0.22

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Apply Configuration

## Accounts and Roles

### Root Account Password

Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: Repeat Password: Password strength: **Weak**

### MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL User Name	Host	User Role	
			<input type="button" value="Add User"/>
			<input type="button" value="Edit User"/>
			<input type="button" value="Delete"/>


&lt; Back

Next &gt;

Cancel



MySQL Installer



Product Version Architecture Quick Action

Product	Version	Architecture	Quick Action
MySQL Server	8.0.22	X64	<a href="#">Reconfigure</a>

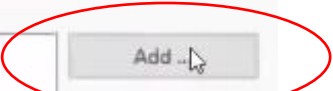
MySQL Installer

MySQL Installer

MySQL Installer

Adding Community

- Select Products
- Download
- Installation
- Installation Complete



Select Products

Please select the products you would like to install on this computer.

Filter: All Software, Current GA, Any Edit

Available Products:

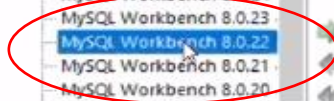
- MySQL Workbench 8.0
- MySQL Workbench 8.0.26
- MySQL Workbench 8.0.25
- MySQL Workbench 8.0.24
- MySQL Workbench 8.0.23
- MySQL Workbench 8.0.22
- MySQL Workbench 8.0.21
- MySQL Workbench 8.0.20
- MySQL Workbench 8.0.19
- MySQL Workbench 8.0.18
- MySQL Workbench 8.0.17
- MySQL Workbench 8.0.16
- MySQL Workbench 8.0.15

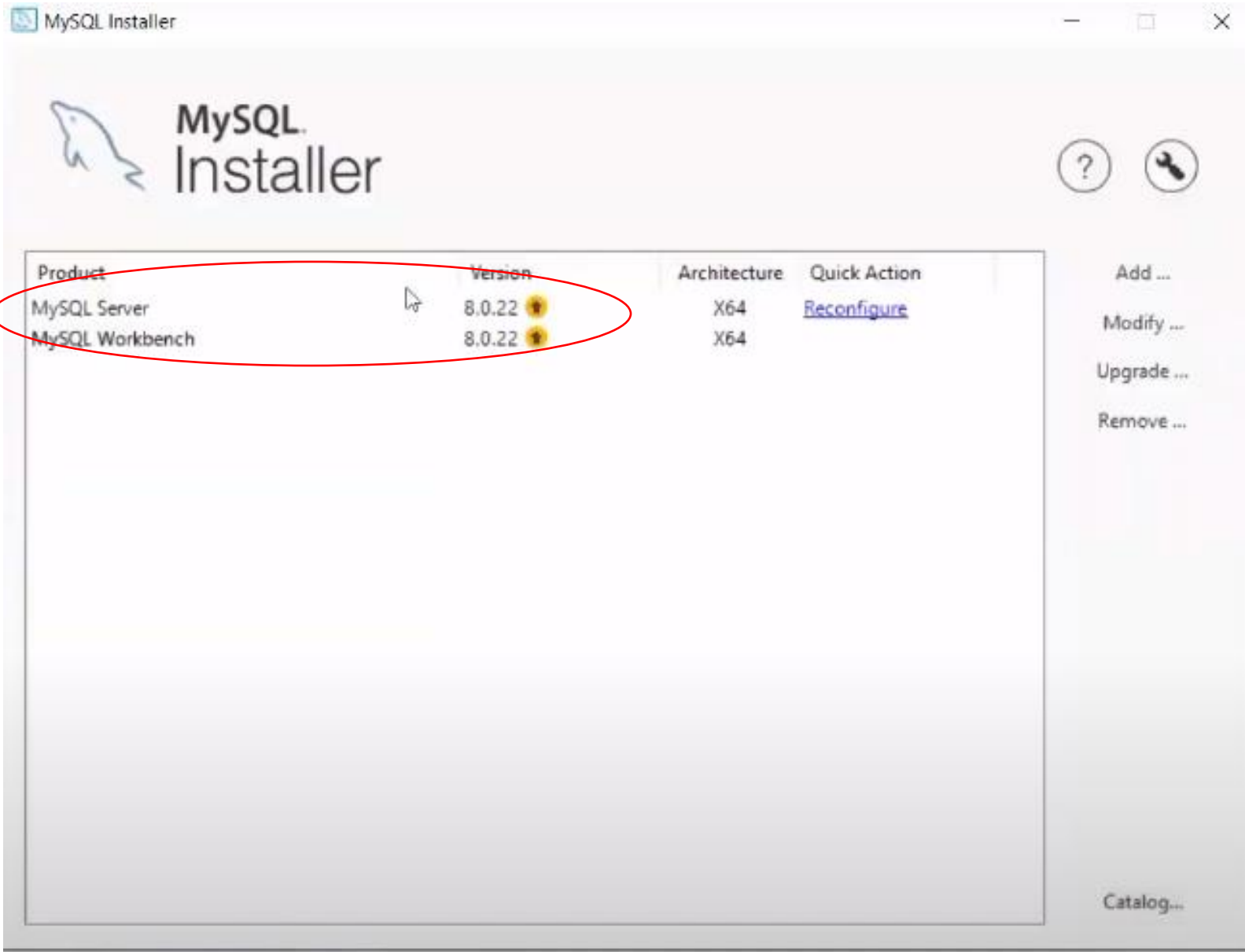
Products To Be Installed:

Enable the Select Features page to customize product features

Published: 11 xianw 2020 p.  
Release Notes: <https://dev.mysql.com/doc/relenotes/workbench/en/news-8-0-22.html>

Next > Cancel







File Edit View Database Tools Scripting Help

# Welcome to MySQL Workbench

Setup New Connection

Connection Name: test\_111 Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: 127.0.0.1 Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: root Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Configure Server Management...

MySQL Connections (+) (-)

RT (auto saved)

- i.homonets
- i.homonets@lgw.igenv.com

the Forums >

MySQL Workbench

**Successfully made the MySQL connection**

Information related to this connection:

Host: 127.0.0.1  
Port: 3306  
User: root  
SSL: enabled with TLS\_AES\_256\_GCM\_SHA384

A successful MySQL connection was made with the parameters defined for this connection.

OK



# Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

[Browse Documentation >](#)

[Read the Blog >](#)

[Discuss on the Forums >](#)

## MySQL Connections

RT (auto saved)	test_db	analytics	test_2	test111
root	root	root	root	root
127.0.0.1:3306	127.0.0.1:3306	127.0.0.1:3306	127.0.0.1:3306	127.0.0.1:3306

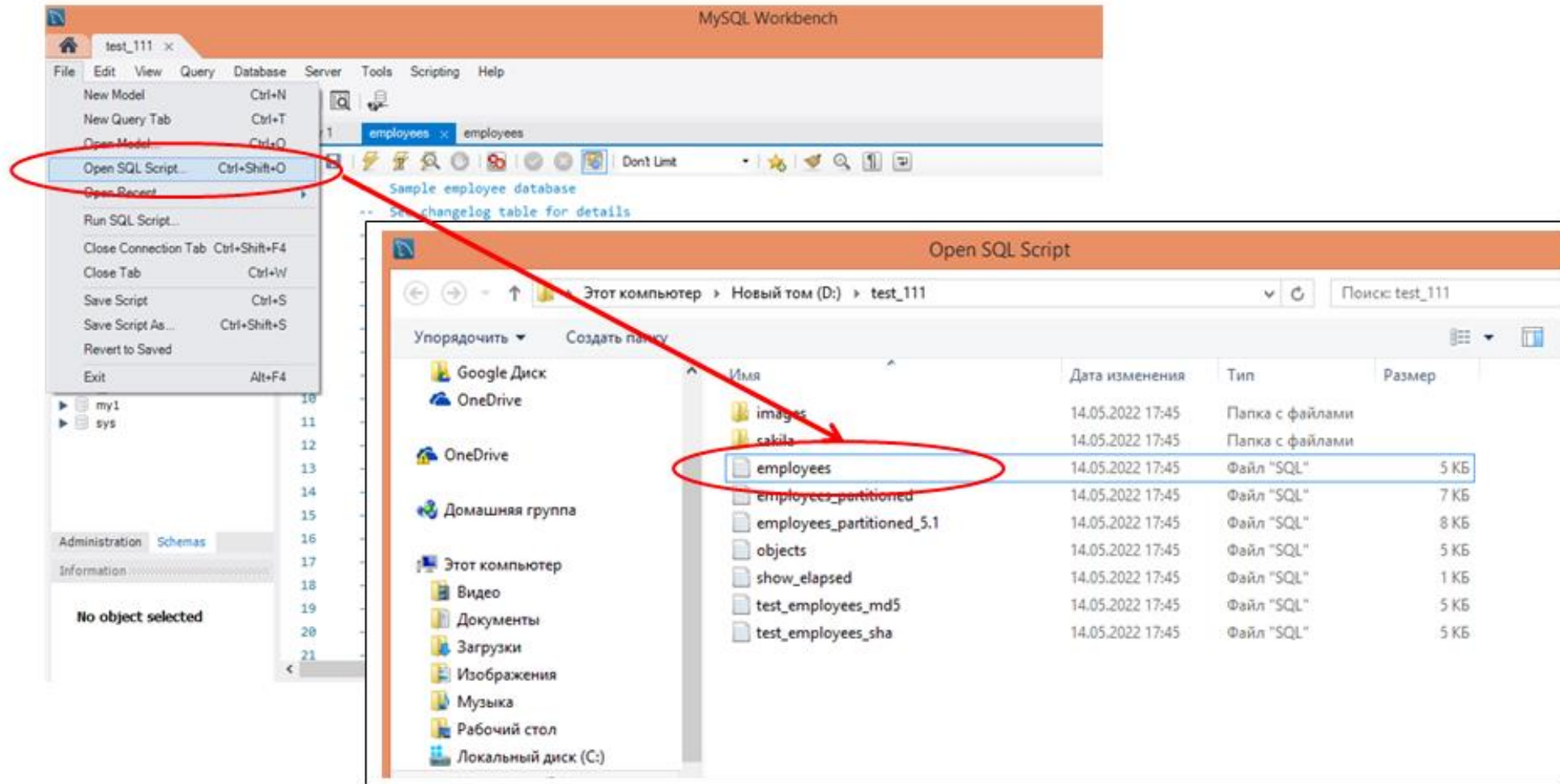
test_111
root
127.0.0.1:3306



The screenshot displays the SQL Server Enterprise Manager interface. At the top, a menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The left-hand side features a Navigator pane with a 'SCHEMAS' section containing a search box labeled 'Filter objects' and a tree view showing 'sys'. The main area is a query window titled 'Query 1' containing the SQL statement `select curdate();`. Below the query window is a 'Result Grid' pane showing the output of the query as a single row with the value '2021-10-03'. The bottom-left pane shows 'Administration Schemas' and 'Information' with the text 'No object selected'.

curdate()
2021-10-03

# Завантаження бази даних в MySQL



Run SQL Script

Operation completed successfully

Output:

```
Preparing...
Importing employees.sql...
Finished executing script
INFO
CREATING DATABASE STRUCTURE
INFO
storage engine: InnoDB
INFO
LOADING departments
INFO
LOADING employees
INFO
LOADING dept_emp
INFO
LOADING dept_manager
INFO
LOADING titles
INFO
LOADING salaries
data_load_time_diff
00:01:14
Operation completed successfully
```

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- sys

Query 1

```
1 select curdate();
```

Result Grid | Filter Rows: | Exports: | Wrap Cell Contents: |

curdate()



# Основні правила написання скриптів в MySQL

The screenshot shows the MySQL Workbench interface with a SQL script in the Query Editor. The script is for creating a database named 'employees' and several tables. Red circles highlight the following lines of code:

```
17 -- California, 94105, USA.  
18 --  
19 -- DISCLAIMER  
20 -- To the best of our knowledge, this data is fabricated, and  
21 -- it does not correspond to real people.  
22 -- Any similarity to existing people is purely coincidental.  
23 --  
24  
25 DROP DATABASE IF EXISTS employees;  
26 CREATE DATABASE IF NOT EXISTS employees;  
27 USE employees;  
28  
29 SELECT 'CREATING DATABASE STRUCTURE' as 'INFO';  
30  
31 DROP TABLE IF EXISTS dept_emp,  
32 dept_manager,  
33 titles,  
34 salaries,  
35 employees,  
36 departments;  
37
```



## Завдання до теми 1

1. Інсталювати реляційну систему управління базами даних MySQL Workbench 8.0 (безкоштовна версія).
2. Підключитися до бази даних Employees.
3. Ознайомитися з елементами бази даних: таблицями, їх вмістом, зв'язками між таблицями.
4. Побудувати ER-модель і реляційну схему бази даних.
5. Визначити атрибути primary key, foreign key, типи даних у таблицях.

## Список рекомендованих джерел до теми 1

1. Reed M. SQL: 3 books in 1 – The Ultimate Beginner, Intermediate & Expert Guides To Master SQL Programming Quickly with Practical Exercises, London, England : Amazon, 2022, 348 p.
2. Korotkevitch D. SQL Server Advanced Troubleshooting and Performance Tuning. Best Practices and Techniques, USA : O'Reilly, 2022, 500 p.
3. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань, книга 2 : системи управління базами даних та знань: навч. посіб. Львів : Магнолія, 2021, 584 с.



# Тема 2

## Створення запитів мовою SQL. Оператор SELECT

### SELECT Syntax

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

Here, *column1*, *column2*, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

```
SELECT * FROM table_name;
```





MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees employees SQL File 5\* x

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
1 SELECT curdate();
```

Result Grid

curdate()
2023-09-10



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees employees SQL File 5 x

SCHMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

```
1 SELECT curdate();
2
3 SELECT *
4 FROM employees.employees;
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10002	1964-06-02	Bezael	Simmel	F	1985-11-21
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10004	1954-05-01	Christian	Koblick	M	1986-12-01
10005	1955-01-21	Kyoichi	Malniak	M	1989-09-12
10006	1953-04-20	Anneke	Preusig	F	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15
10009	1952-04-19	Sumant	Peac	F	1985-02-18
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24
10011	1953-11-07	Mary	Sluis	F	1990-01-22
10012	1960-10-04	Patricio	Bridgland	M	1997-12-18

employees 4 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

Query 1 employees employees SQL File 5 x

1 SELECT curdate();

2

3 • SELECT \*

4 FROM employees.employees;

5

6 • SELECT first\_name, last\_name

7 FROM employees.employees;

Result Grid

first_name	last_name
Georgi	Facello
Bezael	Simmel
Parto	Bamford
Chirstian	Koblick
Kyoichi	Malniak
Anneke	Preusig
Tzvetan	Zielinski
Saniya	Kaloufi
Sumant	Peac
Duangkaew	Piveteau
Mary	Sluis
Patricio	Bridgland

employees 5 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees employees SQL File 5\*

**SCHMAS**

Filter objects

**employees**

- departments
- dept\_emp
- dept\_manager
- employees
- salaries
- titles**
  - Columns
  - Indexes
  - Foreign Keys
  - Triggers
- Views
- Stored Procedures
- Functions

my1 sys

Administration Schemas

Information

**Table: titles**

**Columns:**

- emp\_no** int PK
- title** varchar(50) PK
- from\_date** date PK

```
7 FROM employees.employees;
8
9 • SELECT first_name, last_name, gender AS employee_gender
10 FROM employees.employees;
11
12 • SELECT title
13 FROM titles;
```

Result Grid

title
Senior Engineer
Staff
Senior Engineer
Engineer
Senior Engineer
Senior Staff
Staff
Senior Engineer
Senior Staff
Staff
Assistant Engineer
Assistant Engineer

titles 19 x

Output

Активаци

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees employees SQL File 5'

**SCHEMAS**

Filter objects

employees

Tables

- departments
- dept\_emp
- dept\_manager
- employees
- salaries
- titles

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

my1

sys

Administration Schemas

Information

Table: titles

Columns:

- emp\_no int PK
- title varchar(50) PK
- from\_date date PK
- to\_date date

Query 1

```

12 SELECT title
13 FROM titles;
14
15 SELECT distinct title
16 FROM titles;

```

Result Grid

title
Senior Engineer
Staff
Engineer
Senior Staff
Assistant Engineer
Technique Leader
Manager

titles 20 x

Output

Action Output

#	Time	Action	Message
✓ 20	13:00:46	SELECT * FROM employees employees	300024 row(s) returned
✓ 21	13:00:48	SELECT first_name, last_name FROM employees employees	300024 row(s) returned
✓ 22	13:00:50	SELECT first_name, last_name, gender AS employee_gender FROM employees.empl...	300024 row(s) returned
✗ 23	13:00:51	SELECT titles FROM titles	Error Code: 1054. Unknown column 1
✗ 24	13:00:58	SELECT titles FROM titles	Error Code: 1054. Unknown column 1
✓ 25	13:01:15	SELECT title FROM titles	443308 row(s) returned
✓ 26	13:05:43	SELECT distinct title FROM titles	7 row(s) returned



# Оператор WHERE

## WHERE Syntax

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



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees employees SQL File 5\* x

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions
- my1
- sys

Query 1

```
19
20
21 • SELECT *
22 FROM employees.employees
23 WHERE employees.gender='M';
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10004	1954-05-01	Christian	Koblick	M	1986-12-01
10005	1955-01-21	Kyoichi	Malniak	M	1989-09-12
10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15
10012	1960-10-04	Patricio	Bridgland	M	1992-12-18
10013	1963-06-07	Eberhardt	Terkki	M	1985-10-20
10014	1956-02-12	Berni	Genin	M	1987-03-11
10015	1959-08-19	Guoxiang	Nooteboom	M	1987-07-02
10016	1961-05-02	Kazuhito	Cappelletti	M	1995-01-27

employees 2 x

Output

Action Output

#	Time	Action	Message
1	11:54:37	SELECT * FROM employees.employees WHERE employees.gender='M' LIMIT 0, 500	500 row(s) returned

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees employees SQL File 5\* x

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

Query 1

```
25
26 • SELECT last_name
27 FROM employees.employees
28 WHERE employees.first_name='Parto';
29
```

Result Grid

last_name
Bamford
Wrigley
Hitomi
Heinisuo
Mandell
Pietracaprina
Godskesen
Bertziss
Speek
Bernini

employees 3 x

Output

Action Output

#	Time	Action	Message
1	11:54:37	SELECT * FROM employees.employees WHERE employees.gender='M' LIMIT 0, 500	500 row(s) returned





test\_111 x

File Edit View Query Database Server Tools Scripting Help



Navigator

SCHEMAS

Filter objects

employees

Tables

- departments
- dept\_emp
- dept\_manager
- employees
- salaries
- titles

Views

Stored Procedures

Functions

my1

sys

Query 1 employees employees SQL File 5\* x

29

30

31 SELECT \*

32 FROM employees.employees

33 WHERE employees.emp\_no=10003;

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	emp_no	birth_date	first_name	last_name	gender	hire_date
▶	10003	1959-12-03	Parto	Bamford	M	1986-08-28
•	NULL	NULL	NULL	NULL	NULL	NULL

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1 employees employees SQL File 5 x

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions
- my1
- sys

34

35

36 `SELECT *`

37 `FROM employees.employees`

38 `WHERE employees.hire_date != '1986-08-28';`

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26

Administration Schemas Information

36 `SELECT *`

37 `FROM employees.employees`

38 `WHERE employees.hire_date <> '1986-08-28';`

Result Grid

emp_no	birth_date	first_name	last_name	g
10001	1953-09-02	Georgi	Facello	M
10002	1964-06-02	Bezalel	Simmel	F
10004	1954-05-01	Christian	Koblick	M
10005	1955-01-21	Kyoichi	Malniak	M
10006	1953-04-20	Anneke	Preusig	F
10007	1957-05-23	Tzvetan	Zielinski	F

# УМОВИ AND, OR, NOT

## AND Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2 AND condition3 ...;
```

## OR Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 OR condition2 OR condition3 ...;
```

## NOT Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE NOT condition;
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

Query 1 employees employees SQL File 5\* x

Limit to 500 rows

```
41
42 • SELECT *
43 FROM employees.employees
44 WHERE employees.first_name='Parto' AND employees.gender='F';
45
46
47
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
▶ 10387	1952-11-03	Parto	Wrigley	F	1987-02-19
11052	1960-03-16	Parto	Hitomi	F	1988-11-11
13504	1954-01-09	Parto	Mandell	F	1986-12-21
17531	1958-07-22	Parto	Bertziss	F	1994-10-11
29026	1953-03-18	Parto	Byoun	F	1985-06-12
29120	1954-06-02	Parto	Bergere	F	1987-07-15
34238	1958-09-11	Parto	Asser	F	1989-12-25
40021	1955-03-08	Parto	Delgrande	F	1988-02-02

employees 2 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees employees SQL File 5\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
45
46 SELECT *
47 FROM employees
48 WHERE employees.first_name='Parto'
49         AND employees.gender='F'
50         AND YEAR(hire_date)=1987;
51
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
10387	1952-11-03	Parto	Wrigley	F	1987-02-19
29120	1954-06-02	Parto	Bergere	F	1987-07-15
67380	1962-02-26	Parto	Curless	F	1987-03-14
230846	1955-02-15	Parto	Pereigut	F	1987-04-28
240838	1957-10-10	Parto	Wossner	F	1987-08-07
261225	1953-05-16	Parto	Honiden	F	1987-04-04
272593	1957-05-16	Parto	Sidou	F	1987-09-08
450679	1957-04-23	Parto	Plotkin	F	1987-12-20

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas Information

Query 1 employees employees SQL File 5\* x

```
51
52
53 SELECT *
54 FROM employees.employees
55 WHERE first_name='Parto' OR last_name='Plotkin';
56
57
58
```

Result Grid

	emp_no	birth_date	first_name	last_name	gender	hire_date
▶	10003	1959-12-03	Parto	Bamford	M	1986-08-28
	10387	1952-11-03	Parto	Wrigley	F	1987-02-19
	11052	1960-03-16	Parto	Hitomi	F	1988-11-11
	11399	1958-04-02	Luisa	Plotkin	M	1985-02-09
	11575	1955-02-22	Godehard	Plotkin	F	1991-12-19
	12203	1964-11-19	Parto	Heinisuo	M	1987-07-21
	13504	1954-01-09	Parto	Mandell	F	1986-12-21
	13815	1953-04-19	Guadalupe	Plotkin	M	1985-09-09

employees 3 employees 4 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Query 1 employees employees SQL File 5\* x

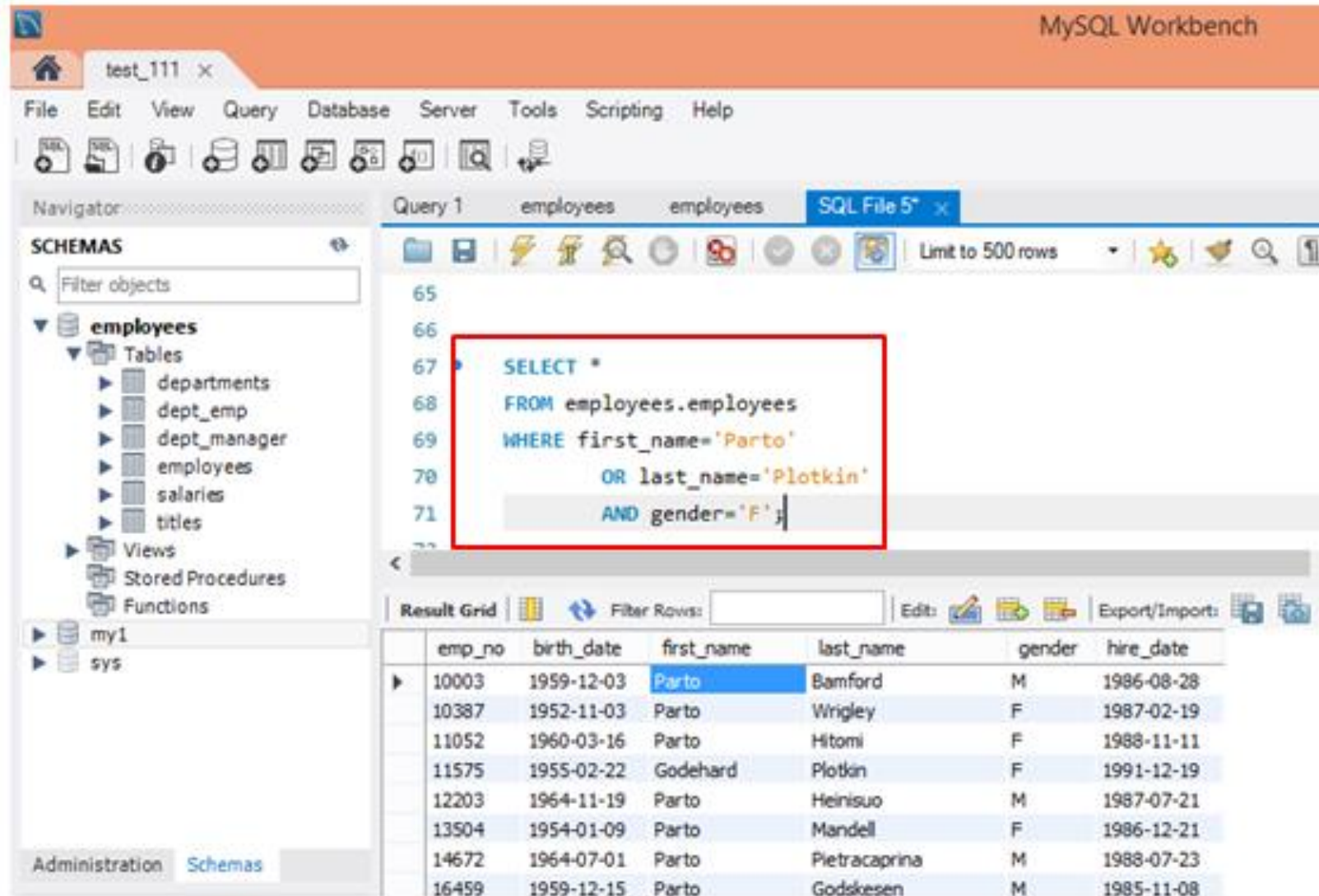
Limit to 500 rows

```
58
59 SELECT *
60 FROM employees.employees
61 WHERE first_name='Parto'
62        OR last_name='Plotkin'
63        OR last_name='Mandell';
64
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
10003	1959-12-03	Parto	Ramford	M	1986-08-28
10096	1954-09-16	Jayson	Mandell	M	1990-01-14
10237	1952-09-22	Yannis	Mandell	F	1989-08-11
10387	1952-11-03	Parto	Wrigley	F	1987-02-19
11052	1960-03-16	Parto	Hitomi	F	1988-11-11
11399	1958-04-02	Luisa	Plotkin	M	1985-02-09
11575	1955-02-22	Godehard	Plotkin	F	1991-12-19
12203	1964-11-19	Parto	Heinisuo	M	1987-07-21

# Пріоритет логічного оператора



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employees' database selected. The main window shows a SQL query in the 'Query 1' editor:

```
65  
66  
67 SELECT *  
68 FROM employees.employees  
69 WHERE first_name='Parto'  
70        OR last_name='Plotkin'  
71        AND gender='F';
```

The query results are displayed in the 'Result Grid' below the editor:

emp_no	birth_date	first_name	last_name	gender	hire_date
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10387	1952-11-03	Parto	Wrigley	F	1987-02-19
11052	1960-03-16	Parto	Hitomi	F	1988-11-11
11575	1955-02-22	Godehard	Plotkin	F	1991-12-19
12203	1964-11-19	Parto	Heinisuo	M	1987-07-21
13504	1954-01-09	Parto	Mandell	F	1986-12-21
14672	1964-07-01	Parto	Pietracaprina	M	1988-07-23
16459	1959-12-15	Parto	Godskesen	M	1985-11-08



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

Query 1 employees employees SQL File 5\* x

Limit to 500 rows

```
65
66
67 • SELECT *
68 FROM employees.employees
69 WHERE first_name='Parto'
70 OR (last_name='Plotkin'
71 AND gender='F')
```

Result Grid Filter Rows: Edit: Export/Import:

emp_no	birth_date	first_name	last_name	gender	hire_date
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10387	1952-11-03	Parto	Wrigley	F	1987-02-19
11052	1960-03-16	Parto	Hitomi	F	1988-11-11
▶ 11575	1955-02-22	Godehard	Plotkin	F	1991-12-19
12203	1964-11-19	Parto	Heinisuo	M	1987-07-21
13504	1954-01-09	Parto	Mandell	F	1986-12-21
14672	1964-07-01	Parto	Pietracaprina	M	1988-07-23
16459	1959-12-15	Parto	Godskesen	M	1985-11-08

employees 3 employees 7 x



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees employees SQL File 5\* x

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
66
67 • SELECT *
68 FROM employees.employees
69 WHERE (first_name='Parto'
70        OR last_name='Plotkin')
71        AND gender='F';
72
73
```

Result Grid | Filter Rows: | Edit: | Export/Import:

emp_no	birth_date	first_name	last_name	gender	hire_date
10387	1952-11-03	Parto	Wrigley	F	1987-02-19
11052	1960-03-16	Parto	Hitomi	F	1988-11-11
11575	1955-02-22	Godehard	Plotkin	F	1991-12-19
13504	1954-01-09	Parto	Mandell	F	1986-12-21
17358	1964-09-19	Masami	Plotkin	F	1995-12-13
17531	1958-07-22	Parto	Bertziss	F	1994-10-11
24595	1957-09-03	Ramalingam	Plotkin	F	1997-06-28
28544	1963-03-11	Doohun	Plotkin	F	1990-06-19

employees 3 employees 8 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Query 1 employees employees SQL File 5\* x

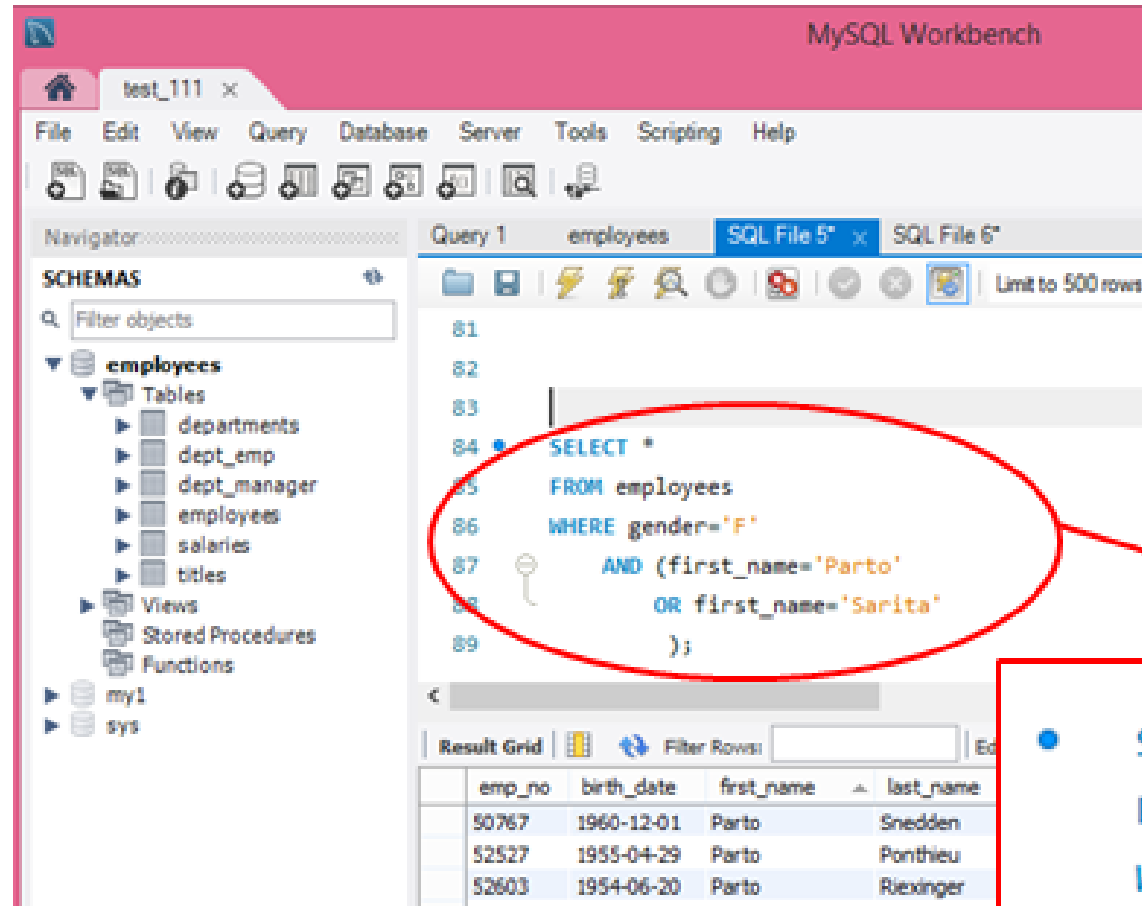
```
73  
74  
75 SELECT *  
76 FROM employees.employees  
77 WHERE (first_name='Parto'  
78 OR last_name='Plotkin')  
79 AND (gender='F'  
80 AND YEAR(hire_date)=1987);
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
10387	1952-11-03	Parto	Wrigley	F	1987-02-19
29120	1954-06-02	Parto	Bergere	F	1987-07-15
67380	1962-02-26	Parto	Curless	F	1987-03-14
70023	1954-07-23	Bodh	Plotkin	F	1987-07-02
230846	1955-02-15	Parto	Perelgut	F	1987-04-28
240838	1957-10-10	Parto	Wossner	F	1987-08-07

employees 1 x

# Оператор IN / NOT IN



MySQL Workbench

Navigator: test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Filter objects

SCHEMAS

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Query 1 employees SQL File 5\* x SQL File 6\*

Limit to 500 rows

```
81  
82  
83  
84 SELECT *  
85 FROM employees  
86 WHERE gender='F'  
87 AND (first_name='Parto'  
88 OR first_name='Sarita'  
89 );
```

Result Grid Filter Rows: | Ed

emp_no	birth_date	first_name	last_name
80767	1960-12-01	Parto	Snedden
52527	1955-04-29	Parto	Ponthieu
52603	1954-06-20	Parto	Rixinger

```
• SELECT *  
FROM employees  
WHERE gender='F'  
AND first_name IN ('Parto', 'Sarita');
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas Information

Query 1 employees SQL File 5\* SQL File 6\*

Limit to 500 rows

```
99
100 SELECT *
101 FROM employees
102 WHERE gender='F'
103     AND first_name IN ('Parto', 'Sarita')
104     AND last_name NOT IN('Wrigley', 'Hitomi');
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
12307	1964-09-15	Sarita	Schlumberger	F	1985-02-23
13504	1954-01-09	Parto	Mandell	F	1986-12-21
13983	1961-08-10	Sarita	Siepmann	F	1985-03-22
15021	1958-05-13	Sarita	Czaja	F	1988-12-21
17531	1958-07-22	Parto	Bertziss	F	1994-10-11
19238	1964-06-06	Sarita	Birge	F	1986-05-11
20102	1961-05-01	Sarita	Morton	F	1991-05-14
23409	1956-03-27	Sarita	Oxman	F	1985-04-19
26435	1961-11-05	Sarita	Konrowski	F	1986-03-20

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas Information

Query 1 employees SQL File 5 x SQL File 6 x

```
107 SELECT *
108 FROM employees
109 WHERE gender='F'
110     AND first_name IN ('Parto', 'Sarita')
111     AND last_name NOT IN('Wrigley', 'Hitomi')
112     AND YEAR(hire_date) IN (1986, 1985);
113
114
```

Limit to 500 rows

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
12307	1964-09-15	Sarita	Schlumberger	F	1985-02-23
13504	1954-01-09	Parto	Mandell	F	1986-12-21
13983	1961-08-10	Sarita	Siepmann	F	1985-03-22
19238	1964-06-06	Sarita	Birge	F	1986-05-11
23409	1956-03-27	Sarita	Oxman	F	1985-04-19
26435	1961-11-05	Sarita	Koprowski	F	1986-03-20
29026	1953-03-18	Parto	Byoun	F	1985-06-12
29194	1952-05-02	Sarita	Peris	F	1986-06-02
50767	1960-12-01	Parto	Charlton	F	1986-07-14

employees 6 x

# Оператор LIKE / NOT LIKE. Підстановчі символи

## The MySQL LIKE Operator

The `LIKE` operator is used in a `WHERE` clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the `LIKE` operator:

- The percent sign (%) represents zero, one, or multiple characters
- The underscore sign (\_) represents one, single character

The percent sign and the underscore can also be used in combinations!

### LIKE Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE columnN LIKE pattern;
```



<b>LIKE Operator</b>	<b>Description</b>
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"





MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas Information

```
113  
114  
115  
116 • SELECT *  
117 FROM employees  
118 WHERE gender='M'  
119 AND emp_no LIKE '125%';  
120
```

Result Grid Filter Rows: Edit: Export/Import:

emp_no	birth_date	first_name	last_name	gender	hire_date
12500	1956-12-21	Jackson	Luke	M	1997-05-11
12503	1961-11-30	Boguslaw	Matzen	M	1991-03-30
12505	1952-08-24	Barton	Goldhammer	M	1991-04-20
12507	1957-10-12	Manton	Plump	M	1988-07-04
12509	1954-05-20	Rutger	Krzyzanowski	M	1991-09-26
12511	1964-11-01	Frederique	Lueh	M	1991-02-26
12514	1952-11-14	Tomofumi	Melter	M	1985-12-03
12519	1963-08-04	Suebskul	Ghazalie	M	1985-05-13
12520	1954-01-21	Guenther	Rosart	M	1980-09-14

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* SQL File 6\*

Limit to 500 rows

121  
122  
123 • `SELECT *`  
124 `FROM employees`  
125 `WHERE gender='M'`  
126 `AND emp_no LIKE '12%'`  
127 `AND birth_date LIKE '%01';`  
128

Result Grid Filter Rows: Edit: Export/Import:

emp_no	birth_date	first_name	last_name	gender	hire_date
12106	1956-10-01	Zissis	Cangellaris	M	1995-10-24
12112	1953-07-01	Satyanarayana	Avouris	M	1986-05-24
12124	1964-08-01	Jingling	Schaaf	M	1986-06-29
12218	1954-08-01	Yaghout	Strooper	M	1985-02-18
12244	1960-02-01	Annemarie	Harlee	M	1987-01-28
12274	1955-02-01	Garnik	Papadias	M	1988-04-01
12283	1959-12-01	Slavian	Rosin	M	1988-02-18
12333	1964-05-01	Gino	Peck	M	1990-09-20



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
129
130 SELECT *
131 FROM employees
132 WHERE gender='M'
133     AND emp_no LIKE '12%'
134     AND birth_date LIKE '%01'
135     AND last_name LIKE '%a%';
136
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wr

emp_no	birth_date	first_name	last_name	gender	hire_date
12106	1956-10-01	Zisis	Cangellaris	M	1995-10-24
12112	1953-07-01	Satyanarayana	Avouris	M	1986-05-24
12124	1964-08-01	Jingling	Schaaf	M	1986-06-29
12244	1960-02-01	Annemarie	Harlee	M	1987-01-28
12274	1955-02-01	Garnik	Papadias	M	1988-04-01
12348	1961-02-01	Krister	Rahimi	M	1985-11-08
12437	1957-05-01	Mohammed	Cronan	M	1987-08-26
12664	1952-05-01	Zhonghua	Parveen	M	1987-08-19
12600	1957-06-01	Trachin	Seta	M	1985-09-24

employees 10 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
145
146 SELECT *
147 FROM employees
148 WHERE gender='M'
149 AND emp_no LIKE '1200_';
150
151
152
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
12000	1957-11-09	Claudi	Angelov	M	1989-01-12
12002	1959-09-17	Seongbae	Kavraki	M	1994-10-14
12003	1961-04-15	Luise	McFarlan	M	1992-11-20
12005	1958-07-27	Shir	DuCasse	M	1993-03-30
12006	1963-08-09	Remco	Erol	M	1989-12-27
12007	1952-06-18	Marko	Dehkordi	M	1989-05-17
12009	1963-09-26	Sariel	Stifter	M	1994-11-28
NULL	NULL	NULL	NULL	NULL	NULL

Administration Schemas Information

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

```
157  
158  
159  
160 SELECT *  
161 FROM employees  
162 WHERE gender='M'  
163 AND emp_no LIKE '_2_05';  
164
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
12005	1958-07-27	Shir	DuCasse	M	1993-03-30
12205	1955-08-24	Sushant	Rossi	M	1986-04-14
12305	1959-02-24	Marek	Ferriere	M	1985-06-06
12405	1964-04-06	Zhanqiu	Kambil	M	1989-07-16
12505	1952-08-24	Barton	Goldhammer	M	1991-04-20
12605	1964-10-26	Petter	Feldmann	M	1987-12-29
12805	1959-03-21	Naraig	Itschner	M	1988-09-11



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Query 1 employees SQL File 5\* x SQL File 6\*

Limit to 500 rows

```
165
166
167 SELECT *
168 FROM employees
169 WHERE gender='M'
170 AND (emp_no LIKE '_2_05'
171 OR last_name LIKE 'r_s%');
172
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
10046	1960-07-23	Lucien	Rosenbaum	M	1992-06-20
10081	1960-12-17	Zhongwei	Rosen	M	1986-10-30
10115	1964-12-25	Chikara	Rissland	M	1986-01-23
10167	1958-05-23	Duangkaew	Rassart	M	1992-04-04
10251	1957-04-25	Alair	Rosenbaum	M	1992-06-25



## The MySQL BETWEEN Operator

The **BETWEEN** operator selects values within a given range. The values can be numbers, text, or dates.

The **BETWEEN** operator is inclusive: begin and end values are included.

### BETWEEN Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

Tables

- departments
- dept\_emp
- dept\_manager
- employees
- salaries
- titles

Views

Stored Procedures

Functions

my1

sys

Query 1 employees SQL File 5\* SQL File 6\*

Limit to 500 rows

```
173
174
175
176 • SELECT *
177 FROM employees
178 WHERE gender='M'
179 AND emp_no BETWEEN 10200 AND 12000;
180
```

Result Grid

Filter Rows: Edit: Export/Import:

emp_no	birth_date	first_name	last_name	gender	hire_date
10200	1961-12-31	Vidya	Awdeh	M	1985-10-16
10202	1956-01-05	Greger	Lichtner	M	1991-10-06
10203	1955-07-27	Steen	Escriba	M	1989-04-06
10208	1960-01-02	Xiping	Klerer	M	1991-12-23
10210	1958-01-24	Yuping	Alpin	M	1994-05-10
10212	1959-05-09	Divier	Esteva	M	1990-07-11



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* SQL File 6\*

Limit to 500 rows

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
185 SELECT *
186 FROM employees
187 WHERE gender='M'
188     AND emp_no BETWEEN 10200 AND 12000
189     AND hire_date BETWEEN '1986-10-01' AND '1991-01-01';
190
191
192
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date
11788	1962-04-11	Georgy	Gopalakrishnan	M	1986-10-03
11675	1964-06-10	Slavian	Pettit	M	1986-10-07
10867	1952-10-01	Billur	Gyorkos	M	1986-10-08
11983	1953-01-04	Jaana	Hedayat	M	1986-10-11
11334	1961-08-13	Yurij	Vural	M	1986-10-12
11100	1955-10-14	Berna	Cochrane	M	1986-10-13
11588	1952-07-20	Kaijung	Panangaden	M	1986-10-13

Administration Schemas

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

```
195
196 SELECT *
197 FROM employees
198 WHERE gender='M'
199     AND ((emp_no BETWEEN 10200 AND 12000 )
200           OR emp_no IN (10001, 10002))
201     AND hire_date BETWEEN '1986-10-01' AND '1991-01-01';
202
```

Result Grid | Filter Rows: | Edit: | Export/Import:

emp_no	birth_date	first_name	last_name	gender	hire_date
10203	1955-07-27	Steen	Escriba	M	1989-04-06
10212	1959-05-09	Divier	Esteva	M	1990-07-11
10214	1962-04-14	Tadahiko	Ciolek	M	1988-02-29
10215	1954-04-02	Xiaobin	Duclos	M	1987-10-19
10216	1958-07-16	Amstein	Ghemri	M	1987-10-30
10221	1957-01-03	Yucai	Granlund	M	1988-06-08

# YMOBa IS NOT NULL / IS NULL

## IS NULL Syntax

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

## IS NOT NULL Syntax

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

Query 1 employees SQL File 5\* x SQL File 6\*

222

223

224 • **SELECT \***

225 **FROM** employees.employees **AS** em

226 **LEFT JOIN** employees.dept\_manager **AS** dm

227 **ON** em.emp\_no=dm.emp\_no;

228

Result Grid

Filter Rows: Export: Wrap Cell Contents: Fetch rows:

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
109995	1960-02-11	Constantijn	Anido	F	1986-06-11	NULL	NULL	NULL	NULL
109996	1952-09-24	Dines	Gelosh	F	1987-06-15	NULL	NULL	NULL	NULL
109997	1955-07-07	Miquel	Borovoy	M	1988-12-28	NULL	NULL	NULL	NULL
109998	1953-09-21	Mizuhito	Heping	M	1986-11-11	NULL	NULL	NULL	NULL
109999	1955-10-28	Janche	Coombs	M	1987-02-22	NULL	NULL	NULL	NULL
110000	1956-07-28	Supot	Herath	M	1987-07-07	NULL	NULL	NULL	NULL
110022	1956-09-12	Margareta	Markovitch	M	1985-01-01	110022	d001	1985-01-01	1991-1...
110039	1963-06-21	Vishwari	Minakawa	M	1986-04-12	110039	d001	1991-10-01	9999-0...
110085	1959-10-28	Ebru	Alpin	M	1985-01-01	110085	d002	1985-01-01	1989-1...
110114	1957-03-28	Isamu	Legleitner	F	1985-01-14	110114	d002	1989-12-17	9999-0...
110183	1953-06-24	Shirish	Ossenbruggen	F	1985-01-01	110183	d003	1985-01-01	1992-0...



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions
- my1
- sys

Administration Schemas Information

```

238 SELECT *
239 FROM employees.employees AS em
240 LEFT JOIN employees.dept_manager AS dm
241     ON em.emp_no=dm.emp_no
242 WHERE dept_no IS NOT NULL;
243

```

Result Grid

	emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
▶	110022	1956-09-12	Margareta	Markovitch	M	1985-01-01	110022	d001	1985-01-01	1991-10-01
	110039	1963-06-21	Vishwani	Minakawa	M	1986-04-12	110039	d001	1991-10-01	9999-01-01
	110085	1959-10-28	Ebru	Alpin	M	1985-01-01	110085	d002	1985-01-01	1989-12-17
	110114	1957-03-28	Isamu	Legleitner	F	1985-01-14	110114	d002	1989-12-17	9999-01-01
	110183	1953-06-24	Shirish	Ossenbruggen	F	1985-01-01	110183	d003	1985-01-01	1992-03-21
	110228	1958-12-02	Karsten	Sigstam	F	1985-08-04	110228	d003	1992-03-21	9999-01-01
	110303	1956-06-08	Krassimir	Wegerle	F	1985-01-01	110303	d004	1985-01-01	1988-09-09
	110344	1961-09-07	Rosine	Cools	F	1985-11-22	110344	d004	1988-09-09	1992-08-02
	110386	1953-10-04	Shem	Keras	M	1988-10-14	110386	d004	1992-08-02	1996-08-30
	110420	1963-07-27	Oscar	Ghazale	M	1992-02-05	110420	d004	1996-08-30	9999-01-01
	110511	1957-07-08	DeForest	Hagimont	M	1985-01-01	110511	d005	1985-01-01	1992-04-25



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects:

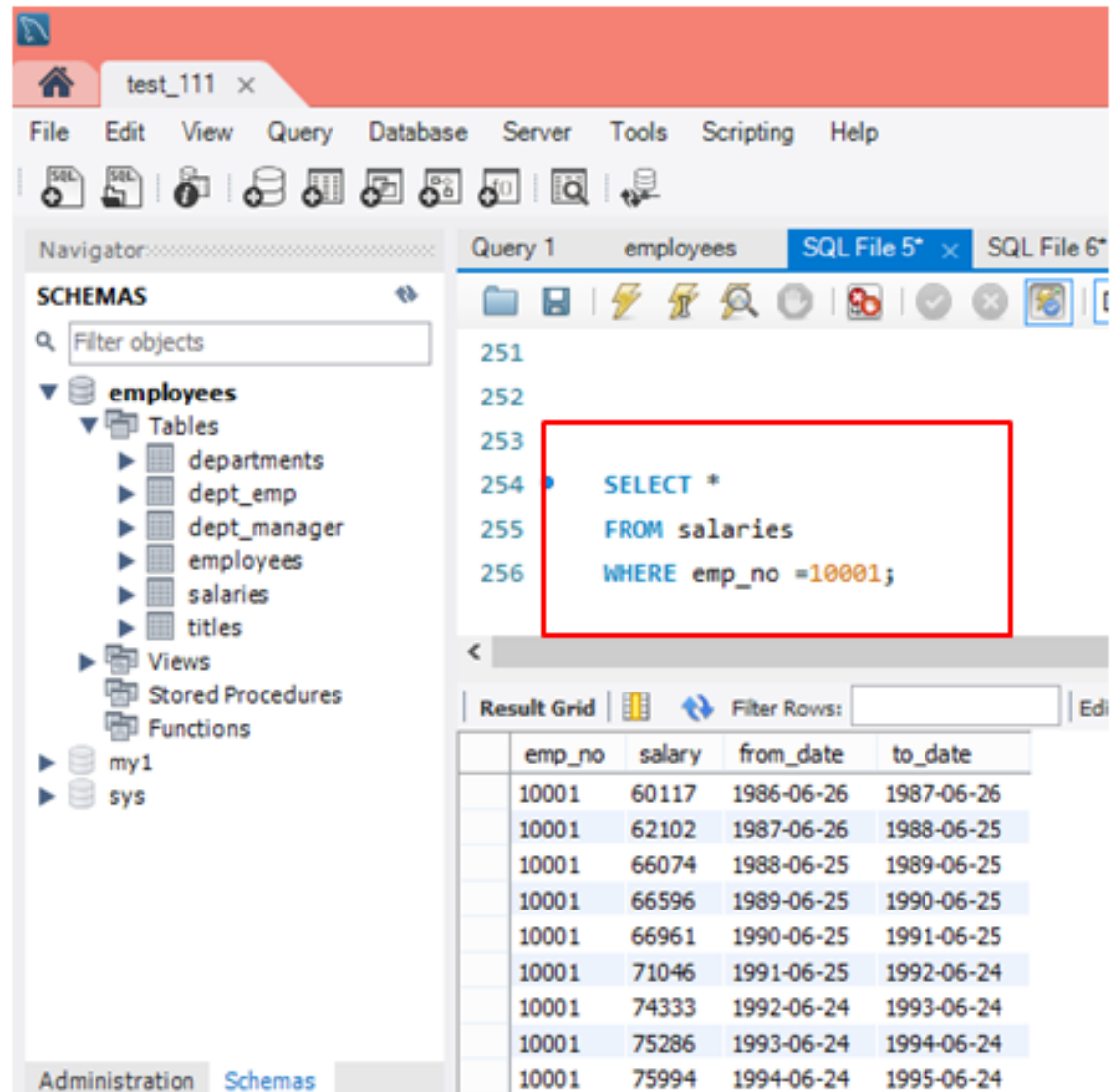
- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

```
245  
246 • SELECT *  
247 FROM employees.employees AS em  
248 LEFT JOIN employees.dept_manager AS dm  
249 ON em.emp_no=dm.emp_no  
250 WHERE dept_no IS NULL;  
251
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26	NULL	NULL	NULL	NULL
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	NULL	NULL	NULL	NULL
10003	1959-12-03	Parto	Bamford	M	1986-08-28	NULL	NULL	NULL	NULL
10004	1954-05-01	Christian	Koblick	M	1986-12-01	NULL	NULL	NULL	NULL
10005	1955-01-21	Kyoichi	Malniak	M	1989-09-12	NULL	NULL	NULL	NULL
10006	1953-04-20	Anneke	Preusig	F	1989-06-02	NULL	NULL	NULL	NULL
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10	NULL	NULL	NULL	NULL
10008	1958-02-19	Saniya	Kaloufi	M	1994-09-15	NULL	NULL	NULL	NULL
10009	1952-04-19	Sumant	Peac	F	1985-02-18	NULL	NULL	NULL	NULL
10010	1963-06-01	Duanokaew	Piveteau	F	1989-08-24	NULL	NULL	NULL	NULL

# Оператори порівняння



The screenshot shows the SQL Developer interface. The Navigator pane on the left displays the 'employees' schema with tables: departments, dept\_emp, dept\_manager, employees, salaries, and titles. The main query window shows the following SQL code:

```
251  
252  
253  
254 SELECT *  
255 FROM salaries  
256 WHERE emp_no =10001;
```

The result grid below the query shows the following data:

emp_no	salary	from_date	to_date
10001	60117	1986-06-26	1987-06-26
10001	62102	1987-06-26	1988-06-25
10001	66074	1988-06-25	1989-06-25
10001	66596	1989-06-25	1990-06-25
10001	66961	1990-06-25	1991-06-25
10001	71046	1991-06-25	1992-06-24
10001	74333	1992-06-24	1993-06-24
10001	75286	1993-06-24	1994-06-24
10001	75994	1994-06-24	1995-06-24

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with the 'employees' database expanded to show tables like 'salaries'. The right pane shows a query window with the following SQL code:

```
257  
258  
259 • SELECT *  
260 FROM salaries  
261 WHERE emp_no =10001  
262 AND salary>71046;
```

Below the query, the 'Result Grid' shows the following data:

emp_no	salary	from_date	to_date
10001	74333	1992-06-24	1993-06-24
10001	75286	1993-06-24	1994-06-24
10001	75994	1994-06-24	1995-06-24
10001	76884	1995-06-24	1996-06-23
10001	80013	1996-06-23	1997-06-23
10001	81025	1997-06-23	1998-06-23
10001	81097	1998-06-23	1999-06-23
10001	84917	1999-06-23	2000-06-22





The screenshot shows a SQL IDE window titled 'test\_111'. The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The Navigator pane on the left shows a tree view of schemas, with 'employees' expanded to show tables like departments, dept\_emp, dept\_manager, employees, salaries, and titles. The main editor displays a SQL query in 'Query 1' for the 'employees' table. The query is: `SELECT * FROM salaries WHERE emp_no = 10001 AND salary < 71046;`. Below the query, the 'Result Grid' shows the output of the query, which is a table with columns 'emp\_no', 'salary', 'from\_date', and 'to\_date'. The results are: 

emp_no	salary	from_date	to_date
10001	60117	1986-06-26	1987-06-26
10001	62102	1987-06-26	1988-06-25
10001	66074	1988-06-25	1989-06-25
10001	66596	1989-06-25	1990-06-25
10001	66961	1990-06-25	1991-06-25

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with the 'employees' schema expanded, showing tables like 'departments', 'dept\_emp', 'dept\_manager', 'employees', 'salaries', and 'titles'. The right pane shows a query window with the following SQL query:

```
257  
258  
259 • SELECT *  
260 FROM salaries  
261 WHERE emp_no =10001  
262 AND salary<>71046;
```

Below the query, the 'Result Grid' displays the following data:

emp_no	salary	from_date	to_date
10001	60117	1986-06-26	1987-06-26
10001	62102	1987-06-26	1988-06-25
10001	66074	1988-06-25	1989-06-25
10001	66596	1989-06-25	1990-06-25
10001	66961	1990-06-25	1991-06-25
10001	74333	1992-06-24	1993-06-24
10001	75286	1993-06-24	1994-06-24

## Завдання до теми 2

1. Виберіть з бази даних і сформууйте такі списки:

- працівників чоловічої статі ‘Mark’;
- працівників жіночої статі, які прийняті на роботу 29.11.1995;
- працівників чоловічої статі, які прийняті на роботу 19.06.1996;
- працівників жіночої статі, що носять імена Kellie та Beshir;
- працівників жіночої статі, які прийняті на роботу 29.11.1995, а також працівників чоловічої статі, які прийняті на роботу 19.06.1996;
- працівників жіночої статі, які прийняті на роботу в такі дати: 07.12.1997, 29.11.1995, 11.11.1994, 27.04.1995, 03.01.1997.

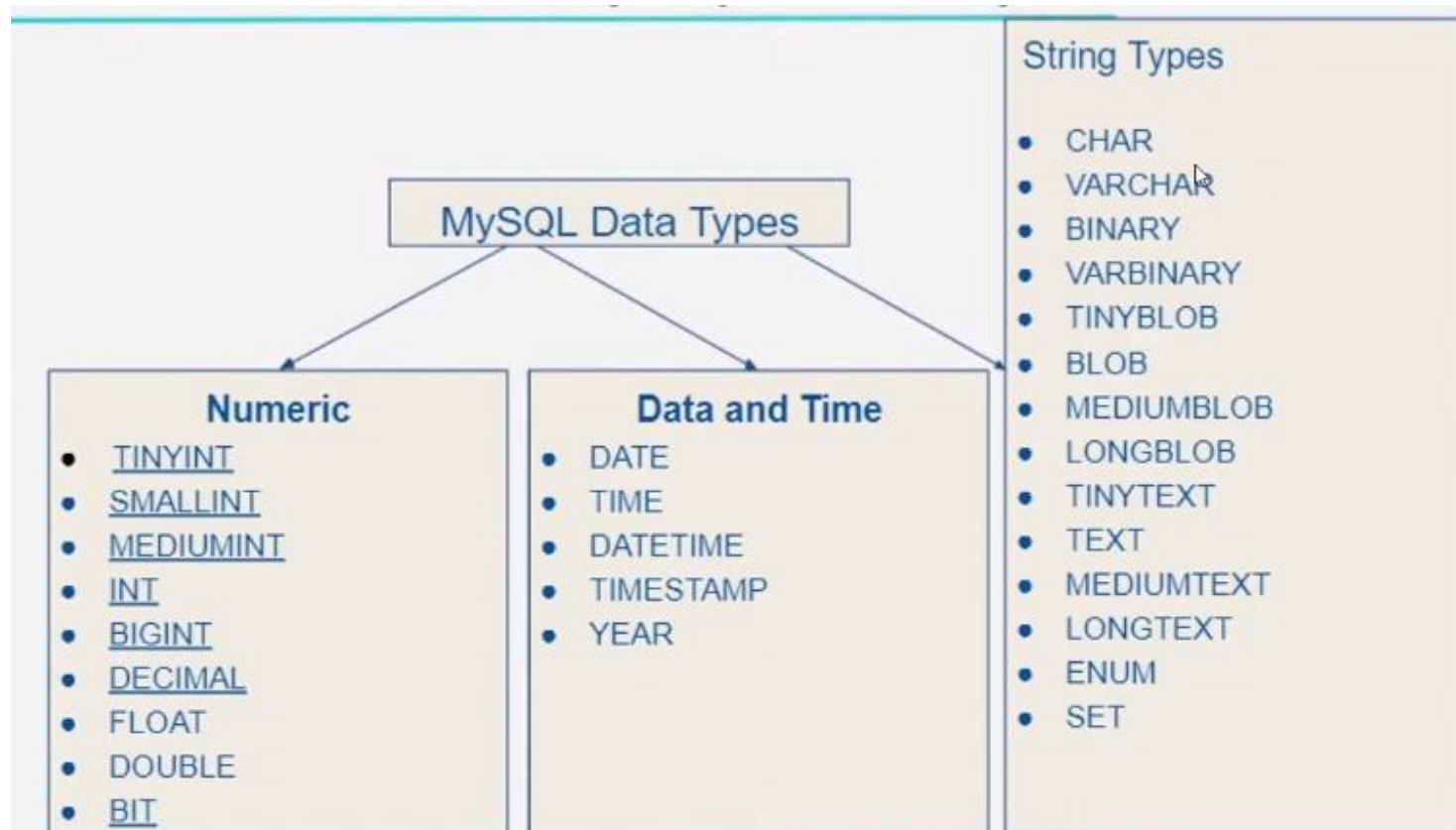
## Список рекомендованих джерел до теми 2

1. Reed M. SQL: 3 books in 1 – The Ultimate Beginner, Intermediate & Expert Guides To Master SQL Programming Quickly with Practical Exercises, London, England : Amazon, 2022, 348 p.
2. Korotkevitch D. SQL Server Advanced Troubleshooting and Performance Tuning. Best Practices and Techniques, USA : O’Reilly, 2022, 500 p.
3. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань, книга 2 : системи управління базами даних та знань: навч. посіб. Львів : Магнолія, 2021, 584 с.



## Тема 3

# Використання функцій агрегації та сортування даних у SQL-запитах



# Функції агрегації

## COUNT() Syntax

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

The `AVG()` function returns the average value of a numeric column.

## AVG() Syntax

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```



## SUM() Syntax

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

## MIN() Syntax

```
SELECT MIN(column_name)  
FROM table_name  
WHERE condition;
```

## MAX() Syntax

```
SELECT MAX(column_name)  
FROM table_name  
WHERE condition;
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Query 1 employees SQL File 5\* x SQL File 6\*

263

264

265

266

267 SELECT COUNT(\*)

268 FROM employees

269 WHERE gender='F'

270 AND first\_name IN ('Parto', 'Sarita')

271 AND last\_name NOT IN('Wrigley', 'Hitomi')

272 ;

Result Grid

COUNT(*)
173

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

```
263
264
265
266
267 SELECT max(birth_date)
268 FROM employees
269 WHERE gender='F'
270        AND first_name IN ('Parto', 'Sarita')
271        AND last_name NOT IN('Wrigley', 'Hitomi')
272 ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

max(birth_date)
1964-11-13



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
263
264
265
266
267 SELECT max(birth_date), min(birth_date)
268 FROM employees
269 WHERE gender='F'
270         AND first_name IN ('Parto', 'Sarita')
271         AND last_name NOT IN('Wrigley', 'Hitomi')
272 ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	max(birth_date)	min(birth_date)
▶	1964-11-13	1952-02-14

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with the 'employees' database expanded to show tables like 'departments', 'dept\_emp', 'dept\_manager', 'employees', 'salaries', and 'titles'. The right pane shows a query window with the following SQL code:

```
276  
277  
278  
279 SELECT avg(salary)  
280 FROM salaries  
281 WHERE emp_no=10001;  
282  
283  
284  
285
```

The result grid at the bottom shows the output of the query:

avg(salary)
75388.9412

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

```
285  
286  
287 • SELECT sum(salary)  
288 FROM salaries  
289 WHERE emp_no=10001;  
290  
291  
292  
293  
294
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sum(salary)
▶ 1281612

# Оператор GROUP BY

The **GROUP BY** statement is often used with aggregate functions ( **COUNT()** , **MAX()** , **MIN()** , **SUM()** , **AVG()** ) to group the result-set by one or more columns.

## GROUP BY Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```



MySQL

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas Information

293 • SELECT first\_name, count(emp\_no)

294 FROM employees

295 WHERE gender='M'

296 GROUP BY first\_name;

297

298

Result Grid

first_name	count(emp_no)
Georgi	147
Parto	141
Chirstian	129
Kyoichi	148
Saniya	154
Patricio	142
Eberhardt	137
Berni	134
Guoxiang	154
Kazuhito	151
Lillian	148
Mavukn	147



test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

299

```

300 • SELECT first_name, last_name, count(emp_no)
301 FROM employees
302 WHERE gender='M'
303 GROUP BY first_name, last_name;
304

```

Result Grid | Filter Rows: | Export: | Wrap

first_name	last_name	count(emp_no)
Georgi	Facello	2
Parto	Bamford	1
Chirstian	Koblick	1
Kyoichi	Maliniak	1
Saniya	Kalloufi	1
Patricio	Bridgland	1
Eberhardt	Terkki	1
Berni	Genin	1
Guoxiang	Nooteboom	1
Kazuhiro	Cappelletti	1

Administration Schemas

MySQL

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Query 1 employees SQL File 5\* x SQL File 6\*

305

306

307 • `SELECT gender, count(emp_no), min(birth_date)`

308 `FROM employees`

309 `GROUP BY gender;`

310

Result Grid

	gender	count(emp_no)	min(birth_date)
▶	M	179973	1952-02-01
	F	120051	1952-02-02

# Унікальне значення

## Syntax

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```





File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

```

318
319 • SELECT distinct title
320 FROM titles
321 ;
322
323

```

Result Grid | Filter Rows: | Export:

title
Senior Engineer
Staff
Engineer
Senior Staff
Assistant Engineer
Technique Leader
Manager



The screenshot shows the SQL Developer interface with a query window titled 'Query 1' containing the following SQL code:

```
324  
325 • SELECT dept_no, COUNT(distinct(emp_no))  
326 FROM dept_emp  
327 WHERE from_date >= '1975-01-01'  
328 GROUP BY dept_no;  
329  
330  
331  
332
```

Below the query, the 'Result Grid' displays the following data:

dept_no	COUNT(distinct(emp_no))
d001	20211
d002	17346
d003	17786
d004	73485
d005	85707
d006	20117
d007	52245
d008	21126
d009	23580

The interface also shows a Navigator pane on the left with the 'employees' schema expanded, listing tables such as departments, dept\_emp, dept\_manager, employees, salaries, and titles. The 'Administration' and 'Schemas' tabs are visible at the bottom.



# Оператор HAVING

## HAVING Syntax

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
HAVING condition  
ORDER BY column_name(s);
```



test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: test\_111 x

Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

329

330

331

332 • `SELECT first_name, last_name, count(emp_no)`

333 `FROM employees`

334 `GROUP BY first_name, last_name`

335 `HAVING count(emp_no)>=4;`

336

337

Result Grid | Filter Rows: | Export:

	first_name	last_name	count(emp_no)
▶	Kazunori	Vieth	4
	Charlene	Maraist	4
	Kristina	Setzner	4
	Alper	Lienhardt	4
	Gad	Krone	4
	Laurentiu	Cesareni	5

Administration Schemas



MySQL

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Query 1 employees SQL File 5\* x SQL File 6\*

339

340 • SELECT YEAR(hire\_date), count(distinct(emp\_no))

341 FROM employees

342 WHERE hire\_date >= '1960-01-01'

343 GROUP BY YEAR(hire\_date)

344 HAVING count(distinct(emp\_no)) >= 20000;

345

Result Grid | Filter Rows: | Export: | Wrap Cell C

	YEAR(hire_date)	count(distinct(emp_no))
▶	1985	35316
	1986	36150
	1987	33501
	1988	31436
	1989	28394
	1990	25610
	1991	22568
	1992	20402



# Сортування даних

The `ORDER BY` keyword is used to sort the result-set in ascending or descending order.

The `ORDER BY` keyword sorts the records in ascending order by default. To sort the records in descending order, use the `DESC` keyword.

## ORDER BY Syntax

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```



test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Query 1 employees SQL File 5\* x SQL File 6

```

355
356 • SELECT *
357 FROM salaries
358 ORDER BY salary;
359
360

```

Result Grid Filter Rows:

	emp_no	salary	from_date	to_date
▶	253406	38623	2002-02-20	9999-01-01
	49239	38735	1996-09-17	1997-09-17
	281546	38786	1996-11-13	1997-06-26
	15830	38812	2001-03-12	2002-03-12
	64198	38836	1989-10-20	1990-10-20
	475254	38849	1993-06-04	1994-06-04
	50419	38850	1996-09-22	1997-09-22
	34707	38851	1990-10-03	1991-10-03
	49239	38859	1995-09-18	1996-09-17
	274049	38864	1996-09-01	1997-09-01
	473390	38872	1995-03-20	1995-09-22
	12444	38874	1990-08-15	1991-08-15

Administration Schemas Information



test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Administration Schemas Information

```

355
356 • SELECT *
357 FROM salaries
358 ORDER BY salary DESC;
359
360

```

Result Grid Filter Rows:

	emp_no	salary	from_date	to_date
▶	43624	158220	2002-03-22	9999-01-01
	43624	157821	2001-03-22	2002-03-22
	254466	156286	2001-08-04	9999-01-01
	47978	155709	2002-07-14	9999-01-01
	253939	155513	2002-04-11	9999-01-01
	109334	155377	2000-02-12	2001-02-11
	109334	155190	2002-02-11	9999-01-01
	109334	154888	2001-02-11	2002-02-11
	109334	154885	1999-02-12	2000-02-12
	80823	154459	2002-02-22	9999-01-01
	493158	154376	2002-05-05	9999-01-01
	253939	154227	2001-04-11	2002-04-11
	205000	153715	2001-10-09	9999-01-01





test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

361

362 • `SELECT emp_no, sum(salary)`

363 `FROM salaries`

364 `WHERE from_date>='1971-01-01'`

365 `GROUP BY emp_no`

366 `ORDER BY sum(salary) DESC;`

367

368

369

Result Grid | Filter Rows:

emp_no	sum(salary)
109334	2553036
43624	2492873
66793	2383923
237542	2381119
47978	2374024
80823	2368170
68086	2305351
102962	2279289

Administration Schemas Information



MySQL V

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

376

377 • `SELECT dept_no, COUNT(distinct(emp_no))`

378 `FROM dept_emp`

379 `WHERE from_date>='1975-01-01'`

380 `GROUP BY dept_no`

381 `ORDER BY count(distinct(emp_no)) DESC;`

382

383

384

Result Grid Filter Rows: Export: Wrap Cell Co

	dept_no	COUNT(distinct(emp_no))
▶	d005	85707
	d004	73485
	d007	52245
	d009	23580
	d008	21126
	d001	20211
	d006	20117
	d003	17786
	d002	17346



# Оператор ALIAS

## Alias Column Syntax

```
SELECT column_name AS alias_name  
FROM table_name;
```

## Alias Table Syntax

```
SELECT column_name(s)  
FROM table_name AS alias_name;
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Administration Schemas

Information

No object selected

392  
393  
394  
395  
396 • `SELECT *, hire_date as h_d`  
397 `FROM employees;`  
398  
399  
400

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	emp_no	birth_date	first_name	last_name	gender	hire_date	h_d
▶	10001	1953-09-02	Georgi	Facello	M	1986-06-26	1986-06-26
	10002	1964-06-02	Bezael	Simmel	F	1985-11-21	1985-11-21
	10003	1959-12-03	Parto	Bamford	M	1986-08-28	1986-08-28
	10004	1954-05-01	Chirstian	Koblick	M	1986-12-01	1986-12-01
	10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12	1989-09-12
	10006	1953-04-20	Anneke	Preusig	F	1989-06-02	1989-06-02
	10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10	1989-02-10
	10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15	1994-09-15
	10009	1952-04-19	Sumant	Peac	F	1985-02-18	1985-02-18
	10010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24	1989-08-24

employees 2 x



test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Administration Schemas

Information

No object selected

401

402 • SELECT em.first\_name as name, dm.\*

403 FROM employees as em

404 LEFT JOIN dept\_manager as dm

405 ON em.emp\_no=dm.emp\_no

406 WHERE dm.dept\_no IS NOT NULL;

407

408

409

Result Grid Filter Rows: Export: Wrap C

	name	emp_no	dept_no	from_date	to_date
▶	Margareta	110022	d001	1985-01-01	1991-10-01
	Vishwani	110039	d001	1991-10-01	9999-01-01
	Ebru	110085	d002	1985-01-01	1989-12-17
	Isamu	110114	d002	1989-12-17	9999-01-01
	Shirish	110183	d003	1985-01-01	1992-03-21
	Karsten	110228	d003	1992-03-21	9999-01-01
	Krassimir	110303	d004	1985-01-01	1988-09-09
	Rosine	110344	d004	1988-09-09	1992-08-02
	Shem	110386	d004	1992-08-02	1996-08-30
	Oscar	110420	d004	1996-08-30	9999-01-01



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Administration Schemas Information

No object selected

416

417

418

419 • `SELECT YEAR(from_date) as from_date, avg(salary) as avg_salary`

420 `FROM salaries`

421 `WHERE from_date>='1960-01-01'`

422 `GROUP BY YEAR(from_date)`

423 `HAVING avg_salary<60000`

424 `ORDER BY avg_salary ASC;`

Result Grid

	from_date	avg_salary
▶	1985	53182.3580
	1986	54084.7786
	1987	54959.6284
	1988	55862.4465
	1989	56840.6728
	1990	57839.4601
	1991	58803.8697
	1992	59758.7416



## Завдання до теми 3

1. Визначити кількість працівників жіночої статі, які прийняті на роботу в 2000 році.
2. Визначити кількість працівників, які прийняті на роботу 12.09.1989 та чиє ім'я не містить символ 'а'.
3. Визначити максимальну зарплату працівників, які прийняті на роботу в 2000 році.
4. Визначити середню зарплату всіх працівників.
5. Згрупуйте дані за атрибутом gender. Який найменший рік у даті прийняття на роботу працівників групи 'М'?

## Список рекомендованих джерел до теми 3

1. Reed M. SQL: 3 books in 1 – The Ultimate Beginner, Intermediate & Expert Guides To Master SQL Programming Quickly with Practical Exercises, London, England : Amazon, 2022, 348 p.
2. Korotkevitch D. SQL Server Advanced Troubleshooting and Performance Tuning. Best Practices and Techniques, USA : O'Reilly, 2022, 500 p.
3. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань, книга 2 : системи управління базами даних та знань: навч. посіб. Львів : Магнолія, 2021, 584 с.



# Тема 4

## Реляційна модель даних

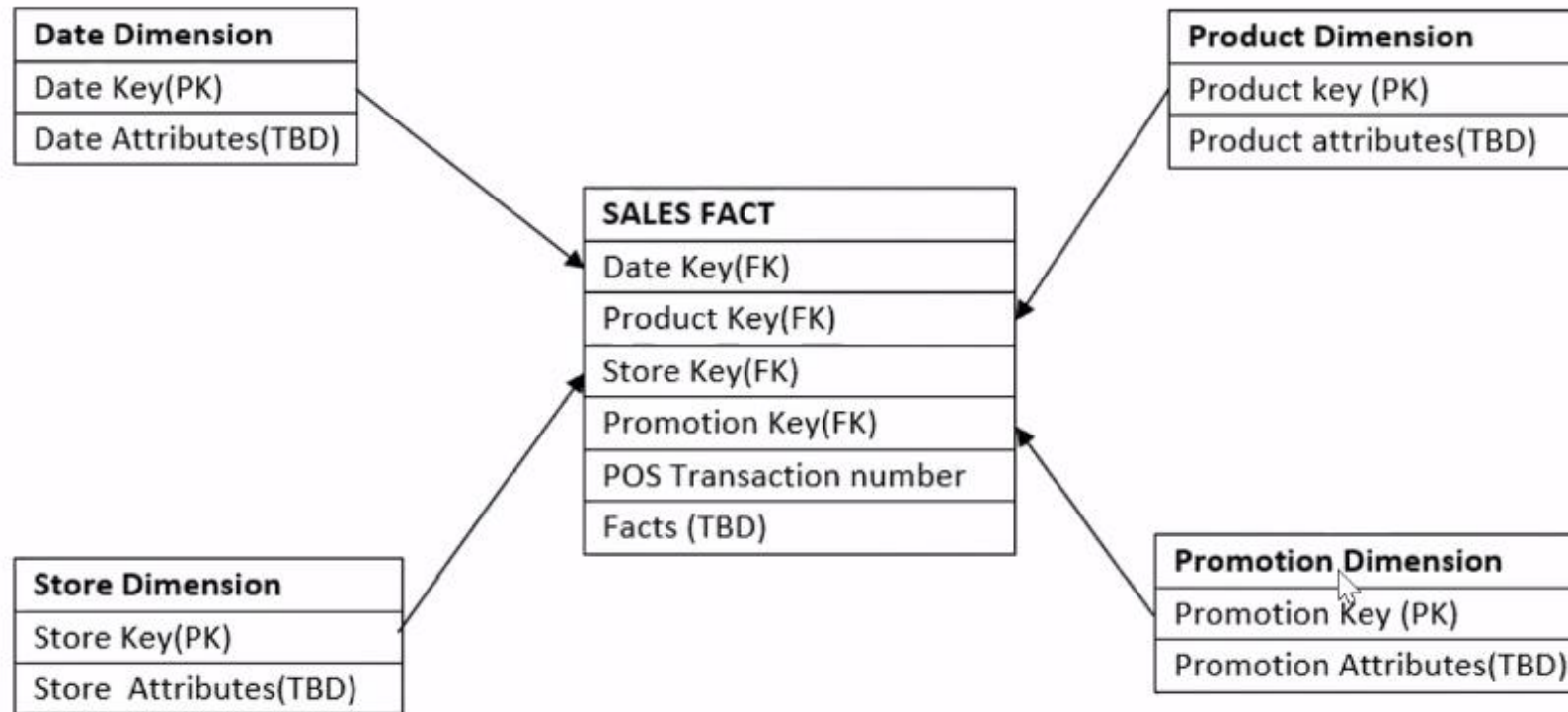
Ключ, атрибут

S#	SNAME	COURSE
S1	Smith	PHP
S2	Adams	Algebra





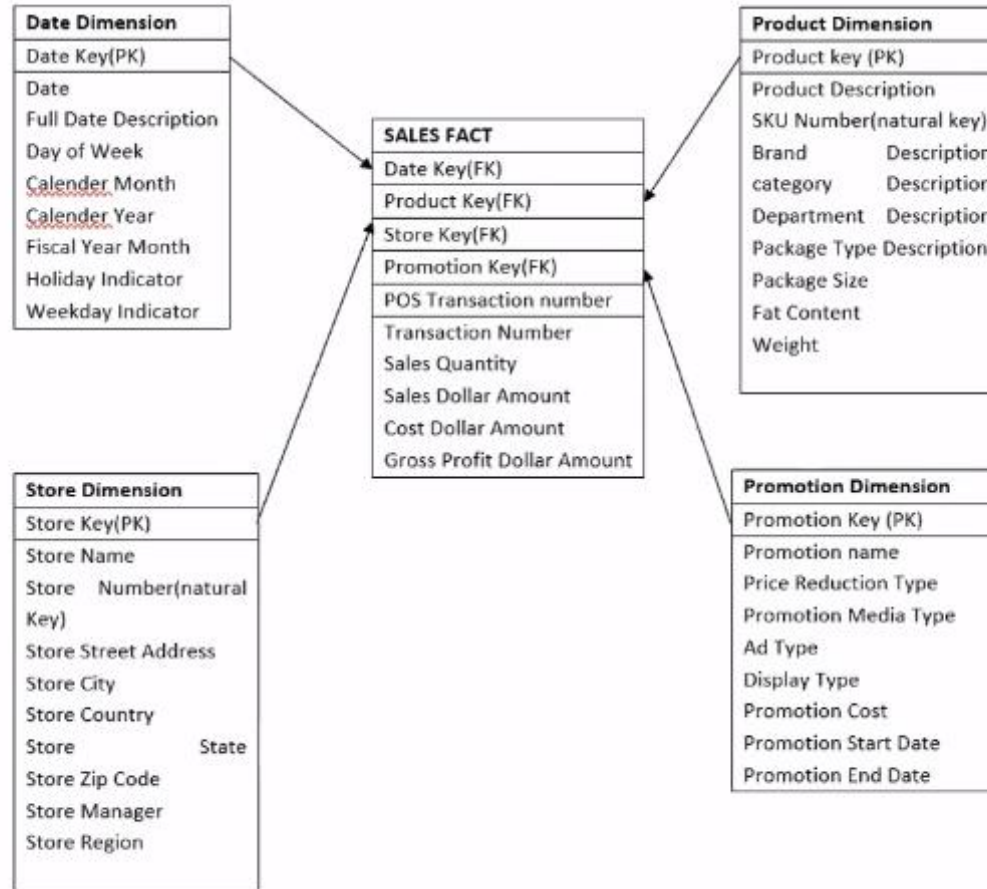
# Концептуальна модель даних



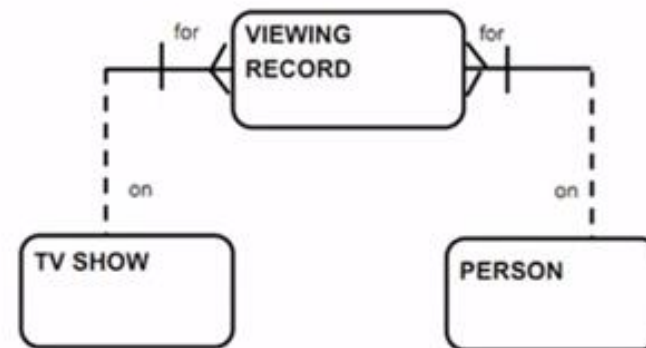
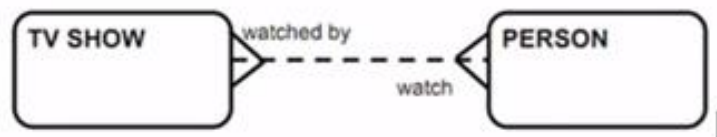
\*TBD = To be decided \*PK=Primary Key \*FK=Foreign key



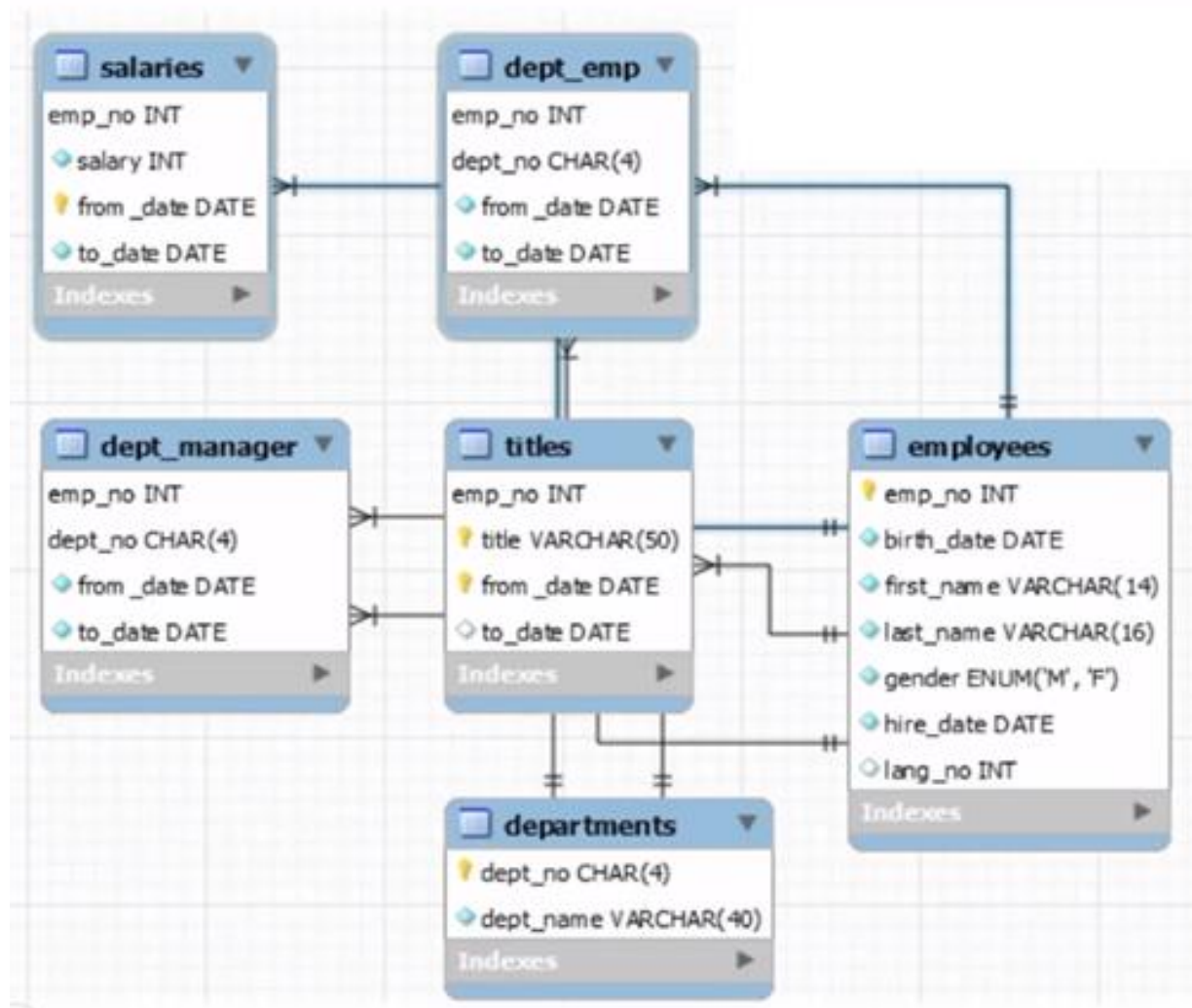
# Логічна модель даних



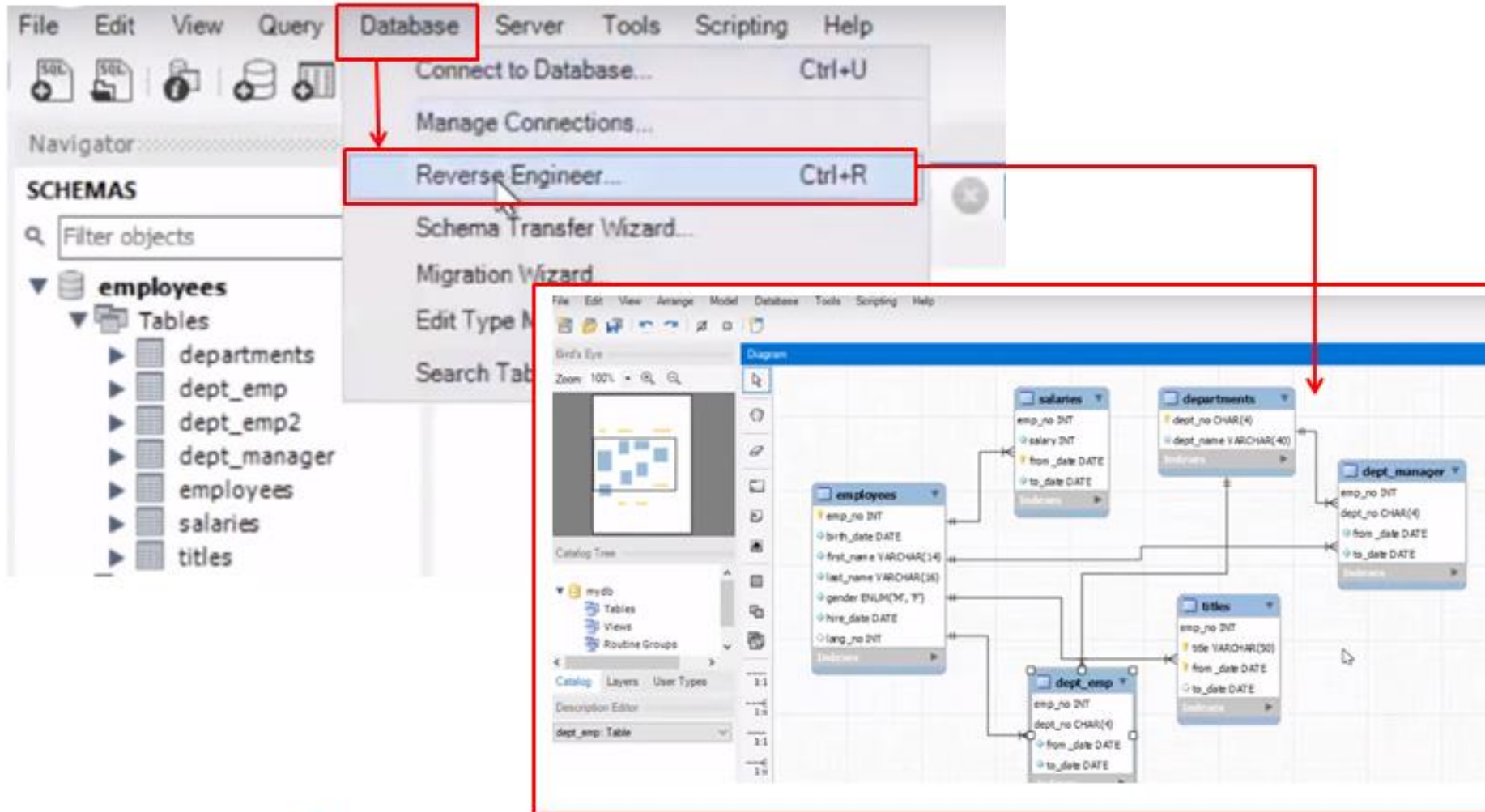
# Взаємозв'язки



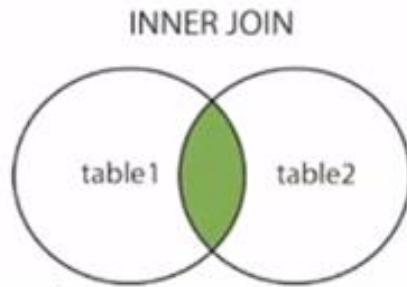
# ER діаграма



# Створення схеми БД в MySQL



# Оператор Inner Join



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1 employees SQL File 5\* SQL File 6\*

428 SELECT \*

429 FROM employees as em

430 INNER JOIN dept\_manager as dm

431 ON em.emp\_no=dm.emp\_no;

432

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
110022	1956-09-12	Margareta	Markovitch	M	1985-01-01	110022	d001	1985-01-01	1991-10-01
110085	1959-10-28	Ebru	Alpin	M	1985-01-01	110085	d002	1985-01-01	1989-12-17
110183	1953-06-24	Shirish	Ossenbruggen	F	1985-01-01	110183	d003	1985-01-01	1992-03-21
110303	1956-06-08	Krassimir	Wiegerle	F	1985-01-01	110303	d004	1985-01-01	1988-09-09
110511	1957-07-08	DeForest	Hagimont	M	1985-01-01	110511	d005	1985-01-01	1992-04-25
110725	1961-03-14	Petermela	Onuegbe	F	1985-01-01	110725	d006	1985-01-01	1989-05-06
111035	1962-02-24	Przemyslawa	Kaebbling	M	1985-01-01	111035	d007	1985-01-01	1991-03-07
111400	1959-11-09	Arie	Staelin	M	1985-01-01	111400	d008	1985-01-01	1991-04-08
111692	1954-10-05	Tonny	Butterworth	F	1985-01-01	111692	d009	1985-01-01	1988-10-17
110114	1957-03-28	Isamu	Legleitner	F	1985-01-01	110114	d002	1989-12-17	9999-01-01
110228	1958-12-02	Karsten	Sigstam	F	1985-08-01	110228	d003	1992-03-21	9999-01-01
110344	1961-09-07	Rosine	Cools	F	1985-11-01	110344	d004	1988-09-09	1992-08-02
110039	1963-06-21	Vishwani	Minakawa	M	1986-04-01	110039	d001	1991-10-01	9999-01-01
110800	1963-02-07	Sanjoy	Quadeer	F	1986-08-01	110800	d006	1991-09-12	1994-06-28
110567	1964-04-25	Leon	DasSarma	F	1986-10-01	110567	d005	1992-04-25	9999-01-01
111133	1955-03-16	Hauke	Zhang	M	1986-12-01	111133	d007	1991-03-07	9999-01-01
111534	1952-06-27	Hilary	Kambil	F	1988-01-01	111534	d008	1991-04-08	9999-01-01

Result 1 v

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Administration Schemas

Information: No object selected

Query 1 employees SQL File 5\* x SQL File 6\*

436 SELECT \*  
 437 FROM employees as em  
 438 INNER JOIN dept\_manager as dm  
 439 ON (em.emp\_no = dm.emp\_no AND  
 440 YEAR(em.hire\_date) = YEAR(dm.from\_date));

Result Grid Filter Rows: Export: Wrap Cell Content:

	emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
▶	110022	1956-09-12	Margareta	Markovitch	M	1985-01...	110022	d001	1985-01-01	1991-10-01
	110085	1959-10-28	Ebru	Alpin	M	1985-01...	110085	d002	1985-01-01	1989-12-17
	110183	1953-06-24	Shirish	Ossenbruggen	F	1985-01...	110183	d003	1985-01-01	1992-03-21
	110303	1956-06-08	Krassimir	Wegerle	F	1985-01...	110303	d004	1985-01-01	1988-09-09
	110511	1957-07-08	DeForest	Hagimont	M	1985-01...	110511	d005	1985-01-01	1992-04-25
	110725	1961-03-14	Peternela	Onuegbe	F	1985-01...	110725	d006	1985-01-01	1989-05-06
	110765	1954-05-22	Rutger	Hofmeyr	F	1989-01...	110765	d006	1989-05-06	1991-09-12
	111035	1962-02-24	Przemysława	Kaelbling	M	1985-01...	111035	d007	1985-01-01	1991-03-07
	111400	1959-11-09	Arie	Staelin	M	1985-01...	111400	d008	1985-01-01	1991-04-08
	111692	1954-10-05	Tonny	Butterworth	F	1985-01...	111692	d009	1985-01-01	1988-10-17
	111784	1956-06-14	Marjo	Giarratana	F	1988-02...	111784	d009	1988-10-17	1992-09-08



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions
- my1
- sys

Administration Schemas

Information: No object selected

```

466 SELECT *
467 FROM employees as em
468 INNER JOIN dept_manager as dm
469     ON em.emp_no = dm.emp_no
470 JOIN
471     (
472         SELECT emp_no, sum(salary) as salary_total
473         from salaries
474         GROUP BY emp_no
475     ) AS s
476     ON em.emp_no = s.emp_no;
477

```

Result Grid

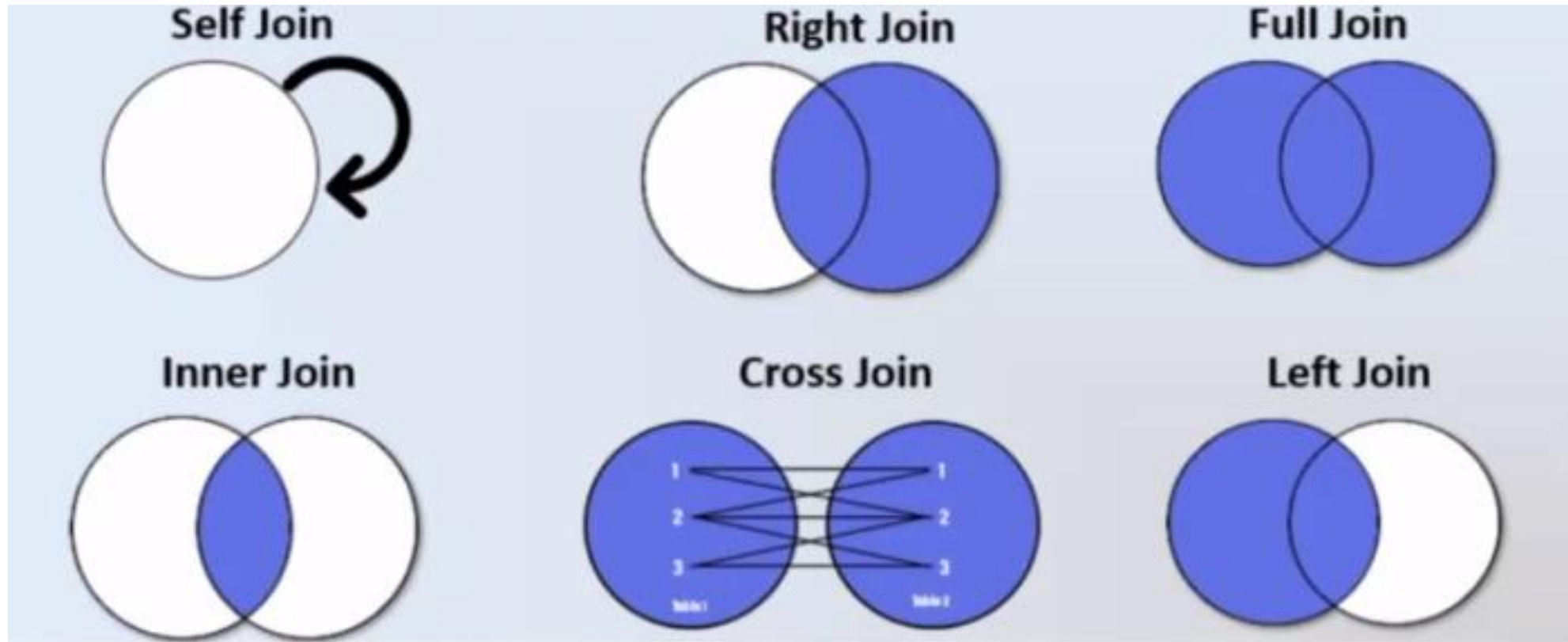
	emp_no	birth_date	first_name	last_name	gende	hire_date	emp_no	dept_no	from_date	to_date	emp_no	salary_total
▶	110022	1956-09-12	Margareta	Markovitch	M	1985-01...	110022	d001	1985-01-01	1991-10-01	110022	1604309
	110039	1963-06-21	Vishwani	Minakawa	M	1986-04...	110039	d001	1991-10-01	9999-01-01	110039	1488700
	110085	1959-10-28	Ebru	Alpin	M	1985-01...	110085	d002	1985-01-01	1989-12-17	110085	1310810
	110114	1957-03-28	Isamu	Legleitner	F	1985-01...	110114	d002	1989-12-17	9999-01-01	110114	1238562
	110183	1953-06-24	Shirish	Ossenbrug...	F	1985-01...	110183	d003	1985-01-01	1992-03-21	110183	1133824
	110228	1958-12-02	Karsten	Sigstam	F	1985-08...	110228	d003	1992-03-21	9999-01-01	110228	964474
	110303	1956-06-08	Krassimir	Wegerle	F	1985-01...	110303	d004	1985-01-01	1988-09-09	110303	1026942

Result 39 x





# Оператор Left/Right Join



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions
- my1
- sys

Administration Schemas

Information

No object selected

Query 1 employees SQL File 5\* x SQL File 6\*

480 • SELECT \*

481 FROM employees as em

482 LEFT JOIN dept\_manager as dm

483 ON em.emp\_no = dm.emp\_no;

Result Grid

Filter Rows: Export: Wrap Cell Content: Fetch rows:

emp_no	birth_date	first_name	last_name	gende	hire_date	emp_no	dept_no	from_date	to_date
110228	1958-12-02	Karsten	Sigstam	F	1985-08...	110228	d003	1992-03-21	9999-0...
110183	1953-06-24	Shirish	Ossenbrug...	F	1985-01...	110183	d003	1985-01-01	1992-0...
110114	1957-03-28	Isamu	Legleitner	F	1985-01...	110114	d002	1989-12-17	9999-0...
110085	1959-10-28	Ebru	Alpin	M	1985-01...	110085	d002	1985-01-01	1989-1...
110039	1963-06-21	Vishwani	Minakawa	M	1986-04...	110039	d001	1991-10-01	9999-0...
110022	1956-09-12	Margareta	Markovitch	M	1985-01...	110022	d001	1985-01-01	1991-1...
10001	1953-09-02	Georgi	Facello	M	1986-06...	NULL	NULL	NULL	NULL
10002	1964-06-02	Bezalel	Simmel	F	1985-11...	NULL	NULL	NULL	NULL
10003	1959-12-03	Parto	Bamford	M	1986-08...	NULL	NULL	NULL	NULL
10004	1954-05-01	Christian	Koblick	M	1986-12...	NULL	NULL	NULL	NULL
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09...	NULL	NULL	NULL	NULL
10006	1953-04-20	Anneke	Preusig	F	1989-06...	NULL	NULL	NULL	NULL
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02...	NULL	NULL	NULL	NULL
10008	1958-02-19	Saniya	Kalloufi	M	1994-09...	NULL	NULL	NULL	NULL



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Administration Schemas

Information

No object selected

```
486 SELECT *
487 FROM employees as em
488 RIGHT JOIN dept_manager as dm
489 ON em.emp_no = dm.emp_no;
490
491
```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
110022	1956-09-12	Margareta	Markovitch	M	1985-01...	110022	d001	1985-01-01	1991-10-01
110039	1963-06-21	Vishwani	Minakawa	M	1986-04...	110039	d001	1991-10-01	9999-01-01
110085	1959-10-28	Ebru	Alpin	M	1985-01...	110085	d002	1985-01-01	1989-12-17
110114	1957-03-28	Isamu	Legleitner	F	1985-01...	110114	d002	1989-12-17	9999-01-01
110183	1953-06-24	Shirish	Ossenbrug...	F	1985-01...	110183	d003	1985-01-01	1992-03-21
110228	1958-12-02	Karsten	Sigstam	F	1985-08...	110228	d003	1992-03-21	9999-01-01
110303	1956-06-08	Krassimir	Wegerle	F	1985-01...	110303	d004	1985-01-01	1988-09-09
110344	1961-09-07	Rosine	Cools	F	1985-11...	110344	d004	1988-09-09	1992-08-02
110386	1953-10-04	Shem	Kieras	M	1988-10...	110386	d004	1992-08-02	1996-08-30
110420	1963-07-27	Oscar	Ghazalie	M	1992-02...	110420	d004	1996-08-30	9999-01-01
110511	1957-07-08	DeForest	Hagimont	M	1985-01...	110511	d005	1985-01-01	1992-04-25
110567	1964-04-25	Leon	DasSarma	F	1986-10...	110567	d005	1992-04-25	9999-01-01
110725	1961-03-14	Peternela	Onuegbe	F	1985-01...	110725	d006	1985-01-01	1989-05-06
110765	1954-05-22	Rutger	Hofmeyr	F	1989-01...	110765	d006	1989-05-06	1991-09-12

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* SQL File 6\*

SCHMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions
- my1
- sys

Administration Schemas

Information

No object selected

493

494 • **SELECT \***

495 **FROM dept\_manager as dm**

496 **CROSS JOIN titles as t**

497 **ON dm.emp\_no = t.emp\_no;**

498

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	emp_no	dept_no	from_date	to_date	emp_no	title	from_date	to_date
▶	110022	d001	1985-01-01	1991-10-01	110022	Manager	1985-01-01	1991-10-01
	110022	d001	1985-01-01	1991-10-01	110022	Senior S...	1991-10-01	9999-01-01
	110039	d001	1991-10-01	9999-01-01	110039	Manager	1991-10-01	9999-01-01
	110039	d001	1991-10-01	9999-01-01	110039	Senior S...	1986-04-12	1991-10-01
	110085	d002	1985-01-01	1989-12-17	110085	Manager	1985-01-01	1989-12-17
	110085	d002	1985-01-01	1989-12-17	110085	Senior S...	1989-12-17	9999-01-01
	110114	d002	1989-12-17	9999-01-01	110114	Manager	1989-12-17	9999-01-01
	110114	d002	1989-12-17	9999-01-01	110114	Senior S...	1985-01-14	1989-12-17
	110183	d003	1985-01-01	1992-03-21	110183	Manager	1985-01-01	1992-03-21
	110183	d003	1985-01-01	1992-03-21	110183	Staff	1992-03-21	9999-01-01
	110228	d003	1992-03-21	9999-01-01	110228	Manager	1992-03-21	9999-01-01
	110228	d003	1992-03-21	9999-01-01	110228	Senior S...	1985-08-04	1992-03-21
	110303	d004	1985-01-01	1988-09-09	110303	Manager	1985-01-01	1988-09-09
	110303	d004	1985-01-01	1988-09-09	110303	Techniq...	1988-09-09	9999-01-01
	110344	d004	1988-09-09	1992-08-02	110344	Manager	1988-09-09	1992-08-02



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator Query 1 employees SQL File 5\* SQL File 6\*

SCHEMAS

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

```

502 SELECT *
503 FROM employees as em
504 LEFT JOIN dept_manager as dm
505     ON em.emp_no = dm.emp_no
506 LEFT JOIN
507     (
508         SELECT emp_no, sum(salary) as salary_total
509         from salaries
510         GROUP BY emp_no
511     ) AS s
512     ON em.emp_no = s.emp_no;

```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date	emp_no	salary_total
110228	1958-12-02	Karsten	Sigstam	F	1985-08...	110228	d003	1992-03-21	9999-0...	110228	964474
110183	1953-06-24	Shirish	Ossenbrug...	F	1985-01...	110183	d003	1985-01-01	1992-0...	110183	1133824
110114	1957-03-28	Isamu	Legleitner	F	1985-01...	110114	d002	1989-12-17	9999-0...	110114	1238562
110085	1959-10-28	Ebru	Alpin	M	1985-01...	110085	d002	1985-01-01	1989-1...	110085	1310810
110039	1963-06-21	Vishwani	Minakawa	M	1986-04...	110039	d001	1991-10-01	9999-0...	110039	1488700
110022	1956-09-12	Margareta	Markovitch	M	1985-01...	110022	d001	1985-01-01	1991-1...	110022	1604309
10001	1953-09-02	Georgi	Facello	M	1986-06...	NULL	NULL	NULL	NULL	10001	1281612
10002	1964-06-02	Bezalel	Simmel	F	1985-11...	NULL	NULL	NULL	NULL	10002	413127

Administration Schemas

Information

No object selected

# Оператор Limit

## LIMIT Syntax

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
LIMIT number;
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

Tables

- departments
- dept\_emp
- dept\_manager
- employees
- salaries
- titles

Views

Stored Procedures

Functions

my1

sys

Administration Schemas

Information

No object selected

Query 1 x employees SQL File 5\* SQL File 6\*

Don't Limit

```
14
15
16 SELECT *
17 FROM employees
18 LIMIT 10;
19
20
21
```

Result Grid

Filter Rows:

Edits

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06...
10002	1964-06-02	Bezael	Simmel	F	1985-11...
10003	1959-12-03	Parto	Bamford	M	1986-08...
10004	1954-05-01	Christian	Koblick	M	1986-12...
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09...
10006	1953-04-20	Anneke	Preusig	F	1989-06...
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02...
10008	1958-02-19	Saniya	Kaloufi	M	1994-09...
10009	1952-04-19	Sumant	Peac	F	1985-02...
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08...

employees 1 x



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 x employees SQL File 5\* SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

14

15

16 SELECT \*

17 FROM employees

18 LIMIT 10;

19

20

21

Result Grid Filter Rows: Edit: Expc

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06...
10002	1964-06-02	Bezael	Simmel	F	1985-11...
10003	1959-12-03	Parto	Bamford	M	1986-08...
10004	1954-05-01	Christian	Koblick	M	1986-12...
10005	1955-01-21	Kyoichi	Malniak	M	1989-09...
10006	1953-04-20	Anneke	Preusig	F	1989-06...
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02...
10008	1958-02-19	Saniya	Kaloufi	M	1994-09...
10009	1952-04-19	Sumant	Peac	F	1985-02...
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08...

employees 1 x



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

```
553
554
555 • SELECT *
556 FROM employees
557 ORDER BY emp_no
558 LIMIT 10 OFFSET 5;
```

Result Grid Filter Rows: Edit:

emp_no	birth_date	first_name	last_name	gender	hire_date
10006	1953-04-20	Anneke	Preusig	F	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15
10009	1952-04-19	Sumant	Peac	F	1985-02-18
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24
10011	1953-11-07	Mary	Sluis	F	1990-01-22
10012	1960-10-04	Patricio	Bridgland	M	1992-12-18
10013	1963-06-07	Eberhardt	Terkki	M	1985-10-20
10014	1956-02-12	Berni	Genin	M	1987-03-11
10015	1959-08-19	Guoxiang	Nooteboom	M	1987-07-02

MySQL

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator:.....

**SCHEMAS**

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Query 1 employees SQL File 5\* x SQL File 6\*

554

```
555 • SELECT *
556 FROM employees
557 WHERE gender='M'
558 ORDER BY hire_date
559 LIMIT 3;
```

560

561

Result Grid | Filter Rows: | Edit: | Exp

	emp_no	birth_date	first_name	last_name	gende	hire_date
▶	110085	1959-10-28	Ebru	Alpin	M	1985-01-01
	110511	1957-07-08	DeForest	Hagimont	M	1985-01-01
	110022	1956-09-12	Margareta	Markovitch	M	1985-01-01
*	NULL	NULL	NULL	NULL	NULL	NULL



# Оператор UNION

## UNION Syntax

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

## UNION ALL Syntax

The `UNION` operator selects only distinct values by default. To allow duplicate values, use `UNION ALL` :

```
SELECT column_name(s) FROM table1  
UNION ALL  
SELECT column_name(s) FROM table2;
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

- Tables
  - new\_table
  - t2
  - t1
  - test
- Views
- Stored Procedures
- Functions

Administration Schemas

Information: Result Grid Filter Rows: Export: Wrap Cell Content:

```
572 • DROP TABLE IF EXISTS t1;
573 • DROP TABLE IF EXISTS t2;
574
575 • CREATE TABLE t1
576 (id int);
577 • CREATE TABLE t2
578 (id int);
579
580 • INSERT INTO t1 VALUES (1),(2),(3);
581 • INSERT INTO t2 VALUES (2),(3),(4);
582
583 • SELECT id
584 FROM t1
585 UNION
586 SELECT id
587 FROM t2;
588
```

Table: t2

Columns: id int

id
1
2
3
4

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with the 'employees' database expanded to show tables like 'departments', 'dept\_emp', 'dept\_manager', 'employees', 'salaries', and 'titles'. The 'my1' database is also expanded to show tables 'new\_table', 't2', 't1', and 'test'. The right pane shows a query window with the following SQL code:

```
588  
589  
590 SELECT id  
591 FROM t1  
592 UNION ALL  
593 SELECT id  
594 FROM t2;  
595  
596
```

The result grid below the query shows the output of the query:

id
1
2
3
1
2
3
4



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Administration Schemas

Information

Schema: sys

```

596 SELECT emp_no, gender
597 FROM employees.employees
598 WHERE gender='M'
599 UNION
600 SELECT emp_no, gender
601 FROM employees.employees
602 WHERE gender='F';
603

```

Result Grid

emp_no	gender	tab
10001	M	t1
10003	M	t1
10004	M	t1
10005	M	t1
10008	M	t1
10012	M	t1
10013	M	t1
10014	M	t1
10015	M	t1
10016	M	t1
10019	M	t1
10020	M	t1
10021	M	t1

Result 16 x

Output

Action Output

#	Time	Action	Message
33	23:24:35	SELECT emp_no, gender, 't1' as tab FROM employees.employees WHERE gender...	300024 row(s) returned



# Підзапити

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'employees' expanded. The main editor shows a SQL query using a subquery to filter employees hired on or after 1991-01-01. The 'Result Grid' at the bottom shows the output of the query, listing first names and their counts.

```
604
605 • SELECT t1.first_name, count(t1.first_name)
606 FROM
607 (SELECT *
608 FROM employees
609 WHERE hire_date>='1991-01-01'
610 ) as t1
611 GROUP BY t1.first_name;
612
613
```

first_name	count(t1.first_name)
Saniya	91
Patricio	93
Kazuhiro	93
Cristinel	82
Lillian	98
Mayuko	95
Shahaf	85
Suzette	81
Yongqiao	101



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator:.....

SCHEMAS

Filter objects

- employees
  - Tables
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

Query 1 employees SQL File 5\* x SQL File 6\*

613

614 • `SELECT emp_no,salary`

615 `FROM salaries`

616 `WHERE salary>=(SELECT avg(salary) FROM salaries);`

617

618

619

620

621

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	emp_no	salary
▶	10001	66074
	10001	66596
	10001	66961
	10001	71046
	10001	74333
	10001	75286
	10001	75994

Administration Schemas

Information .....



test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

**SCHEMAS**

Filter objects

- employees
  - Tables
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

618  
619  
620  
621 • `SELECT first_name, last_name, emp_no`  
622 `FROM employees`  
623 `WHERE emp_no NOT IN`  
624 `(SELECT DISTINCT emp_no`  
625 `FROM dept_manager);`  
626

Result Grid Filter Rows: Edit:

	first_name	last_name	emp_no
▶	Georgi	Facello	10001
	Bezalel	Simmel	10002
	Parto	Bamford	10003
	Christian	Koblick	10004
	Kyoichi	Maliniak	10005
	Anneke	Preusig	10006
	Tzvetan	Zielinski	10007
	Saniya	Kalloufi	10008
	Sumant	Peac	10009
	Duangkaew	Piveteau	10010

Administration Schemas Information

No object selected

employees 7 x



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees SQL File 5\* x SQL File 6\*

SCHMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
627
628 • SELECT min(t1.salary), max(t1.salary), avg(t1.salary)
629 FROM
630     (SELECT YEAR(from_date), sum(salary) as salary
631      FROM salaries
632      GROUP BY year(from_date)
633      ) as t1;
634
635
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	min(t1.salary)	max(t1.salary)	avg(t1.salary)
▶	972864875	17535667603	10082264301.0556

# Експорт / імпорт даних з / в csv

from csv

The image shows a MySQL Workbench interface with a CSV file named 'SIM\_CARDS.csv' open. The CSV data is displayed in a table with columns: SCRD\_ID, ICC, CHANGE\_USER, CHANGE\_DATE, PUK1, and PUK2. The data includes rows for users like VLADIMIR, VERSHNYAK, KRASNOSCHEKVV, KOVALIYU, and IKOVAL. A blue arrow points from the CSV data to the 'Table Data Import Wizard' option in the 'my1' schema context menu. Another blue arrow points from the context menu towards the right side of the image.

	A	B	C	D	E	F
1	SCRD_ID	ICC	CHANGE_USER	CHANGE_DATE	PUK1	PUK2
2	49955	8938003990300136	VLADIMIR	19.11.1999 12:18:00	66912496	48978586
3	4112784	8938003990240811	VERSHNYAK	04.09.2007 12:36:32	95190296	17222682
4	5132648	8938003990247275	KRASNOSCHEKVV	22.04.2009 16:30:36	83527723	19300156
5	5750982	8938003990250658	KRASNOSCHEKVV	16.10.2009 09:42:06	12606379	
6	5843335	8938003990251582	KRASNOSCHEKVV	16.10.2009 09:58:52	3779492	
7	7486276	8938003990265151	VERSHNYAK	22.07.2011 10:36:24	14901239	
8	9677604	8938003990271919	VERSHNYAK	19.09.2013 12:52:18	47446689	
9	9677625	8938003990271920	VERSHNYAK	19.09.2013 12:52:18	6001808	
10	9677627	8938003990271920	VERSHNYAK	19.09.2013 12:52:18	914264	
11	9934682	8938003990273490	VERSHNYAK	19.09.2013 15:04:54	91493296	
12	11744967	8938003990275982	VERSHNYAK	16.12.2014 10:05:30	13561352	
13	11744968	8938003990275982	VERSHNYAK	16.12.2014 10:05:30	82924232	
14	11744969	8938003990275982	VERSHNYAK	16.12.2014 10:05:30	78719356	
15	11745003	8938003990275983	VERSHNYAK	16.12.2014 10:05:31	15828113	
16	11745004	8938003990275983	VERSHNYAK	16.12.2014 10:05:31	63726319	
17	11745040	8938003990275983	VERSHNYAK	16.12.2014 10:05:31	33664110	
18	12023206	8938003990277489	VERSHNYAK	04.03.2015 15:07:10	95446635	
19	12023212	8938003990277489	VERSHNYAK	04.03.2015 15:07:10	22641616	
20	12023214	8938003990277489	VERSHNYAK	04.03.2015 15:07:10	65359544	
21	12023215	8938003990277489	VERSHNYAK	04.03.2015 15:07:10	9467647	
22	12023220	8938003990277489	VERSHNYAK	04.03.2015 15:07:10	55641898	
23	13877585	8938003992425059	VERSHNYAK	16.12.2014 10:05:31	80000000	
24	16903607	8938003990294241	VERSHNYAK	16.12.2014 10:05:31	80000000	
25	18057221	8938003990297575	KOVALIYU	16.03.2018 16:36:41	70719762	
26	18502938	8938003990331911	KOVALIYU	16.03.2018 16:36:41	38866076	
27	19794348	8938003990334325	KOVALIYU	20.04.2018 11:58:37	37261870	
28	22128987	8938003990344171	IKOVAL	01.11.2018 12:58:01	23709209	





Table Data Import

Select Destination

Select destination table and additional options.

Use existing table: test\_db.tasks\_2

Create new table: test\_db . SIM\_CARDS

Drop table if exists

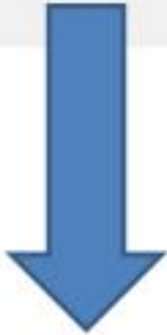


Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Source Column	Field Type
<input checked="" type="checkbox"/> SCRD_ID	int
<input checked="" type="checkbox"/> ICC	bigint
<input checked="" type="checkbox"/> CHANGE_USER	text
<input checked="" type="checkbox"/> CHANGE_DATE	text
<input checked="" type="checkbox"/> PUK1	text
<input checked="" type="checkbox"/> PUK2	int

SCRD_ID	ICC	CHANGE_...	CHANGE_...	PUK1	PUK2
49955	893800399...	VLADIMIR	19.11.1999...	66912496	48978586
4112764	893800399...	VERSHINYAK	04.09.2007...	95199296	17222682
5132648	893800399...	KRASNOSE...	22.04.2009...	83527723	19300156
5750982	893800399...	KRASNOSE...	16.10.2009...	12606379	76900384
5843335	893800399...	KRASNOSE...	16.10.2009...	03779492	84355775

< Back   Next >   Cancel



The screenshot shows a database management tool interface. The top menu includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The Navigator pane on the left shows a tree view of the 'test\_db' schema with tables like checklists\_old, employees\_dup, sim\_cards, stats, t1, t2, tasks, tasks2, tasks\_2, Views, Stored Procedures, and Functions. The main query editor contains the SQL query: `SELECT * FROM test_db.sim_cards;`. A blue arrow points to the query editor. Below the query editor, the 'Result Grid' shows the data returned by the query. The table has columns: SCRD\_ID, ICC, CHANGE\_USER, CHANGE\_DATE, PUK1, and PUK2. The data is as follows:

SCRD_ID	ICC	CHANGE_USER	CHANGE_DATE	PUK1	PUK2
49955	8938003990300136001	VLADIMIR	19.11.1999 12:18:00	66912496	48978586
4112764	8938003990240611642	VERSHNYAK	04.09.2007 12:36:32	95199296	17222682
5132648	8938003990247275987	KRASNOSCHEKVY	22.04.2009 16:30:39	83527723	19300156
5750982	8938003990250658574	KRASNOSCHEKVY	16.10.2009 09:42:05	1306379	76900384
5843335	8938003990251582104	KRASNOSCHEKVY	16.10.2009 09:56:52	3779492	84355775
7486276	8938003990265151615	VERSHNYAK	22.07.2011 10:36:24	14901239	91613350
9677604	8938003990271919930	VERSHNYAK	19.09.2013 12:52:18	47446689	24313524



to csv

The screenshot shows the SQL Server Enterprise Manager interface. The central pane displays a query: `SELECT * FROM test_db.sim_cards limit 10;` The result grid below shows data from the `sim_cards` table. The 'Export' button in the result grid is circled in red. An 'Export Data File' dialog box is open, showing the file name 'SIM\_CARDS.csv' and the format 'CSV (comma separated) (\*.csv)', both of which are also circled in red.

**Table: sim\_cards**

**Columns:**

- SCRD\_ID int
- ICC bigint
- CHANGE\_USER text
- CHANGE\_DATE text
- PUK1 bigint
- PUK2 int

SCRD_ID	ICC	CHANGE_USER	CHANGE_DATE	PUK1	PUK2
49955	8938003990300136001	VLADIMIR	19.11.1999 12:18		
4112764	8938003990240611642	VERSHNIYAK	04.09.2007 12:36		
5132648	8938003990247275987	KRASNOSCHEKVV	22.04.2009 16:30		
5750982	8938003990250658574	KRASNOSCHEKVV	16.10.2009 09:42:05	12606379	76900384
5843335	8938003990251582104	KRASNOSCHEKVV	16.10.2009 09:56:52	3779492	84355775
7486276	8938003990265151615	VERSHNIYAK	22.07.2011 10:36:24	14901239	91613350
9677604	8938003990271919930	VERSHNIYAK	19.09.2013 12:52:18	47446689	24313524
9677625	8938003990271920144	VERSHNIYAK	19.09.2013 12:52:18	6601808	57650851
9677627	8938003990271920169	VERSHNIYAK	19.09.2013 12:52:18	914264	96822627
9934662	8938003990273490518	VERSHNIYAK	19.09.2013 15:04:54	91493296	50191608

to csv

The screenshot shows the SQL Server Enterprise Manager interface with the 'Table Data Export Wizard' dialog box open. The wizard is configured to export data from the 'test\_db.sim\_cards' table. The columns selected for export are SCRD\_ID, ICC, CHANGE\_USER, CHANGE\_DATE, PUK1, and PUK2. The 'Select / Deselect all entries' checkbox is checked. The dialog box has a 'Back' button, a 'Next >' button, and a 'Cancel' button. The 'Advanced >>' button is also visible at the bottom left of the dialog box.

**Table Data Export Wizard**

Select data for export

Select source table for export: **test\_db.sim\_cards**

Select columns you'd like to export

Export	Column name
<input checked="" type="checkbox"/>	SCRD_ID
<input checked="" type="checkbox"/>	ICC
<input checked="" type="checkbox"/>	CHANGE_USER
<input checked="" type="checkbox"/>	CHANGE_DATE
<input checked="" type="checkbox"/>	PUK1
<input checked="" type="checkbox"/>	PUK2

Select / Deselect all entries

Row Offset:  Count:

Advanced >>



# Оператор CASE

## CASE Syntax

```
CASE  
  WHEN condition1 THEN result1  
  WHEN condition2 THEN result2  
  WHEN conditionN THEN resultN  
  ELSE result  
END;
```



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```
647
648
649
650 SELECT *,
651 CASE gender
652     WHEN 'M' THEN 'male_group'
653     WHEN 'F' THEN 'female_group'
654     ELSE 'test'
655 END as gender_droup
656 FROM employees;
657
658
659
```

Result Grid

Filter Rows: | Exports: | Wrap Cell Content: | Fetch rows:

	emp_no	birth_date	first_name	last_name	gende	hire_date	gender_droup
▶	10001	1953-09-02	Georgi	Facello	M	1986-06-26	male_group
	10002	1964-06-02	Bezael	Simmel	F	1985-11-21	female_group
	10003	1959-12-03	Parto	Bamford	M	1986-08-28	male_group
	10004	1954-05-01	Christian	Koblick	M	1986-12-01	male_group
	10005	1955-01-21	Kyoichi	Malniak	M	1989-09-12	male_group
	10006	1953-04-20	Anneke	Preusig	F	1989-06-02	female_group

No object selected

Result 1 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

Query 1 employees SQL File 5\* x SQL File 6\*

Limit to 100 rows

```
659
660
661 SELECT *,
662 CASE gender
663     WHEN 'H' THEN 'male_group'
664     WHEN 'K' THEN 'female_group'
665
666 END as gender_droup
667 FROM employees;
668
```

Result Grid

	emp_no	birth_date	first_name	last_name	gende	hire_date	gender_droup
▶	10001	1953-09-02	Georgi	Facello	M	1986-06-26	NULL
	10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	NULL
	10003	1959-12-03	Parto	Bamford	M	1986-08-28	NULL
	10004	1954-05-01	Christian	Koblick	M	1986-12-01	NULL
	10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12	NULL
	10006	1953-04-20	Anneke	Preusig	F	1989-06-02	NULL

Result 2 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* SQL File 6\*

SCHEMAS

Filter objects:

- employees
  - Tables
    - departments
    - dept\_emp
    - dept\_manager
    - employees
    - salaries
    - titles
  - Views
  - Stored Procedures
  - Functions
- my1
- sys

```

672 SELECT *,
673     CASE gender
674       WHEN 'M' THEN 'male_group'
675       WHEN 'F' THEN 'female_group'
676       ELSE 'test'
677     END as gender_droup,
678     CASE
679       WHEN emp_no IN (10001, 10002) THEN 'id_g_1'
680       WHEN emp_no IN (10004, 10006) THEN 'id_g_2'
681       ELSE 'test'
682     END AS id_group
683 FROM employees;
684

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

emp_no	birth_date	first_name	last_name	gende	hire_date	gender_droup	id_group
10001	1953-09-02	Georgi	Facello	M	1986-06-26	male_group	id_g_1
10002	1964-06-02	Bezael	Simmel	F	1985-11-21	female_group	id_g_1
10003	1959-12-03	Parto	Bamford	M	1986-08-28	male_group	test
10004	1954-05-01	Christian	Koblick	M	1986-12-01	male_group	id_g_2
10005	1955-01-21	Kyoichi	Malniak	M	1989-09-12	male_group	test
10006	1953-04-20	Anneke	Preusig	F	1989-06-02	female_group	id_g_2
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10	female_group	test
10008	1958-02-19	Saniya	Kaloufi	M	1994-09-15	male_group	test
10009	1952-04-19	Sumant	Peac	F	1985-02-18	female_group	test

Administration Schemas

Information

No object selected

Result 3 x

MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1 sys

Administration Schemas

Information

No object selected

```

672 SELECT *,
673     CASE gender
674         WHEN 'M' THEN 'male_group'
675         WHEN 'F' THEN 'female_group'
676         ELSE 'test'
677     END as gender_droup,
678     CASE
679         WHEN emp_no IN (10001, 10002) THEN 'id_g_1'
680         WHEN emp_no IN (10004, 10006) THEN 'id_g_2'
681         ELSE 'test'
682     END AS id_group,
683     CASE
684         WHEN (emp_no IN (10001, 10002)) AND (first_name LIKE 'Georg%') THEN 'id_gr_1'
685         WHEN emp_no IN (10004, 10006) THEN 'id_g_2'
686         ELSE 'test'
687     END AS id_group_2
688 FROM employees;
689

```

Result Grid

emp_no	birth_date	first_name	last_name	gender	hire_date	gender_droup	id_group	id_group_2
10001	1953-09-02	Georgi	Facello	M	1986-06-26	male_group	d_g_1	id_gr_1+Georg
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	female_group	d_g_1	test
10003	1959-12-03	Parto	Bamford	M	1986-08-28	male_group	test	test
10004	1954-05-01	Christian	Koblick	M	1986-12-01	male_group	d_g_2	id_g_2

## Завдання до теми 4

1. Визначити зарплату менеджера департаменту номер d003.
2. Із таблиці зарплат створити штучний стовпчик `level_sal` за такою формою:  
`low_salary` від 30 000 до 50 000;  
`mid_salary` від 50 001 до 100 000;  
`hig_salary` від 100 001 до 150 000.
3. За цією ж формою створити групи і відобразити середню зарплату працівників групи `hig_salary`, округлену до цілого значення. Розрахувати середню зарплату працівників групи `hig_salary` (цілу частину числа).

## Список рекомендованих джерел до теми 4

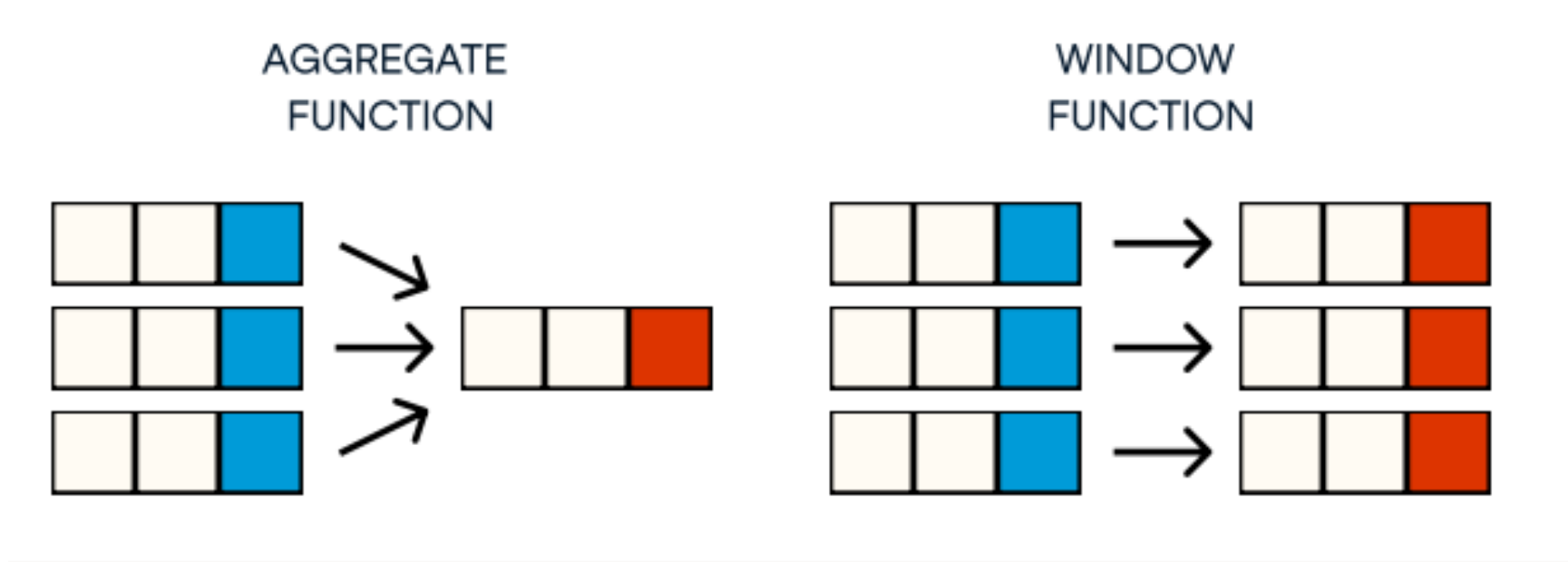
1. Reed M. SQL: 3 books in 1 – The Ultimate Beginner, Intermediate & Expert Guides To Master SQL Programming Quickly with Practical Exercises, London, England : Amazon, 2022, 348 p.
2. Korotkevitch D. SQL Server Advanced Troubleshooting and Performance Tuning. Best Practices and Techniques, USA : O'Reilly, 2022, 500 p.
3. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань, книга 2 : системи управління базами даних та знань: навч. посіб. Львів : Магнолія, 2021, 584 с.
4. Дяченко О.Ф. Математичні основи баз даних : навч. посіб. Маріуполь : МДУ, – Вінниця : Твори, 2020, 136 с.



## Тема 5

# Використання віконних функцій для написання SQL-запитів

### Що таке віконні функції

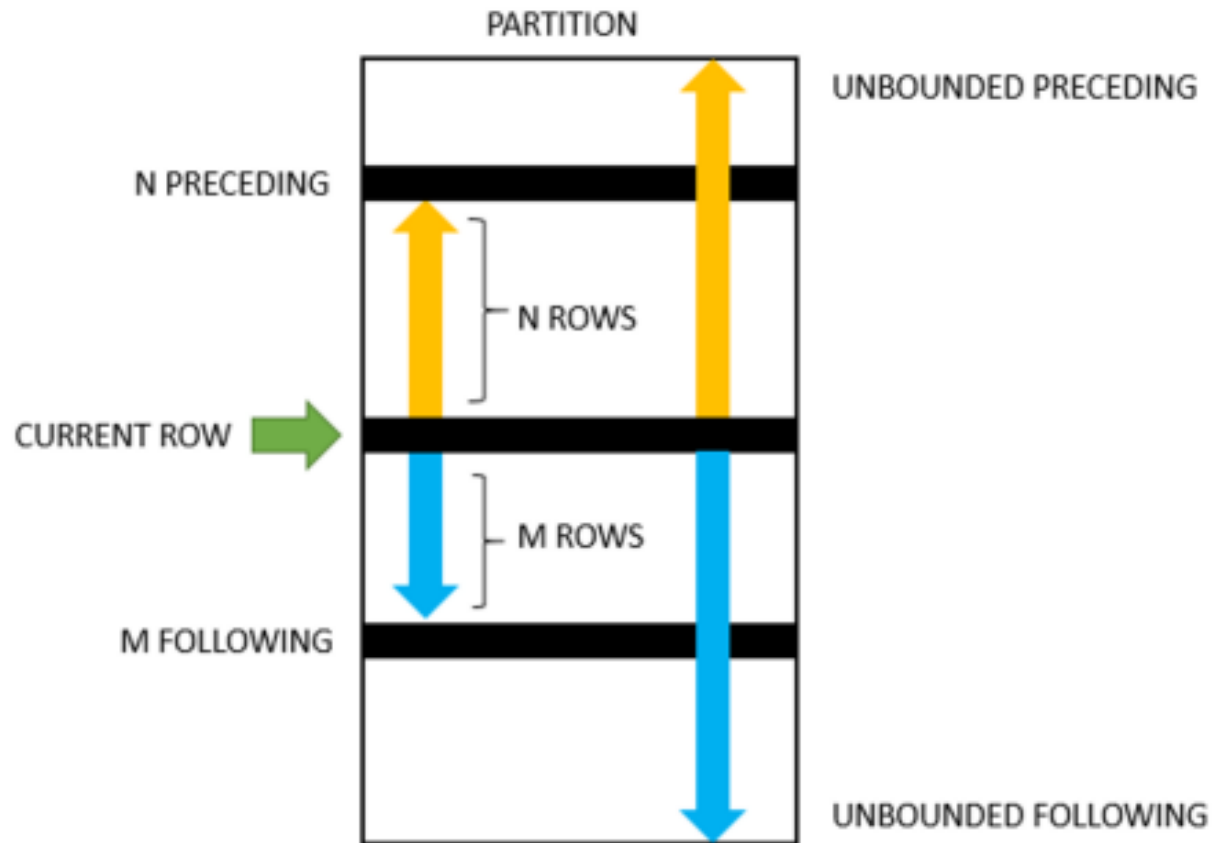


SELECT	list of columns, window functions
FROM	table / joint tables / subquery
WHERE	filtering clause
GROUP BY	list of columns
HAVING	aggregation filtering clause
ORDER BY	list of columns / window functions





# Що таке вікно?



# Загальний синтаксис

window\_function\_name(expression) OVER (

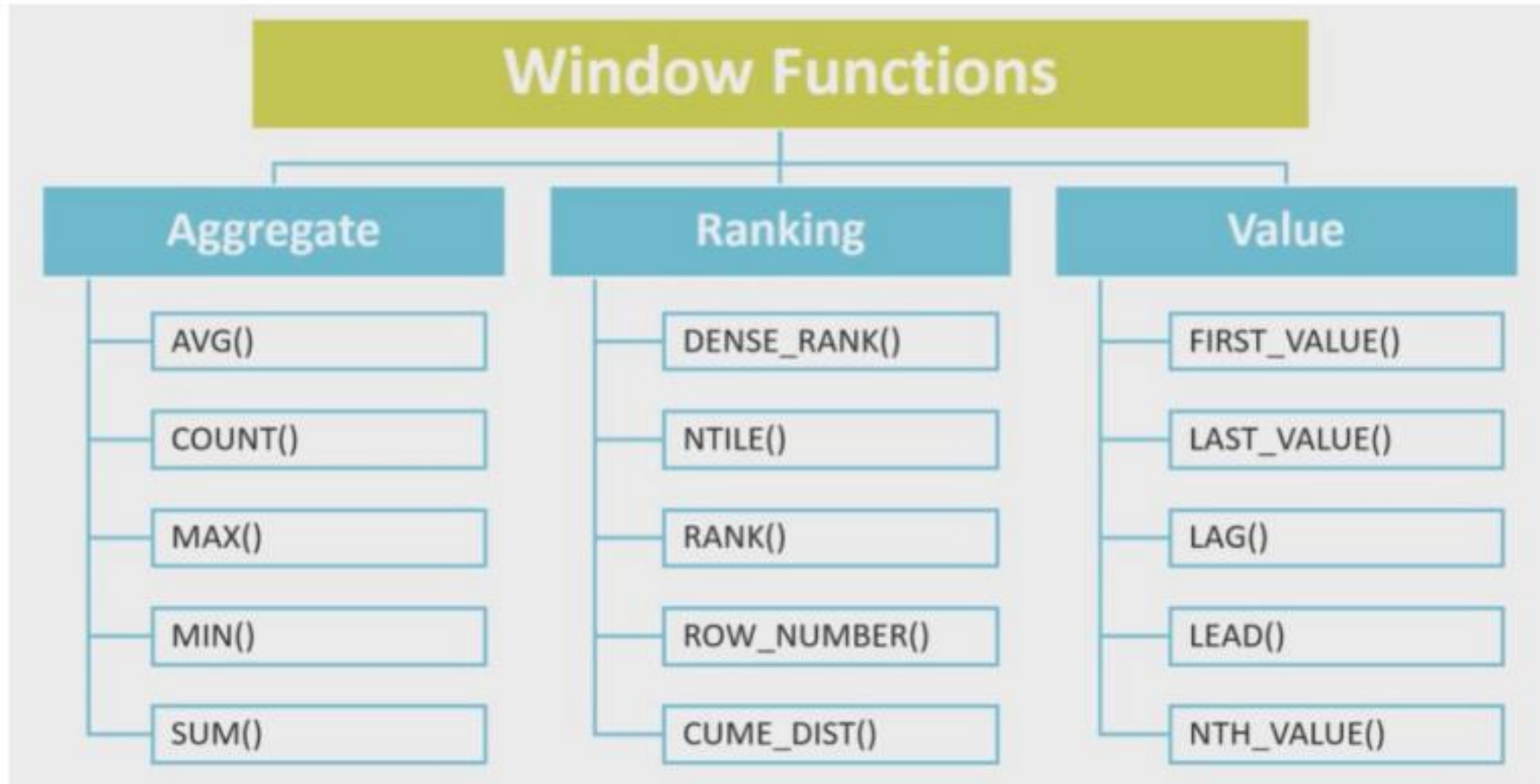
- [partition\_defintion]
- [order\_definition]
- [frame\_definition]

)

\*RANGE (ROWS) BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW (UNBOUNDED FOLLOWING)



# Типи функцій



## SQL Window Function Overview

	BusinessEntityID	TerritoryID	SalesYTD	(No column name)
1	274	NULL	559697.5639	559697.5639
2	285	NULL	172524.4512	732222.0151
3	287	NULL	519905.932	1252127.9471
4	280	1	1352577.1325	1352577.1325
5	283	1	1573012.9383	2925590.0708
6	284	1	1576562.1966	4502152.2674
7	275	2	3763178.1787	3763178.1787
8	277	3	3189418.3662	3189418.3662
9	276	4	4251368.5497	4251368.5497
10	281	4	2458535.6169	6709904.1666
11	279	5	2315185.611	2315185.611
12	278	6	1453719.4653	1453719.4653
13	282	6	2604540.7172	4058260.1825
14	290	7	3121616.3202	3121616.3202
15	288	8	1827066.7118	1827066.7118
16	286	9	1421810.9242	1421810.9242
17	289	10	4116871.2277	4116871.2277

```
SELECT BusinessEntityID
, TerritoryID
, SalesYTD
```

```
SUM(SalesYTD) OVER
```

```
(PARTITION BY TerritoryID
```

```
ORDER BY BusinessEntityID)
```

```
FROM Sales.SalesPerson
ORDER BY TerritoryID
```

OVER makes SUM a window function.

Window Frame: Rows from start of partition up to current row plus equal rows according to ORDER BY

$1352577.1 + 1573012.9 = 2925590.0$

Partition for every new TerritoryID

Total Resets at Partition



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* x SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information: No object selected

```

699
700 • select *,
701     max(salary) over (partition by emp_no) as max_sal_emp_no
702 from salaries
703 where emp_no between 10001 and 10010
704 order by emp_no;

```

Result Grid

emp_no	salary	from_date	to_date	max_sal_emp_no
10001	76884	1995-06-24	1996-06-23	88958
10001	80013	1996-06-23	1997-06-23	88958
10001	81025	1997-06-23	1998-06-23	88958
10001	81097	1998-06-23	1999-06-23	88958
10001	84917	1999-06-23	2000-06-22	88958
10001	85112	2000-06-22	2001-06-22	88958
10001	85097	2001-06-22	2002-06-22	88958
10001	88958	2002-06-22	9999-01-01	88958
10002	65828	1996-08-03	1997-08-03	72527
10002	65909	1997-08-03	1998-08-03	72527
10002	67534	1998-08-03	1999-08-03	72527
10002	69366	1999-08-03	2000-08-02	72527
10002	71963	2000-08-02	2001-08-02	72527
10002	72527	2001-08-02	9999-01-01	72527
10003	40006	1995-12-03	1996-12-02	43699
10003	43616	1996-12-02	1997-12-02	43699
10003	43466	1997-12-02	1998-12-02	43699



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

```

706
707
708
709 • select *,
710     max(salary) over (partition by emp_no) as max_sal_emp_no,
711     avg(salary) over (partition by emp_no) as avg_sal_emp_no
712 from salaries
713 where emp_no between 10001 and 10010
714 order by emp_no;
715

```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

emp_no	salary	from_date	to_date	max_sal_emp_no	avg_sal_emp_no
10001	85097	2001-06-22	2002-06-22	88958	75388.9412
10001	88958	2002-06-22	9999-01-01	88958	75388.9412
10002	65828	1996-08-03	1997-08-03	72527	68854.5000
10002	65909	1997-08-03	1998-08-03	72527	68854.5000
10002	67534	1998-08-03	1999-08-03	72527	68854.5000
10002	69366	1999-08-03	2000-08-02	72527	68854.5000
10002	71963	2000-08-02	2001-08-02	72527	68854.5000
10002	72527	2001-08-02	9999-01-01	72527	68854.5000
10003	40006	1995-12-03	1996-12-02	43699	43030.2857
10003	43616	1996-12-02	1997-12-02	43699	43030.2857
10003	43466	1997-12-02	1998-12-02	43699	43030.2857
10003	43636	1998-12-02	1999-12-02	43699	43030.2857
10003	43478	1999-12-02	2000-12-01	43699	43030.2857
10003	43699	2000-12-01	2001-12-01	43699	43030.2857

Administration Schemas

Information

No object selected

## Віконні функції ранжування

В функціях ранжування після ключового слова **OVER** обов'язковим має бути задання умови **ORDER BY**, за яким має бути сортування ранжування.

**ROW\_NUMBER()** - функція обчислює послідовність ранг (порядковий номер) рядочків в середині партиції, **НЕЗАЛЕЖНО** від того, чи є в рядочках повторювальні значення.

**RANK()** - функція обчислює ранг кожного рядочка в середині партиції. Якщо є повторювальні значення, функція повертає однаковий ранг для таких рядочків, пропускаючи при цьому наступний числовий ранг.

**DENSE\_RANK()** – те ж саме, що і **RANK**, тільки у випадку однакових значень **DENSE\_RANK** не пропускає наступний числовий ранг, а йде послідовно.

**Значення NULL будуть визначатися однаковим рангом**



## Приклад:

```
select name, subject, grade,  
       row_number() over (partition by name order by grade desc),  
       rank() over (partition by name order by grade desc),  
       dense_rank() over (partition by name order by grade desc)  
from student_grades;
```

asc name	asc subject	123 grade	123 row_number	123 rank	123 dense_rank
Маша		5	1	1	1
Маша		4	2	2	2
Маша		3	3	3	3
Маша		3	4	3	3
Петя		5	1	1	1
Петя		4	2	2	2
Петя		4	3	2	2
Петя		3	4	4	3

пропуск значення «3»

без пропусків значень «3»





## Функції зміщення

Це функції, які дозволяють переміщаючись по виділеній партиції таблиці, звертатися до попереднього значення рядка або крайніх значень рядків у партиції.

**LAG()** - функція, що повертає попереднє значення стовпця за порядком сортування.

**LEAD()** - функція, що повертає наступне значення стовпця за порядком сортування.

**FIRST\_VALUE()/LAST\_VALUE()** - функції, що повертають перше або останнє значення стовпця у зазначеній партиції. В аргументі вказують стовпець, значення якого потрібно повернути. У віконній функції під словом OVER обов'язково вказують ORDER BY умови.



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1 employees SQL File 5\* SQL File 6\*

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

```

727
728
729 • select *,
730     first_value(from_date) over (partition by emp_no order by from_date) as first_sal_date
731 from salaries
732 where emp_no between 10001 and 10010
733 order by emp_no;
734

```

Result Grid

Filter Rows: Exports: Wrap Cell Contents

emp_no	salary	from_date	to_date	first_sal_date
10001	85097	2001-06-22	2002-06-22	1986-06-26
10001	88958	2002-06-22	9999-01-01	1986-06-26
10002	65828	1996-08-03	1997-08-03	1996-08-03
10002	65909	1997-08-03	1998-08-03	1996-08-03
10002	67534	1998-08-03	1999-08-03	1996-08-03
10002	69366	1999-08-03	2000-08-02	1996-08-03
10002	71963	2000-08-02	2001-08-02	1996-08-03
10002	72527	2001-08-02	9999-01-01	1996-08-03
10003	40006	1995-12-03	1996-12-02	1995-12-03
10003	43616	1996-12-02	1997-12-02	1995-12-03
10003	43466	1997-12-02	1998-12-02	1995-12-03
10003	43636	1998-12-02	1999-12-02	1995-12-03
10003	43478	1999-12-02	2000-12-01	1995-12-03
10003	43699	2000-12-01	2001-12-01	1995-12-03

Result 20 x



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

Query 1 employees SQL File 5\* x SQL File 6\*

727

728

729 • `select *,`

730 `first_value(from_date) over (partition by emp_no order by from_date) as first_sal_date,`

731 `first_value(salary) over (partition by emp_no order by from_date) as first_sal`

732 `from salaries`

733 `where emp_no between 10001 and 10010`

734 `order by emp_no;`

Result Grid

emp_no	salary	from_date	to_date	first_sal_date	first_sal
10001	81097	1998-06-23	1999-06-23	1986-06-26	60117
10001	84917	1999-06-23	2000-06-22	1986-06-26	60117
10001	85112	2000-06-22	2001-06-22	1986-06-26	60117
10001	85097	2001-06-22	2002-06-22	1986-06-26	60117
10001	88958	2002-06-22	9999-01-01	1986-06-26	60117
10002	65828	1996-08-03	1997-08-03	1996-08-03	65828
10002	65909	1997-08-03	1998-08-03	1996-08-03	65828
10002	67534	1998-08-03	1999-08-03	1996-08-03	65828
10002	69366	1999-08-03	2000-08-02	1996-08-03	65828
10002	71963	2000-08-02	2001-08-02	1996-08-03	65828
10002	72527	2001-08-02	9999-01-01	1996-08-03	65828
10003	40006	1995-12-03	1996-12-02	1995-12-03	40006
10003	43616	1996-12-02	1997-12-02	1995-12-03	40006
10003	43466	1997-12-02	1998-12-02	1995-12-03	40006

Result 21 x



test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 employees SQL File 5\* SQL File 6\*

SCHMAS

Filter objects

employees

- Tables
  - departments
  - dept\_emp
  - dept\_manager
  - employees
  - salaries
  - titles
- Views
- Stored Procedures
- Functions

my1

sys

Administration Schemas

Information

No object selected

```

729 • select *,
730     first_value(from_date) over (partition by emp_no order by from_date) as first_sal_date,
731     first_value(salary) over (partition by emp_no order by from_date) as first_sal,
732     first_value(salary) over (partition by emp_no order by from_date desc) as last_sal
733 from salaries
734 where emp_no between 10001 and 10010
735 order by emp_no;
736

```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

emp_no	salary	from_date	to_date	first_sal_date	first_sal	last_sal
10001	66961	1990-06-25	1991-06-25	1986-06-26	60117	88958
10001	66596	1989-06-25	1990-06-25	1986-06-26	60117	88958
10001	66074	1988-06-25	1989-06-25	1986-06-26	60117	88958
10001	62102	1987-06-26	1988-06-25	1986-06-26	60117	88958
10001	60117	1986-06-26	1987-06-26	1986-06-26	60117	88958
10002	72527	2001-08-02	9999-01-01	1996-08-03	65828	72527
10002	71963	2000-08-02	2001-08-02	1996-08-03	65828	72527
10002	69366	1999-08-03	2000-08-02	1996-08-03	65828	72527
10002	67534	1998-08-03	1999-08-03	1996-08-03	65828	72527
10002	65909	1997-08-03	1998-08-03	1996-08-03	65828	72527
10002	65828	1996-08-03	1997-08-03	1996-08-03	65828	72527
10003	43311	2001-12-01	9999-01-01	1995-12-03	40006	43311
10003	43699	2000-12-01	2001-12-01	1995-12-03	40006	43311
10003	43478	1999-12-02	2000-12-01	1995-12-03	40006	43311

Result 23 x



MySQL Workbench

test\_111 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

Tables

- departments
- dept\_emp
- dept\_manager
- employees
- salaries
- titles

Views

Stored Procedures

Functions

my1

sys

Administration Schemas

Information

No object selected

Query 1 employees SQL File 5\* x SQL File 6\*

739

740

741 • `select *,`

742 `lag(salary,1,0) over (partition by emp_no order by from_date) as prev_salary`

743 `from salaries`

744 `where emp_no between 10001 and 10010`

745 `order by emp_no;`

746

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

emp_no	salary	from_date	to_date	prev_salary
10001	85097	2001-06-22	2002-06-22	85112
10001	88958	2002-06-22	9999-01-01	85097
10002	65828	1996-08-03	1997-08-03	0
10002	65909	1997-08-03	1998-08-03	65828
10002	67534	1998-08-03	1999-08-03	65909
10002	69366	1999-08-03	2000-08-02	67534
10002	71963	2000-08-02	2001-08-02	69366
10002	72527	2001-08-02	9999-01-01	71963
10003	40006	1995-12-03	1996-12-02	0
10003	43616	1996-12-02	1997-12-02	40006
10003	43466	1997-12-02	1998-12-02	43616
10003	43466	1000-12-02	1000-12-02	43466



## Завдання до теми 5

1. Створити стовпець для розрахунку накопичувальної суми заробітної плати кожного співробітника.
2. Розрахувати, яка різниця поточної отриманої кожним співробітником зарплати від максимальної.
3. Розрахувати ранг отриманої зарплати кожним співробітником.
4. Знайти різницю між отриманою співробітником зарплатою порівняно з попереднім періодом.

## Список рекомендованих джерел до теми 5

1. Reed M. SQL: 3 books in 1 – The Ultimate Beginner, Intermediate & Expert Guides To Master SQL Programming Quickly with Practical Exercises, London, England : Amazon, 2022, 348 p.
2. Korotkevitch D. SQL Server Advanced Troubleshooting and Performance Tuning. Best Practices and Techniques, USA : O'Reilly, 2022, 500 p.
3. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань, книга 2 : системи управління базами даних та знань: навч. посіб. Львів : Магнолія, 2021, 584 с.

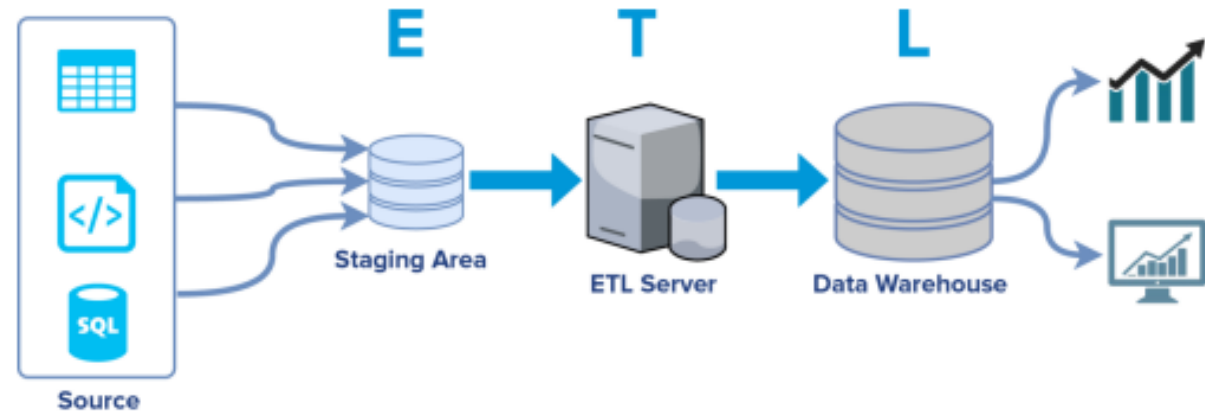


# Тема 6

## Технологія ETL. Візуалізація даних

### Extract - Transform - Load:

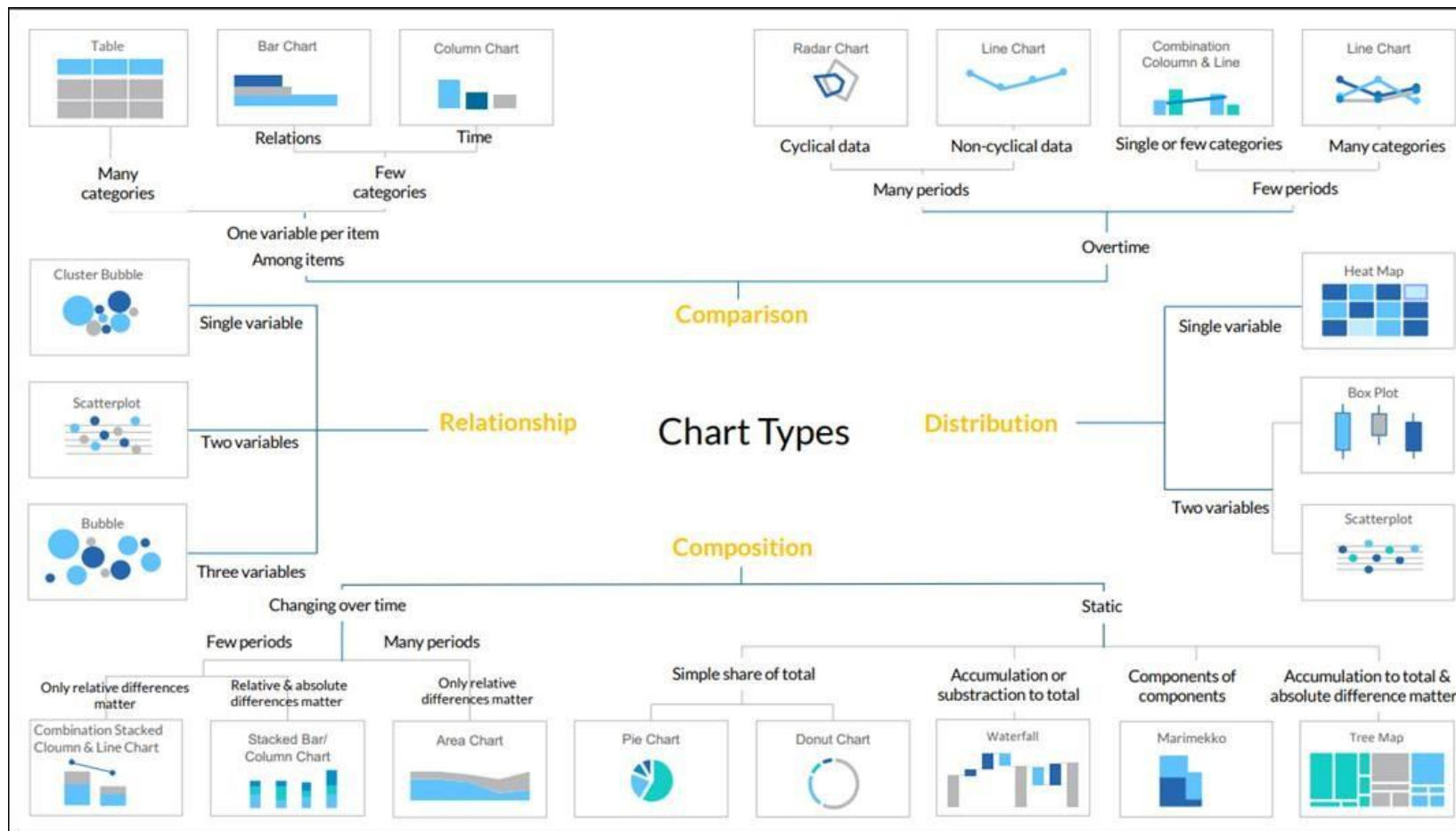
- Отримання даних  
(пошук, завантаження, читання)
- Перетворення  
(обробка, очистка, агрегація, перевірка)
- Завантаження  
(підготовка даних та їх структури до аналізу)







# Підбір інструментів візуалізації



<https://blogs.sap.com/2020/11/13/sap-successfactors-people-analytics-report-stories-data-visualization-series-selecting-the-right-chart-part-1/>



# Використання результатів запиту

The screenshot displays the MySQL Workbench interface. The main window shows a query window with the following SQL code:

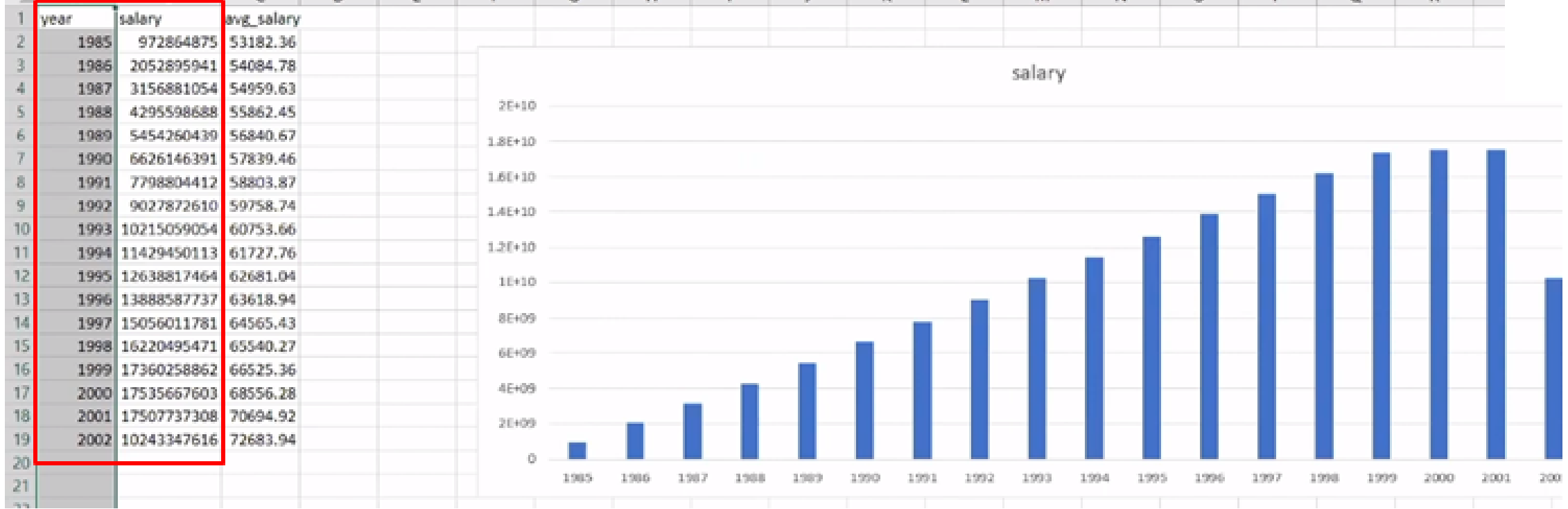
```
692  
693 • SELECT YEAR(from_date),  
694       sum(salary) as salary, avg(salary) as avg_salary  
695 FROM salaries  
696 GROUP BY YEAR(from_date);  
697
```

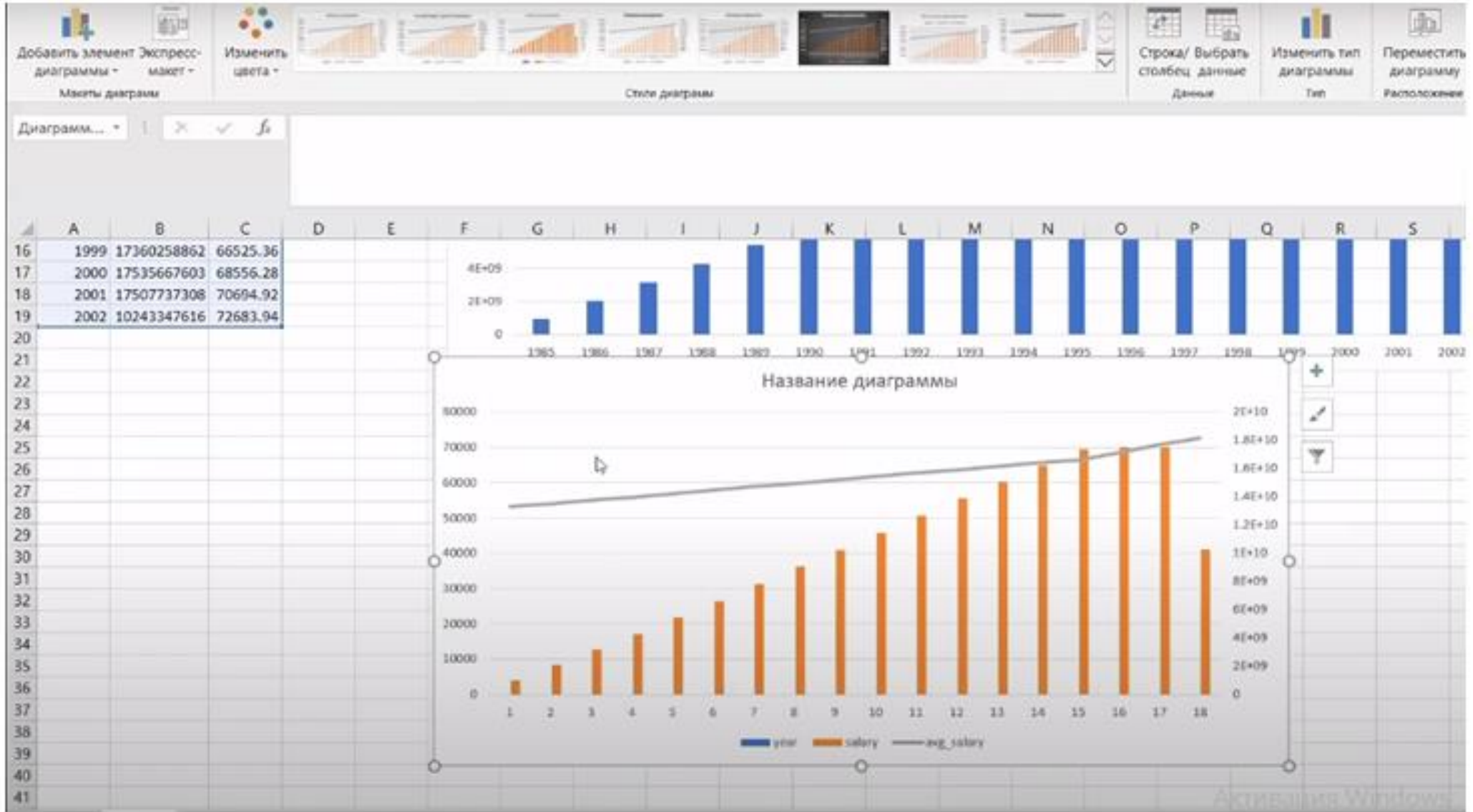
The query results are displayed in a table with the following data:

YEAR(from_date)	salary	avg_salary
1986	2052895941	54084.7786
1987	3156881054	54959.6284
1988	4295598688	55862.4465
1989	5454260439	56840.6728
1990	6626146391	57839.4601
1991	7798804412	58803.8697
1992	9027872610	59758.7416
1993	10215059054	60753.6565
1994	11429450113	61727.7589
1995	12638817464	62681.0430
1996	13888587737	63618.9426
1997	15056011781	64565.4264
1998	16220495471	65540.2683
1999	17360258862	66525.3619
2000	17535667603	68556.2781
2001	17507737308	70694.9159
2002	10243347616	72683.9397
1985	972864875	53182.3580

An "Export Resultset" dialog box is open, showing the file name "data\_ex.csv" and the file type "CSV (separated) (\*.csv)". A blue arrow points from the "Export" button in the main window to the dialog box.

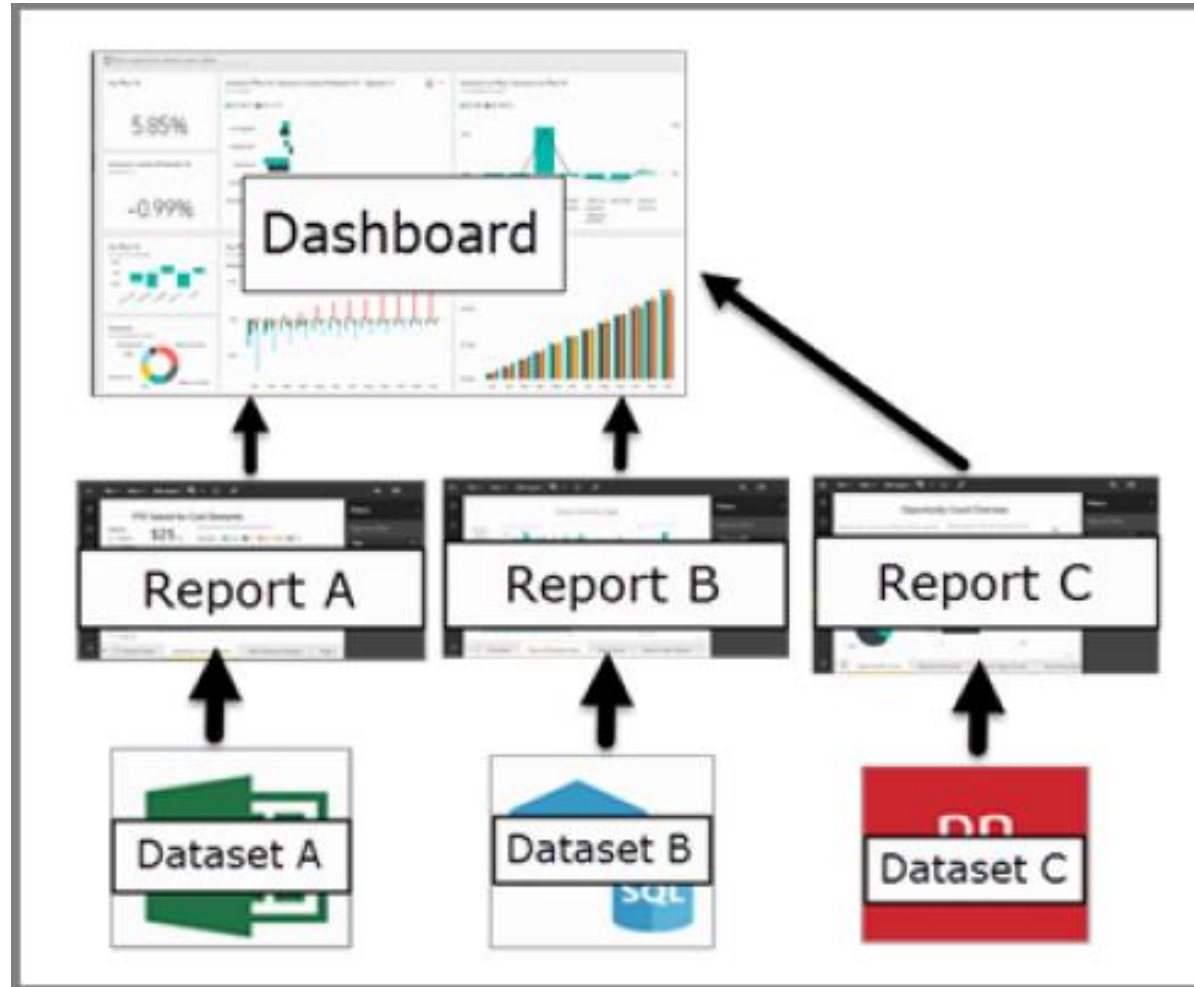






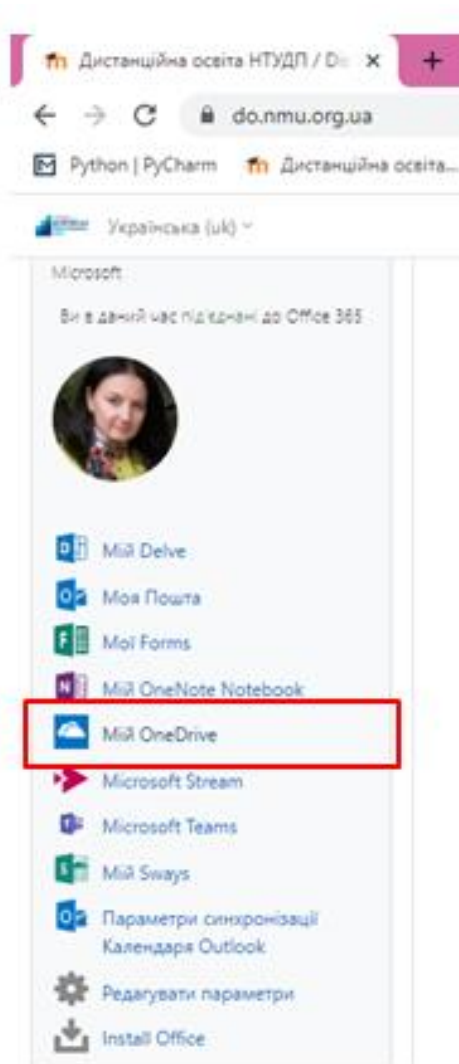
# Побудова звітів та дашбордів в Power BI

<https://learn.microsoft.com/en-us/power-bi/create-reports/service-dashboards>

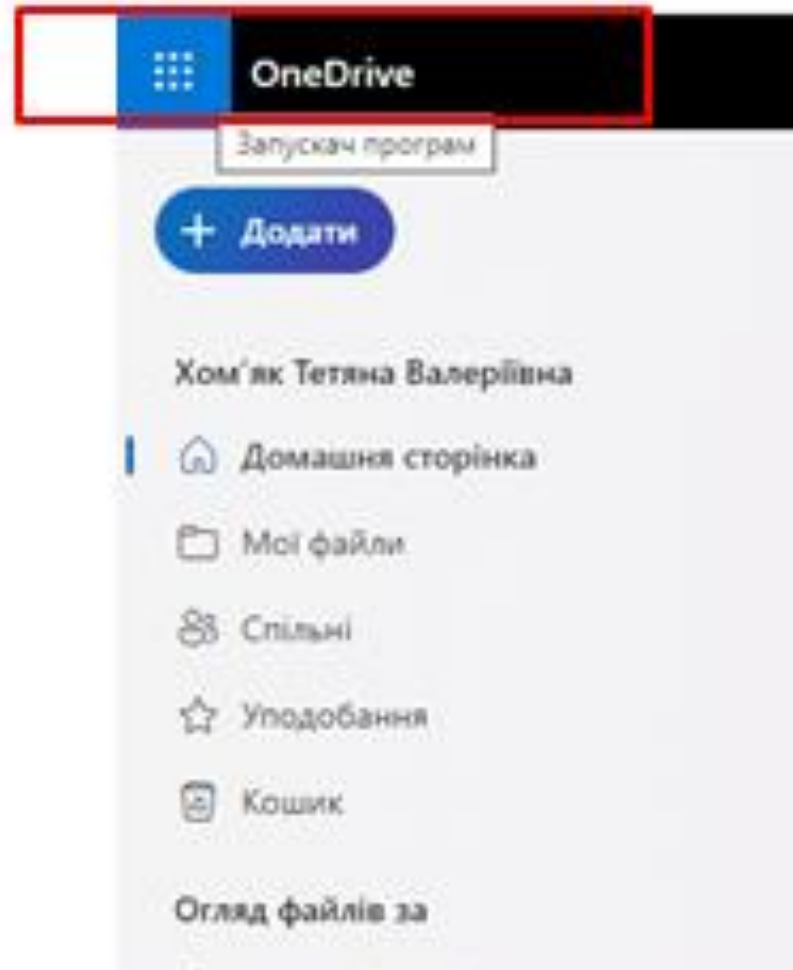




# 1 крок (використання Power BI з OneDrive)

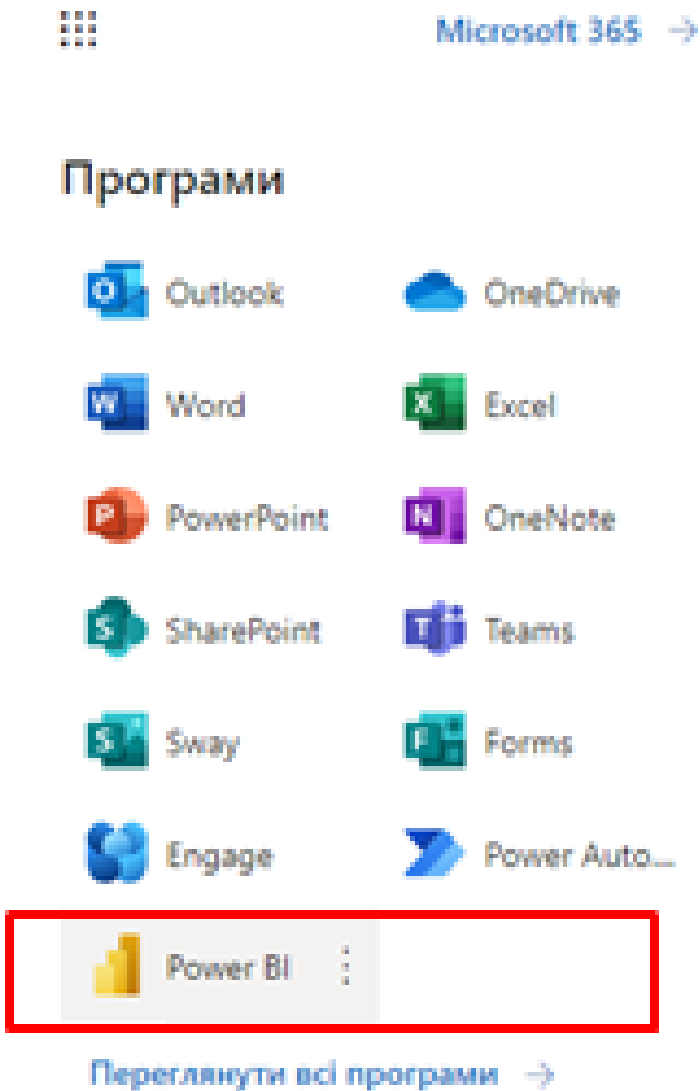


## 2 крок (запускач програм)





## 3 крок (Power BI)



Дистанційна освіта НТУДП / Dis x Головна – OneDrive x Power BI x +

app.powerbi.com/home?tenant=6cbb82a7-8f8b-4e21-8922-e8dd1f03e636&experience=power-bi

Python | PyCharm Дистанційна освіта... Курс робоча пошта Список телефонів... Start | Learn Вступ до курсу мо... Системний аналіз |... >> | Другие закладки

Power BI Главная Поиск Пробная версия: Осталось 59 дн.

Представляем приложение Power BI в Teams Работайте над данными вместе с коллегами и выполняйте необходимые действия. Выберите Открыть в Teams для начала работы [Дополнительные сведения](#) **Открыть в Teams**

**+ Создать отчет** Место сохранения новых элементов: Моя рабочая область

Избранные и часто используемые

**Моя рабочая область**

**Недавние** Избранное Мои приложения  **Фильтр**

Имя	Тип	Открыто	Расположение
10001 salary	Отчет	41 мин назад	Mv workspace

Активация Windows  
Чтобы активировать Windows, перейдите к параметрам компьютера.



# 4 крок (робоча область Power BI)

The screenshot shows the Power BI 'My workspace' (Моя рабочая область) interface. The browser address bar shows the URL: `app.powerbi.com/groups/me/list?tenant=6cbb82a7-8f8b-4e21-8922-e8dd1f03e636&experience=power-bi`. The interface includes a search bar, a 'Создать' (Create) button, and a table of workspace items.

Имя	Тип	Владелец	Обновлено	Следующее обновление	Подтвержде
10001_salary	Отчет	Хом'як Тетяна Вал...	07.11.23, 11:32:25	—	—
BI_importExcel	Отчет	Хом'як Тетяна Вал...	23.08.22, 19:19:10	—	—
BI_importExcel	Набор данных	Хом'як Тетяна Вал...	23.08.22, 19:19:10	Н/Д	—
BI_importExcel.pbix	Информационная ...	Хом'як Тетяна Вал...	—	—	—
Dataflow 1	Поток данных 2-г...	Хом'як Тетяна Вал...	07.11.23, 11:28:54	Н/Д	—
my1	Набор данных (по ...	Хом'як Тетяна Вал...	06.11.23, 22:25:55	Н/Д	—
my1	Хранилище	Хом'як Тетяна Вал...	—	Н/Д	—

Additional interface elements include a sidebar with navigation options (Главная, Создать, Обзор, Центр данных..., Приложения, Рабочие области, Моя рабочая...), a search bar with 'Поиск', and a 'Пробная версия: Осталось 59 дн.' (Trial version: 59 days left) notification.



## 5 крок (створення набору даних)

Power BI Моя рабочая область

Создать

- Отчет
- Отчет на страницу
- Система показателей
- Панель мониторинга
- Набор данных**
- Потоковый набор данных
- Поток данных потоковой передачи
- Дополнительные параметры

Объедините источники данных в набор данных для визуализации или предоставления общего доступа к нему.

Тип	Владелец	Обновлено	Следующее обновление	Подтвержде
Отчет	Хом'як Тетяна Вал...	07.11.23, 11:32:25	—	—
...	Хом'як Тетяна Вал...	23.08.22, 19:19:10	—	—
р данных	Хом'як Тетяна Вал...	23.08.22, 19:19:10	Н/Д	—
Информационная ...	Хом'як Тетяна Вал...	—	—	—
Поток данных 2-г...	Хом'як Тетяна Вал...	07.11.23, 11:28:54	Н/Д	—
Набор данных (по ...	Хом'як Тетяна Вал...	06.11.23, 22:25:55	Н/Д	—
Хранилище	Хом'як Тетяна Вал...	—	Н/Д	—
Набор данных	Хом'як Тетяна Вал...	07.11.23, 09:55:27	Н/Д	—



Дистанційна освіта НТУДП / Di... x Головна - OneDrive x Power BI x +

app.powerbi.com/groups/d1b0b398-0182-4507-9fff-8e181420d630/quickcreate/dataset?experience=power-bi

Python | PyCharm Дистанційна освіта... Курс робоча пошта Список телефонів... Start | Learn Вступ до курсу мо... Системний аналіз |...

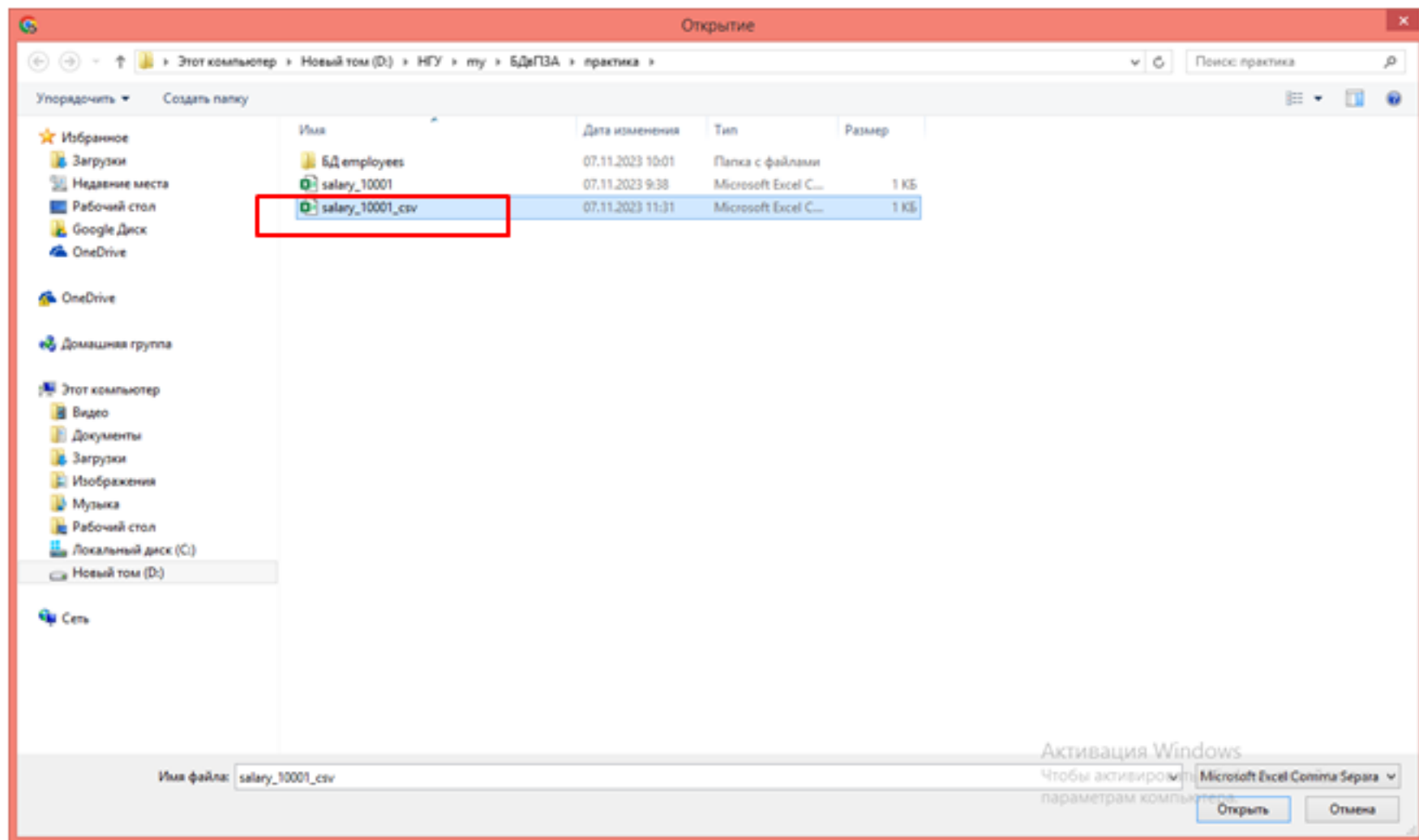
Power BI My workspace Поиск Пробная версия: Осталось 59 дн.

### Добавьте данных, чтобы начать создавать отчет

Excel CSV Вставьте данные или введите их вручную

Не видите источник, который ищете? [Загрузка настольного приложения](#).





## 6 крок (створення звіту)

The screenshot shows the Microsoft Power BI 'My workspace' interface. The 'Создать' (Create) button is highlighted with a red box, and its dropdown menu is open, showing options like 'Отчет на страницу', 'Система показателей', 'Панель мониторинга', 'Набор данных', 'Потоковый набор данных', 'Поток данных потоковой передачи', and 'Дополнительные параметры'. A table of workspace items is visible in the background.

Тип	Владелец	Обновлено	Следующее обновление	Подтвержде
Отчет	My workspace	07.11.23. 12:30:03	—	—
Отчет	My workspace	23.08.22. 19:19:10	—	—
Набор данных	My workspace	23.08.22. 19:19:10	Н/Д	—
Информационная ...	My workspace	—	—	—
Поток данных 2-г...	Хом'як Тетяна Вал...	07.11.23. 11:28:54	Н/Д	—
Набор данных (по ...	My workspace	06.11.23. 22:25:55	Н/Д	—
Хранилище	Хом'як Тетяна Вал...	—	Н/Д	—
Набор данных	My workspace	07.11.23. 09:55:27	Н/Д	—



Дистанційна освіта НТУДП / D: x Головна – OneDrive x Power BI x +

app.powerbi.com/groups/d1b0b398-0182-4507-9fff-8e181420d630/quickcreate?experience=power-bi

Python | PyCharm Дистанційна освіта... Курс робоча пошта Список телефоні... Start | Learn Вступ до курсу мо... Системний аналіз |... » | Другие заклад

Power BI My workspace Поиск Пробная версия: Осталось 59 дн.

### Добавьте данных, чтобы начать создавать отчет

Вставьте данные или введите их вручную

Выберите опубликованный набор данных.

Выберите опубликованный набор данных.

Не видите источник, который ищете? [Загрузка настольного приложения.](#)





Дистанційна освіта НТУДП / D: x | Головна – OneDrive x | Power BI x +

app.powerbi.com/groups/d1b0b398-0182-4507-9fff-8e181420d630/quickcreate?experience=power-bi

Python | PyCharm | Дистанційна освіта... | Курс | робоча пошта | Список телефонів... | Start | Learn | Вступ до курсу мо...

Центр данных OneLake

### Выберите набор данных для использования в отчете

Все | Мои данные | Одобрено в вашей организации

Обозреватель

Имя	Владелец	Обновлено	Расположение
salary_10001_csv	Хом'як Тетяна Вале...	07.11.23, 12:30:03	Моя рабочая область
salary_10001	Хом'як Тетяна Вале...	07.11.23, 11:29:23	Моя рабочая область
salary_10001	Хом'як Тетяна Вале...	07.11.23, 09:55:27	Моя рабочая область
BI_importExcel	Хом'як Тетяна Вале...	23.08.22, 19:19:10	Моя рабочая область
DataflowsStagingWarehouse	Хом'як Тетяна Вале...	06.11.23, 22:26:49	Моя рабочая область
DataflowsStagingLakehouse	Хом'як Тетяна Вале...	06.11.23, 22:26:48	Моя рабочая область
my1	Хом'як Тетяна Вале...	06.11.23, 22:25:55	Моя рабочая область



# 7 крок (побудова візуального елемента)

Power BI My workspace

Поиск

Пробная версия: Осталось 59 дн.

Главная Создать Обзор Центр данных... Приложения Рабочие области My workspace

Файл Вид Представление чтения Макет для мобильных приложений Открыть модель данных

Создание визуальных элементов по вашим данным  
Выберите или перетащите поля из области Данные на холст отчета.

Фильтры

Поиск

Фильтры на этой странице ...  
Добавьте сюда поля с дан...

Фильтры на всех страницах ...  
Добавьте сюда поля с дан...

Визуализации

Создать визуальный элемент

Данные

Поиск

salary\_10001\_csv

- emp\_no
- from\_date
- from\_Month
- from\_Year
- salary
- to\_date
- to\_Month
- to\_Year

Значения

Добавьте сюда поля с дан...

Детализация

Несколько отчетов

Сохранить все фильтры

Добавьте сюда поля дета...

Страница 1

Дистанційна освіта НТУДП / D... БДАПЗА - OneDrive Power BI

app.powerbi.com/groups/me/datasets/6453b64c-eeea-47bd-8d9f-c44671291108?experience=power-bi

Python | PyChartm Дистанційна освіта... Курс робоча пошта Список телефонів... Start | Learn Вступ до курсу мо... Системний аналіз |... Другие закладки

Power BI Моя рабочая область Поиск Пробная версия: Осталось 59 дн.

Файл Представление чтения Макет для мобильных приложений Открыть модель данных

Назад к отчету СУММА SALARY ПО FROM\_DATE

from_date	Salary (тыс.)	Percentage (%)
1986-06-26	88,96	6,94%
1987-06-26	60,12	4,69%
1988-06-25	62,1	4,85%
1989-06-25	66,07	5,16%
1990-06-25	66,6	5,2%
1991-06-25	66,96	5,22%
1992-06-24	71,05	5,54%
1993-06-24	74,33	5,8%
1994-06-24	75,29	5,87%
1995-06-24	75,99	5,93%
1996-06-23	76,88	6%
1997-06-23	80,01	6,24%
1998-06-23	81,03	6,32%
1999-06-23	81,1	6,33%
2000-06-22	84,92	6,63%
2001-06-22	85,11	6,64%
2002-06-22	85,1	6,64%

from\_date

- 1986-06-26
- 1987-06-26
- 1988-06-25
- 1989-06-25
- 1990-06-25
- 1991-06-25
- 1992-06-24
- 1993-06-24
- 1994-06-24
- 1995-06-24
- 1996-06-23
- 1997-06-23
- 1998-06-23
- 1999-06-23
- 2000-06-22
- 2001-06-22
- 2002-06-22

Фильтры

Визуализации Создать визуальный элемент

Данные salary\_10001\_csv

- emp\_no
- from\_date
- from\_Month
- from\_Year
- salary
- to\_date
- to\_Month
- to\_Year

Подробнее

Добавьте сюда поля с дан...

Подсказки

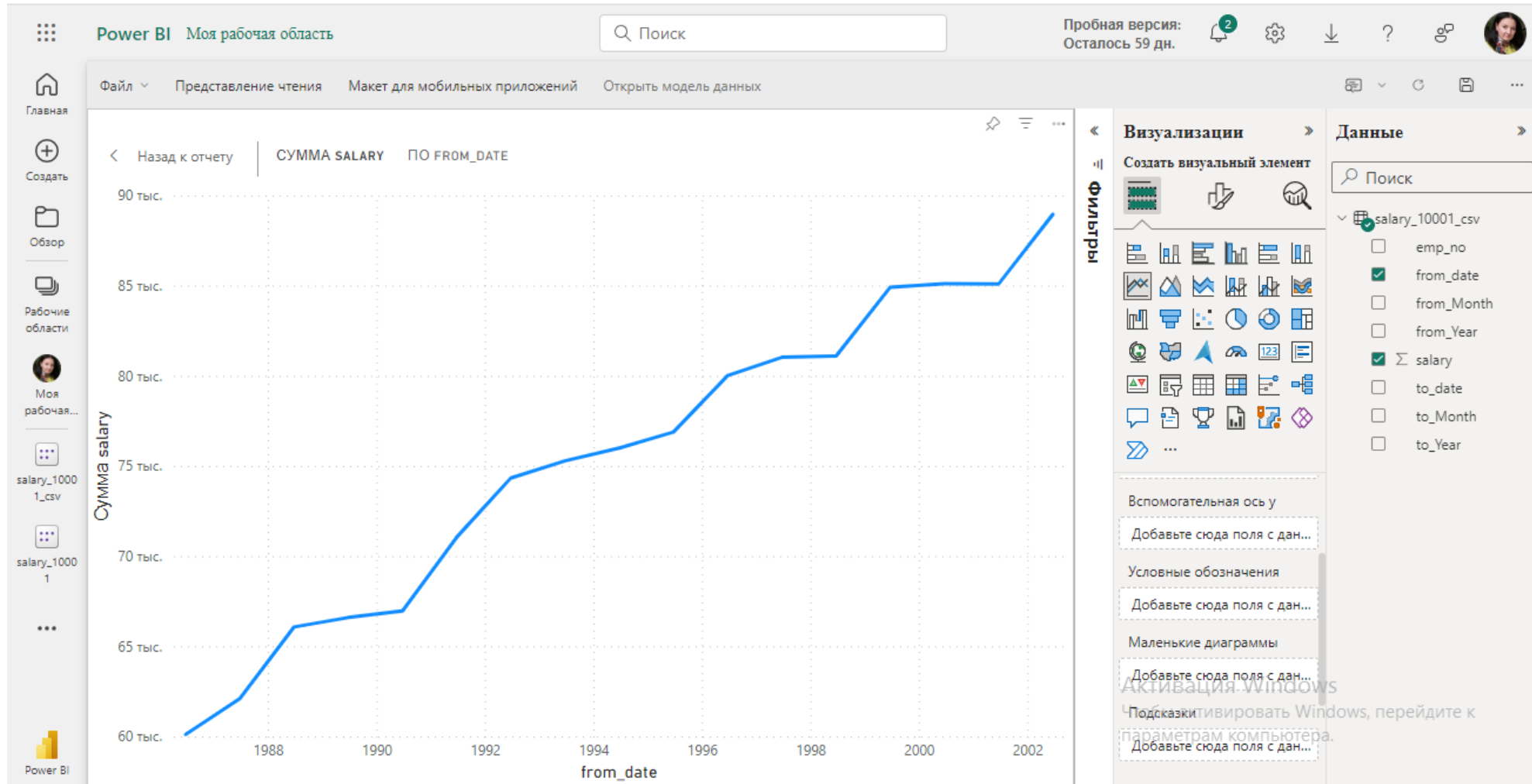
Добавьте сюда поля с дан...

Детализация

Несколько отчетов

Сохранить все параметры

Активация Windows



Дистанційна освіта НТУДП / Dis x БДвПЗА – OneDrive x Power BI x +

app.powerbi.com/groups/me/datasets/6453b64c-cccc-47bd-8d9f-c44671291108?experience=power-bi

Python | PyCharm Дистанційна освіта... Курс робоча пошта Список телефонів... Start | Learn Вступ до курсу мо... Системний аналіз |... >> | Другие закладки

Power BI Моя рабочая область Поиск Пробная версия: Осталось 59 дн.

Главная Файл Представление чтения Макет для мобильных приложений Открыть модель данных

Назад к отчету СУММА SALARY ПО FROM\_DATE

from_date	Сумма salary (тыс.)
1988	60
1989	62
1990	66
1991	67
1992	70
1993	73
1994	74
1995	75
1996	78
1997	80
1998	81
1999	81
2000	85
2001	85
2002	88

Сумма salary

from\_date

Визуализации Создать визуальный элемент Фильтры

Ось Y Сумма salary

Условные обозначения

Добавьте сюда поля с дан...

Маленькие диаграммы

Добавьте сюда поля с дан...

Данные

Поиск

salary\_10001\_csv

- emp\_no
- from\_date
- from\_Month
- from\_Year
- Σ salary
- to\_date
- to\_Month
- to\_Year

Подсказки Чтобы активировать Windows, перейдите к параметрам компьютера. Добавьте сюда поля с дан...



Дистанційна освіта НТУДП / Dis x Головна – OneDrive x Power BI x +

app.powerbi.com/groups/d1b0b398-0182-4507-9fff-8e181420d630/datasets/6453b64c-cccc-47bd-8d9f-c44671291108?experience=power-bi

Python | PyCharm Дистанційна освіта... Курс робоча пошта Список телефонів... Start | Learn Вступ до курсу мо... Системний аналіз |... Другие закладки

Power BI My workspace Поиск Пробная версия: Осталось 59 дн.

Файл Представление чтения Макет для мобильных приложений Открыть модель данных

Сумма salary по from\_Year

Сумма salary по from\_date

Сумма salary по from\_date, from\_date и from\_date

Фильтры

Поиск

Фильтры для этого визуальн... ..

Сумма salary (все)

from\_date (все)

Добавьте сюда поля с дан...

Фильтры на этой странице ...

Добавьте сюда поля с дан...

Фильтры на всех страницах ...

Добавьте сюда поля с дан...

Визуализации

Создать визуальный элемент

Ось X

from\_date

Ось Y

Сумма salary

Условные обозначения

Добавьте сюда поля с дан...

Данные

Поиск

salary\_10001\_csv

- emp\_no
- from\_date
- from\_Month
- from\_Year
- salary
- to\_date
- to\_Month
- to\_Year

Страница 1



## Завдання до теми 6

1. Скласти перелік усіх працівників із бази даних Employees, зарплата яких перевищує 150 000 доларів США на рік.
2. Зробити візуалізацію зарплати працівників за допомогою додатка Power BI.

## Список рекомендованих джерел до теми 6

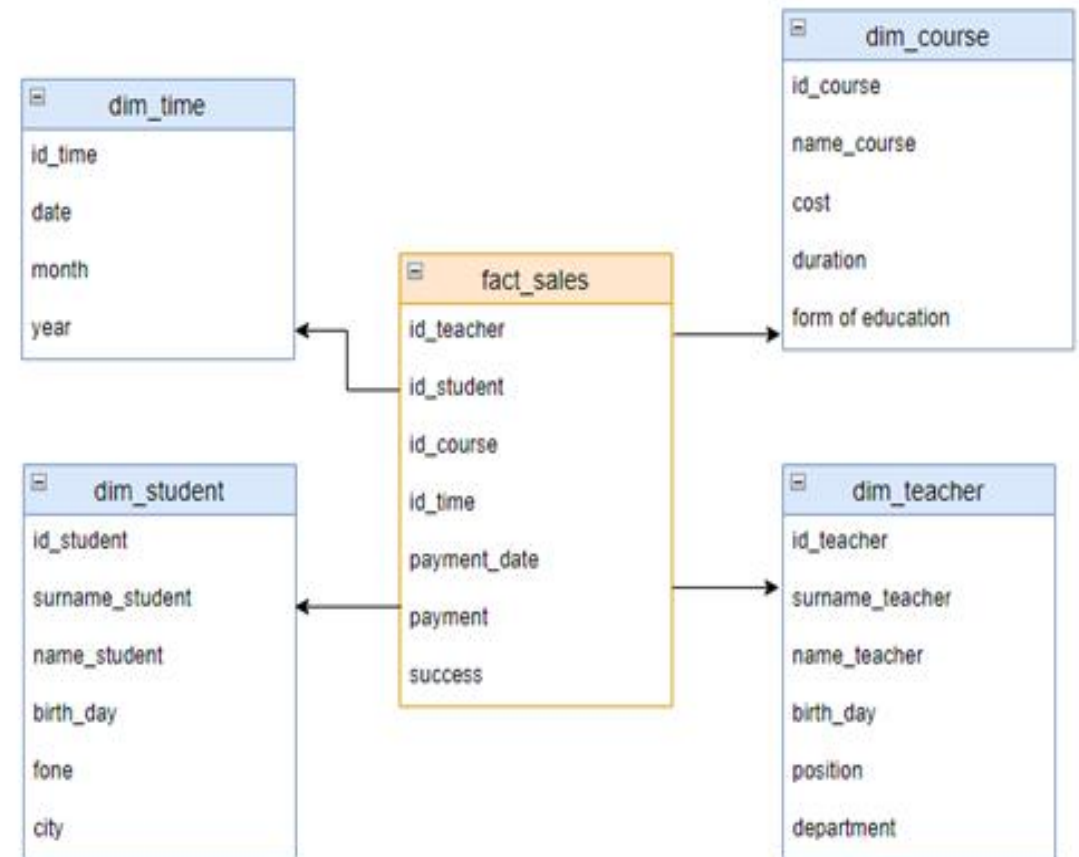
1. Reed M. SQL: 3 books in 1 – The Ultimate Beginner, Intermediate & Expert Guides To Master SQL Programming Quickly with Practical Exercises, London, England : Amazon, 2022, 348 p.
2. Korotkevitch D. SQL Server Advanced Troubleshooting and Performance Tuning. Best Practices and Techniques, USA : O'Reilly, 2022, 500 p.
3. Документація Microsoft Power BI. Режим доступу: <https://learn.microsoft.com/ru-ru/power-bi/fundamentals/power-bi-overview>.
4. Microsoft Power BI Cookbook: Creating Business Intelligence Solutions of Analytical Data Models, Reports, and Dashboards. Birmingham : Packt Publishing Ltd, 2017, 802 p.



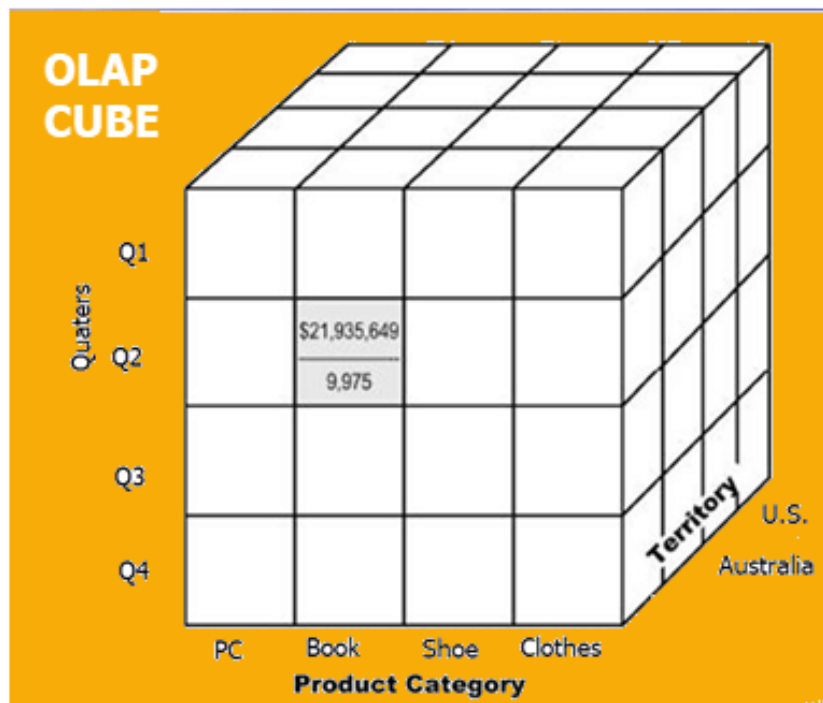
## Тема 7

# Проектування сховищ даних і створення OLAP-куба для розв'язування професійних аналітичних задач

**Сховище даних** (*data warehouse*) - предметно-орієнтований, інтегрований, незмінний, що підтримує хронологію, набір даних, організований для цілей підтримки прийняття рішень. Основна мета створення data warehouse в тому, щоб зробити усі значимі для управління бізнесом дані доступними в стандартизованій формі, придатними для аналізу та отримання необхідних звітів. Для досягнення цього потрібно отримати дані із існуючих внутрішніх та зовнішніх, доступних для комп'ютера, джерел.







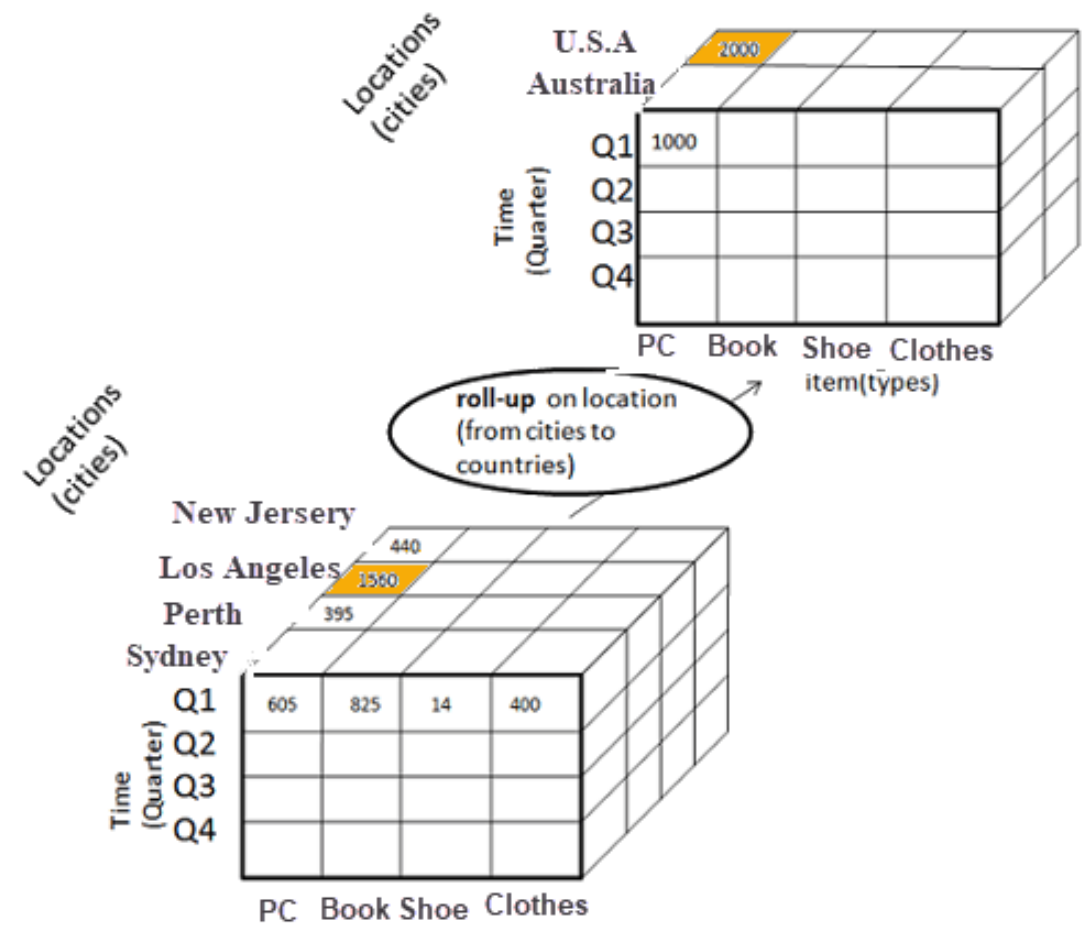
Куб OLAP

**OLAP** – аббревіатура від англійського On-Line Analytical Processing – це технологія обробки інформації, що дозволяє швидко отримувати відповіді на багатовимірні аналітичні запити. OLAP є частиною такого ширшого поняття, як бізнес-аналітика, що також включає такі дисципліни як реляційна звітність та добування даних. Служить для підготовки бізнес-звітів з продажів, маркетингу, в цілях управління, для прогнозування, фінансової звітності та в схожих областях.

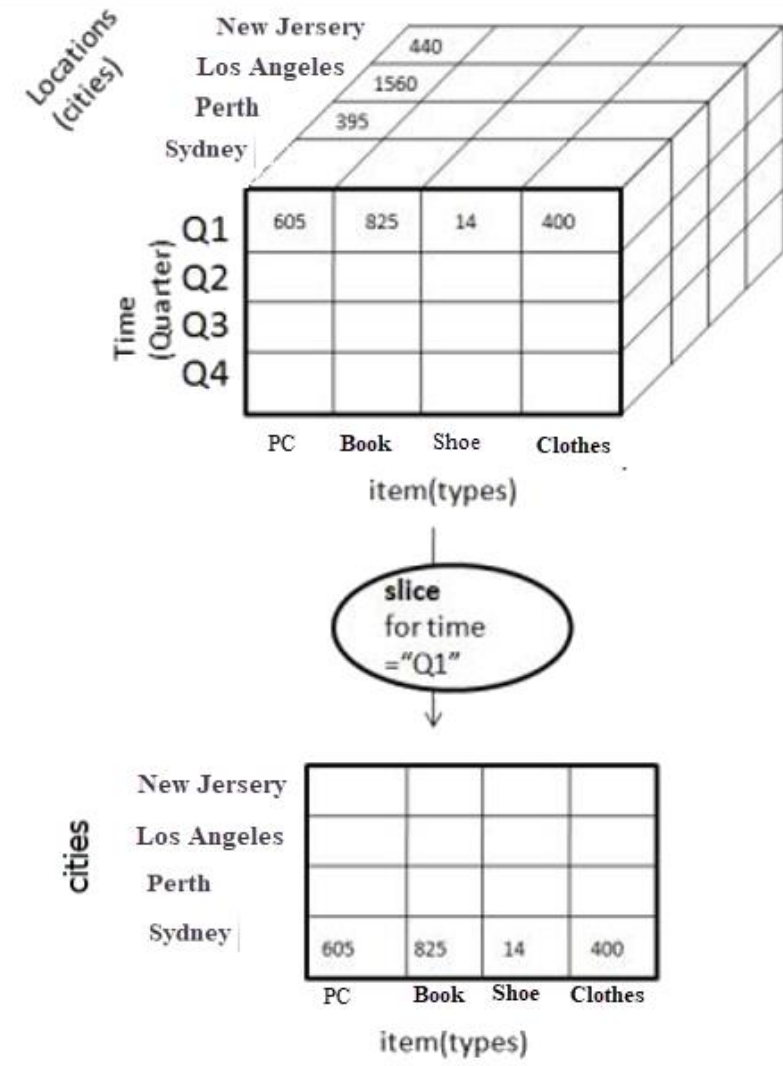
Ядром будь-якої OLAP-системи є ідея OLAP-куба (багатовимірний куб, або гіперкуб). OLAP-структура, створена з робочих даних, називається OLAP-кубом. Він складається з чисельних фактів (розмірів), розподілених за вимірами. Зазвичай куб створюється за допомогою з'єднання таблиць із застосуванням схеми «зірка», або схеми «сніжинка». В центрі «зірки» знаходиться таблиця, яка містить ключові факти, за якими робляться запити. Множинні таблиці з вимірами приєднані до таблиці фактів. Ці таблиці показують, як можуть аналізуватися агреговані реляційні дані. Кількість можливих агрегацій визначається кількістю способів, якими первинні дані можуть бути ієрархічно відображені.



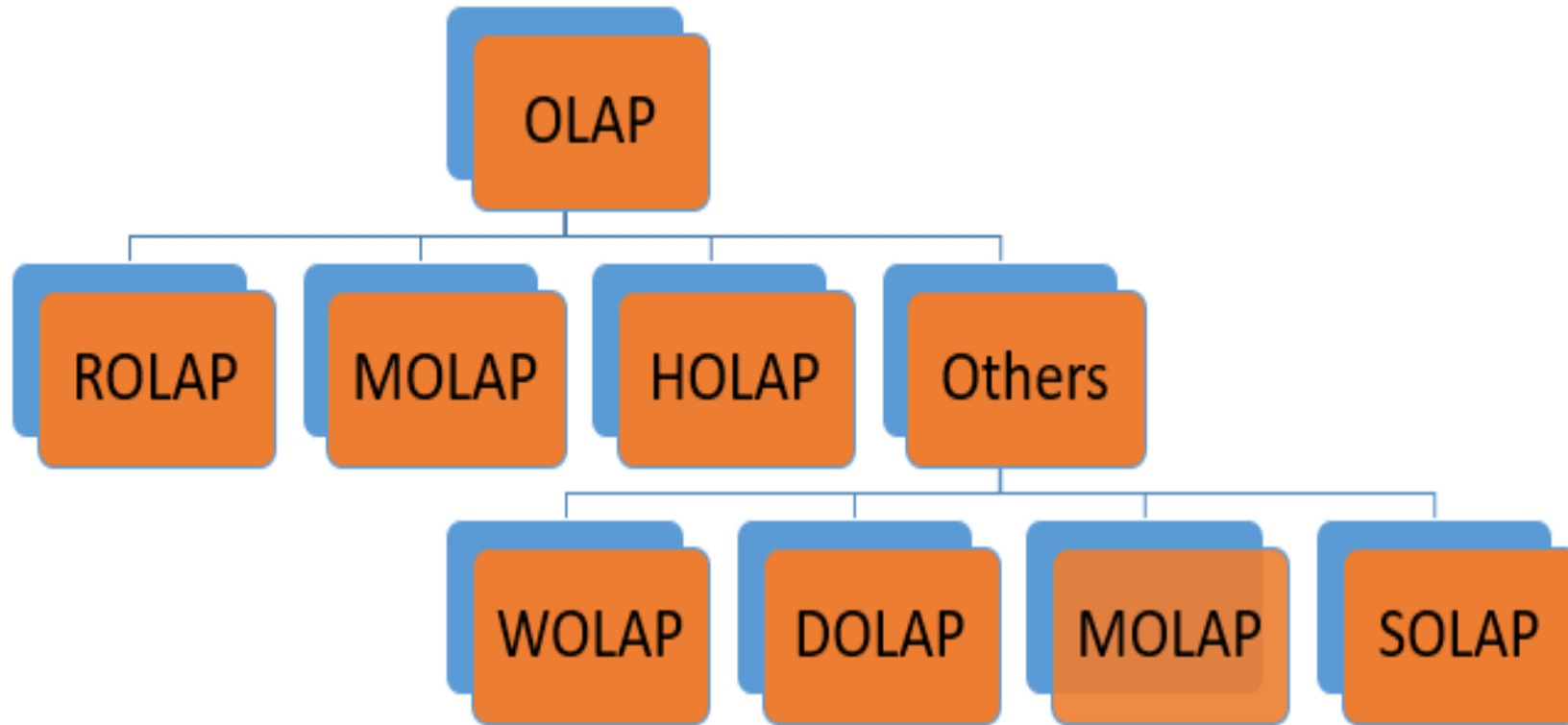
# Згорання в OLAP-куб



# Переріз OLAP-кубу



# Типы систем OLAP



## Переваги OLAP

- OLAP – це платформа для всіх типів бізнесу, включаючи планування, бюджетування, звітування та аналіз;
- інформація та обчислення є узгодженими в кубі OLAP – це важлива перевага;
- швидко створювати та аналізувати сценарії «що, якщо»;
- простий пошук у базі даних OLAP за широкими або конкретними термінами;
- OLAP надає будівельні блоки для інструментів бізнес-моделювання, інструментів інтелектуального аналізу даних, інструментів звітування про продуктивність;
- дозволяє користувачам розрізати дані за різними параметрами, розмірами та фільтрами;
- зручно використовувати для аналізу часових рядів;
- потужна система онлайн-аналітичного процесу візуалізації, яка забезпечує швидший час відгуку.

## Недоліки OLAP

- OLAP вимагає організації даних у вигляді зірки або сніжинки. Ці схеми складні у впровадженні та адмініструванні;
- не можна мати велику кількість вимірів в одному кубі OLAP;
- доступ до даних транзакцій за допомогою системи OLAP неможливий;
- будь-які зміни в кубі OLAP потребують повного оновлення куба. Це трудомісткий процес.



## Завдання до теми 7

1. Розробити сховище на основі бази даних Employees.
2. На основі побудованого сховища даних спроектувати OLAP-куб (самостійно визначити його виміри).

## Список рекомендованих джерел до теми 7

1. Gerardus B. OLAP cube. Toronto : 5STARCOOKS, 2022, 307 p.
2. Reed M. SQL: 3 books in 1 – The Ultimate Beginner, Intermediate & Expert Guides To Master SQL Programming Quickly with Practical Exercises, London, England : Amazon, 2022, 348 p.
3. Korotkevitch D. SQL Server Advanced Troubleshooting and Performance Tuning. Best Practices and Techniques, USA : O'Reilly Media, 2022, 500 p.
4. Ситник Н. В. Організація баз та сховищ даних [Електронний ресурс] : практикум. Київ : Київський нац. екон. ун-т, 2017, 148 с.
5. Хом'як Т.В. Аналіз та моделювання діяльності додаткових курсів на кафедрі: дипл. проект магістра. Запоріжжя : Запорізька політехніка, 2023, 87 с.



Навчальне видання

- Хом'як Тетяна Валеріївна
- Хабарлак Костянтин Сергійович
- Гаранжа Дмитро Миколайович

□

□ **БАЗИ ДАНИХ У ПРОФЕСІЙНИХ ЗАДАЧАХ АНАЛІТИКИ**

Навчальний наочний посібник

Видано в авторській редакції.

Підписано до видання 27.06.2024.

Електронний ресурс. Авт. арк. 8,4.

Підготовлено до видання

в Національному технічному університеті «Дніпровська політехніка».

Свідоцтво про внесення до Державного реєстру ДК № 1842 від 11.06.2004.

49005, м. Дніпро, просп. Д. Яворницького, 19

# Відомості про авторів

**Хом'як Тетяна Валеріївна** – кандидат фізико-математичних наук, доцент кафедри системного аналізу та управління Національного технічного університету «Дніпровська політехніка». Наукові інтереси пов'язані з розробкою клієнт-сервісних інформаційних систем, аналізом та візуалізацією даних, моделюванням складних систем, систем підтримки і прийняття рішень, розв'язуванням професійних задач аналітики. Автор понад 100 публікацій наукових та навчально-методичних праць, серед яких навчальні посібники, понад 60 наукових робіт у фахових вітчизняних і зарубіжних періодичних виданнях, у т. ч. включених до наукометричної бази Scopus, тез доповідей на наукових конференціях та міжнародних конгресах. З 2019 року член громадської організації «Системні дослідження».

**Хабарлак Костянтин Сергійович** – доктор філософії за спеціальністю «Комп'ютерні науки». Доцент кафедри системного аналізу та управління Національного технічного університету «Дніпровська політехніка». Наукові інтереси включають аналіз та обробку зображень, нейронні мережі, засоби зберігання, обробки та аналізу даних різної природи. Є автором понад 50 наукових публікацій та навчально-методичних праць, зокрема у фахових виданнях і тих, що внесені до наукометричних баз Scopus та Web of Science, тез доповідей на наукових конференціях.

**Гаранжа Дмитро Миколайович** – старший викладач кафедри системного аналізу та управління Національного технічного університету «Дніпровська політехніка». Наукові інтереси включають питання розробки інформаційних систем для виробничих процесів, аналізу та візуалізації даних, моделювання бізнес-процесів. Автор наукових публікацій та навчально-методичних праць, зокрема у фахових виданнях і тих, що внесені до наукометричних баз, а також тез доповідей наукових конференцій.