

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра системного аналізу та управління

БАЗИ ДАНИХ В ПРОФЕСІЙНИХ ЗАДАЧАХ АНАЛІТИКИ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня магістра
освітньо-професійної програми «Системний аналіз»
зі спеціальності 124 Системний аналіз

Дніпро
НТУ «ДП»
2024

Бази даних в професійних задачах аналітики [Електронний ресурс]: методичні рекомендації до виконання практичних робіт для здобувачів ступеня магістра спеціальності 124 Системний аналіз / уклад.: Л.С. Коряшкіна, Т.В. Хом'як, Д.М. Гаранжа, А.В. Малієнко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 47 с.

Укладачі:

Л.С. Коряшкіна, канд. фіз.-мат. наук, доц.;

Т.В. Хом'як, канд. фіз.-мат. наук, доц.;

Д.М. Гаранжа, ст. викл.;

А.В. Малієнко, канд. техн. наук, доц.

Затверджено до видання науково-методичною комісією зі спеціальності 124 Системний аналіз (протокол № 4 від 31.05.2024) за поданням кафедри системного аналізу та управління (протокол № 6 від 31.05.2024).

Наведено теоретичні відомості за темами лекцій, приклади вирішення задач, завдання до практичних робіт з критеріями їх оцінювання, список рекомендованих джерел.

Орієнтовано на активізацію навчальної діяльності здобувачів ступеня магістра спеціальності «Системний аналіз» та закріплення практичних навичок у засвоєнні дисципліни «Бази даних в професійних задачах аналітики».

Відповідальний за випуск завідувач кафедри системного аналізу та управління Т.А. Желдак, канд. техн. наук, доц.

ЗМІСТ

Вступ	4
Практична робота № 1. Імпорт бази даних з різних джерел.....	5
Практична робота № 2. Створення запитів мовою SQL.....	10
Практична робота № 3. Застосування віконних функції.....	20
Практична робота № 4. Створення візуалізації в Power BI.....	24
Практична робота № 5. Створення OLAP-кубу.....	27
Список використаних джерел.....	44

ВСТУП

Найпопулярнішим інструментом аналітиків виступають бази даних. Всі бізнес-процеси та функції зводяться до обробки та візуалізації даних, що зберігаються в базах даних. Ефективність прийняття бізнес-рішень залежить від того, наскільки якісно і правильно було відібрано та проаналізовано дані. Саме на основі аналізу даних будуються гіпотези, проводяться тестування та генеруються рішення про продукт. Таким чином, головними навичками системного аналітика є володіння мовою SQL, структурою SQL-запитів, основними аналітичними функціями SQL, а також вміння використовувати віконні функції при написанні запитів, розробляти OLAP-куби для проведення ефективного аналізу, створювати інтерактивні звіти та дашборди.

Мета дисципліни – сформувати у здобувачів навички застосування ефективних інструментальних засобів відбору за певними критеріями структурованих та неструктурованих даних для проведення аналізу та їх візуалізації, що дозволить ухвалювати необхідні рішення в різних сферах діяльності.

Під час вивчення дисципліни студенти набувають таких дисциплінарних результатів навчання:

- здійснювати завантаження БД з різних джерел даних в СУБД, здійснювати відбір даних за допомогою SQL-запитів / підзапитів за критеріями;
- застосовувати функції агрегації та сортувати дані в SQL-запитах;
- використовувати віконні функції при написанні SQL-запитів;
- здійснювати проектування сховищ даних;
- імпортувати результати SQL-запитів в Power BI з метою розробки інтерактивних звітів та дашбордів;
- здійснювати розгортання OLAP-кубу для швидкої обробки даних.

Це видання має допомогти студентам у виконанні практичних завдань з дисципліни. Методичні рекомендації включають необхідні короткі теоретичні відомості, детальні приклади виконання завдань, контрольні питання та завдання для індивідуальної роботи студентів.

Наведені теми практичних робіт мають на меті практичне закріплення теоретичних відомостей щодо написання запитів до бази даних для відбору за відповідним критерієм чи розрахунку, візуалізації отриманих даних, побудови OLAP-кубів. Розроблені завдання до бази даних мають допомогти здобувачам спеціальності 124 Системний аналіз у вирішенні професійних задач аналітики, які виникають в практичній діяльності фахівців.

Практичні навички, отримані при вивченні даної дисципліни, можуть бути використані в роботі системних аналітиків, дата-аналітиків, продуктових аналітиків.

Практична робота № 1. Імпорт бази даних з різних джерел

Мета: закріпити теоретичні знання і розвинути практичні навички з методів і підходів щодо отримання даних з різних джерел.

Теоретичні відомості

Для виконання практичних робіт з дисципліни рекомендується спочатку інсталиувати MySQL Workbench (безкоштовна версія не нижче 8) за посиланням <https://dev.mysql.com/downloads/workbench/> (рис. 1.1).

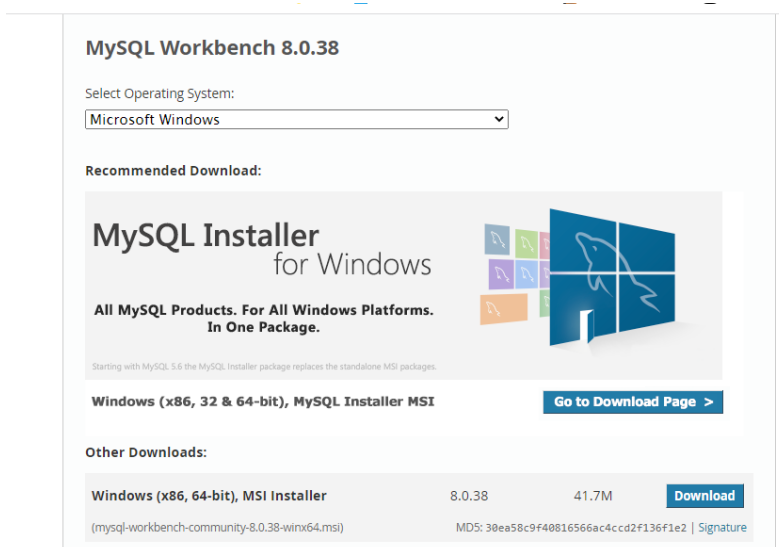


Рис. 1.1. Вікно завантаження MySQL Workbench

При інсталяції MySQL Workbench треба обрати тип – Custom (рис. 1.2) та версію серверної частини (рис. 1.3). Параметри налаштування серверної частини MySQL Server – за замовчуванням із вказанням логіна та пароля [10].

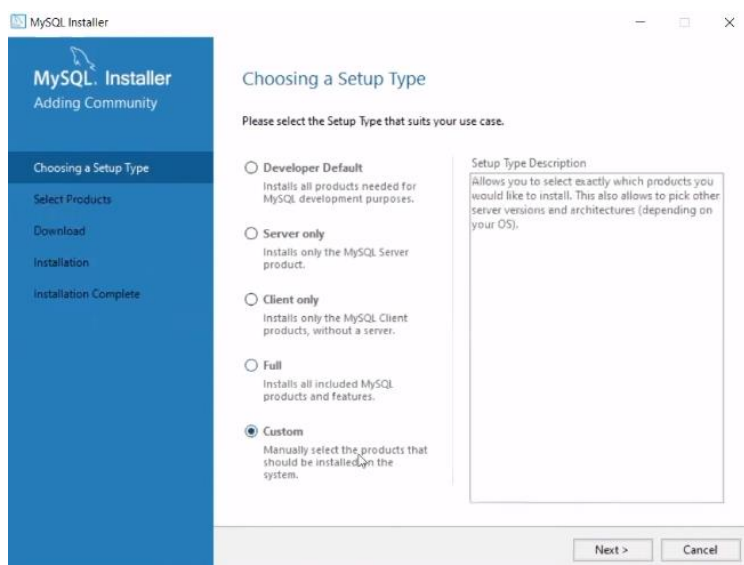


Рис. 1.2. Вікно обрання типу інсталяції MySQL Workbench

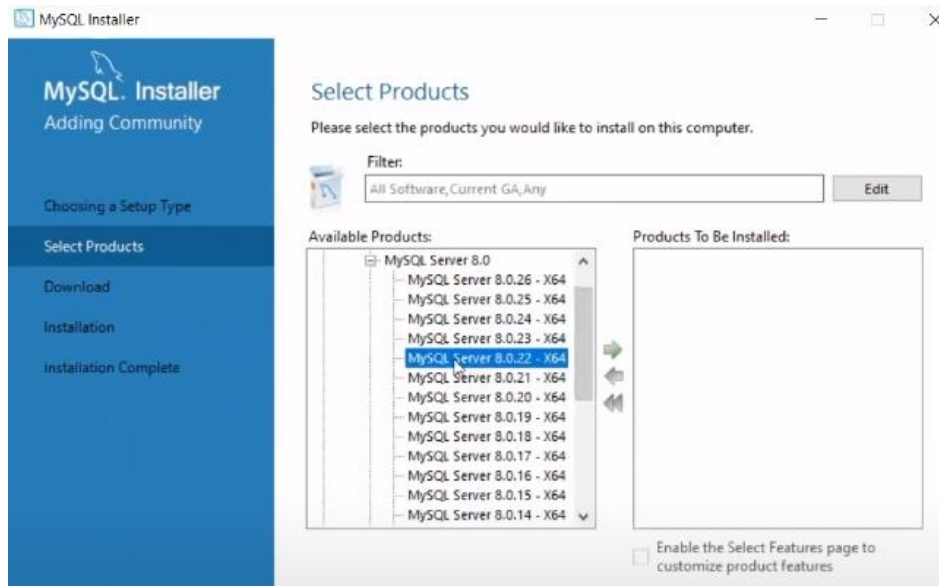


Рис. 1.3. Вікно обрання інсталяції MySQL Server

Після успішної установки серверної частини можна запустити інсталятор, де треба обрати кнопку Add (рис. 1.4) та продовжити інсталяцію клієнтської частини MySQL (версії серверної та клієнтської частин мають співпадати) (рис. 1.5).

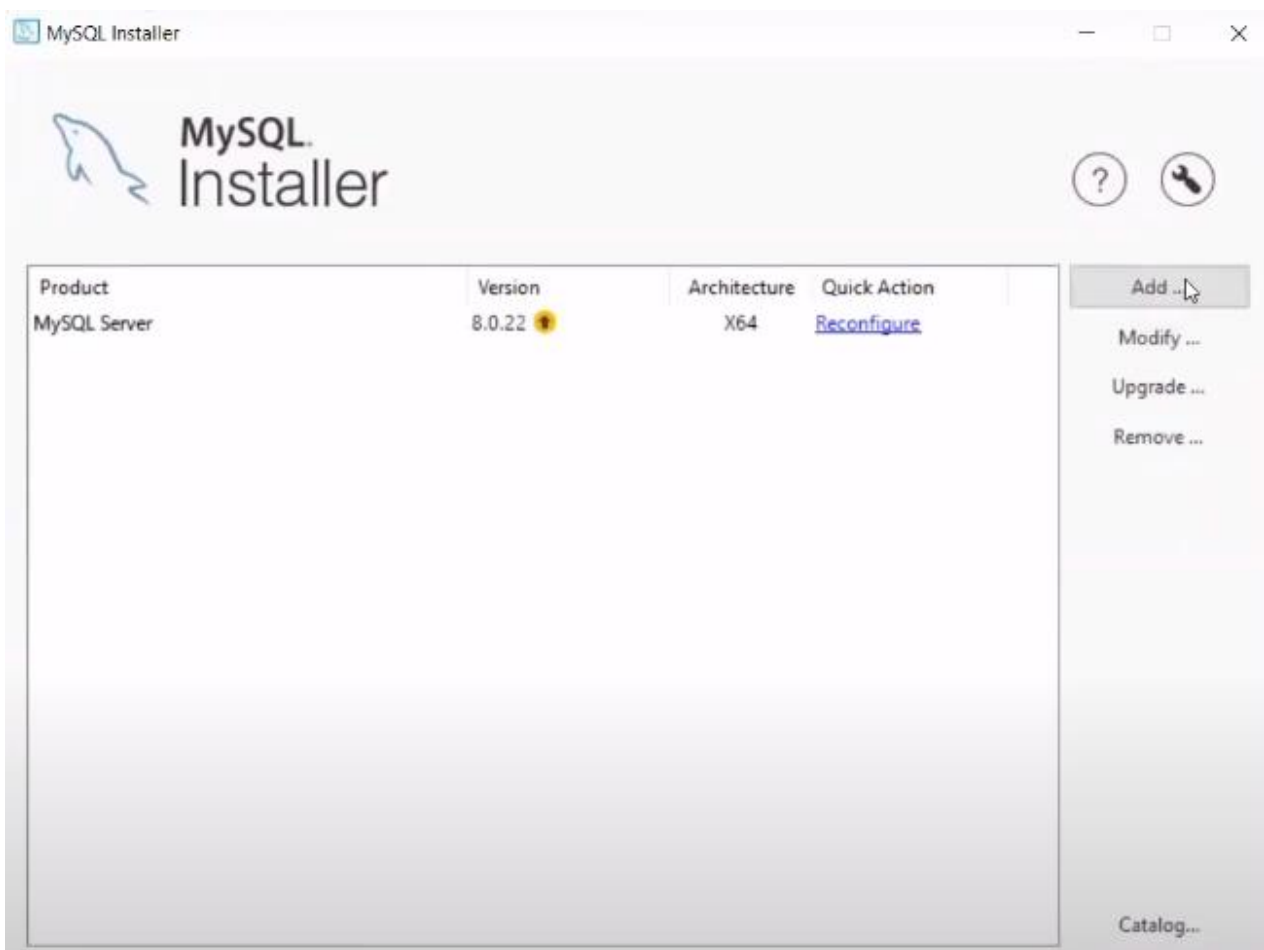


Рис. 1.4. Вікно інсталяції MySQL

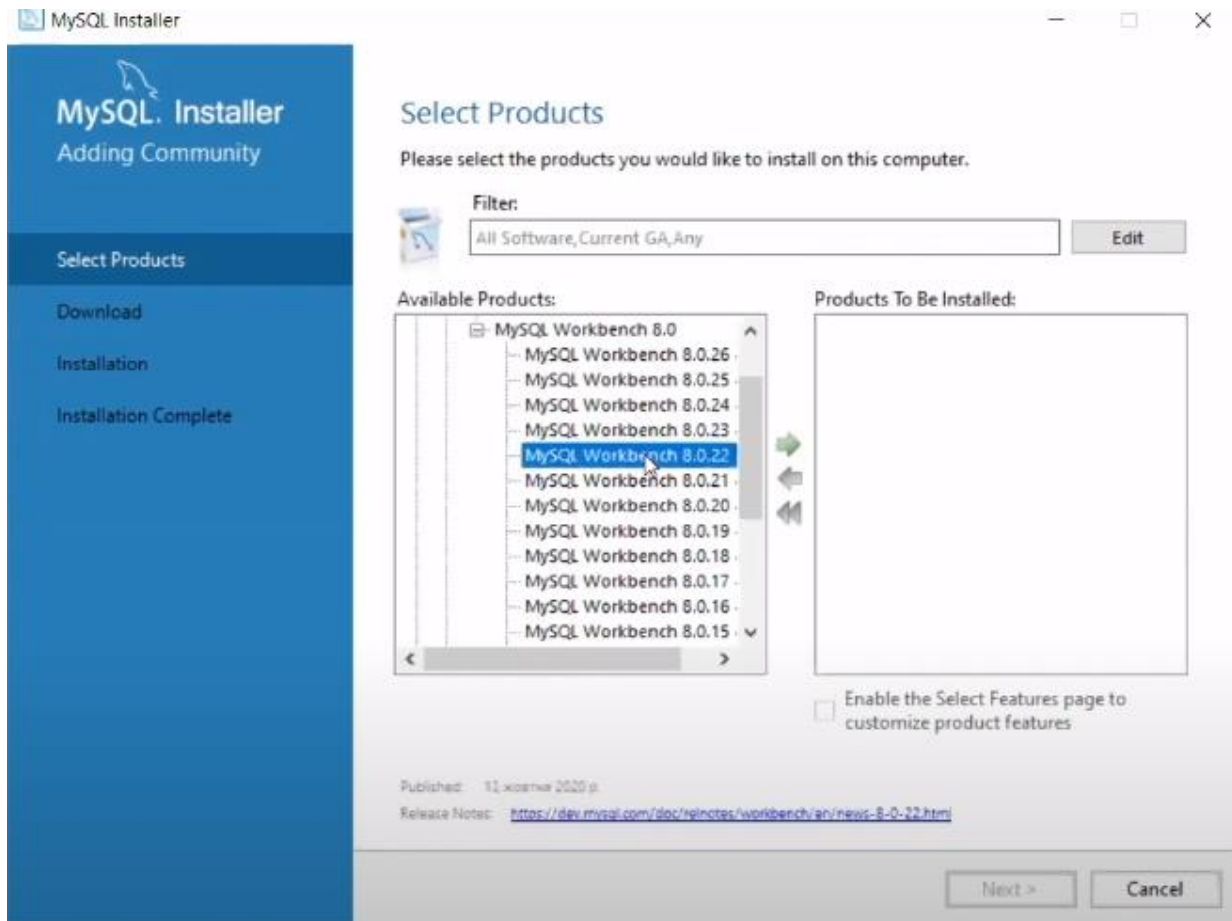


Рис. 1.5. Вікно обрання інсталяції MySQL Workbench

Після успішної інсталяції MySQL Workbench треба створити MySQL Connections для подальшої роботи (рис. 1.6).

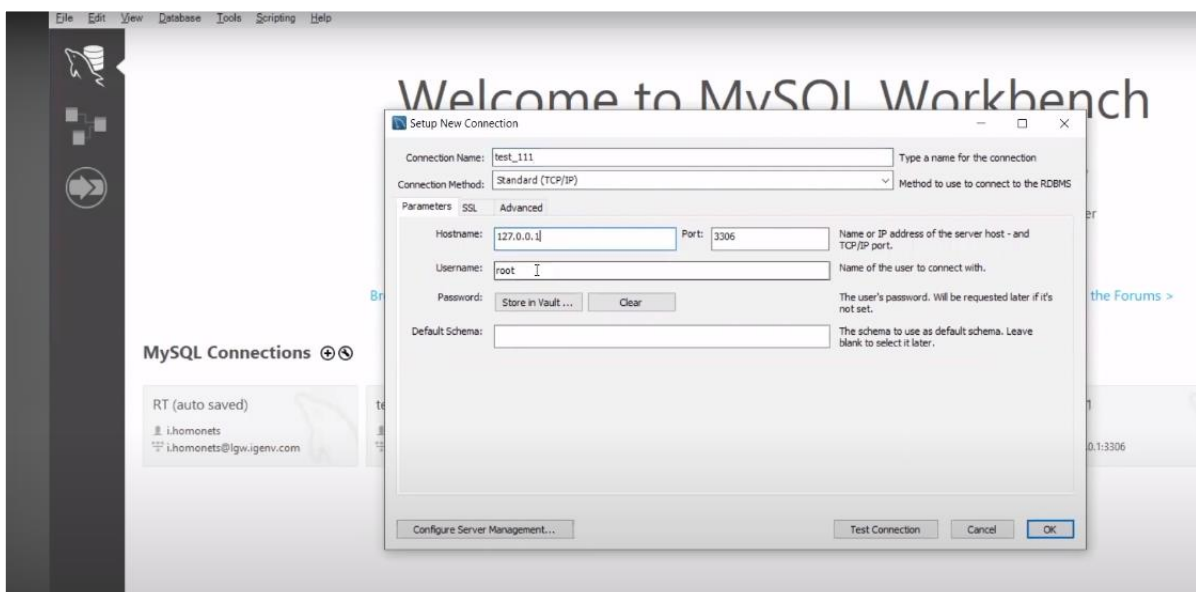


Рис. 1.6. Вікно створення MySQL Connections.

Для перевірки коректності інсталяції та створеного зв'язку між серверною та клієнтською частинами можна запустити MySQL Workbench і у вікні написання запитів ввести команду `select curdate();` і запустити на виконання, маємо отримати результат у вигляді поточної дати (рис. 1.7).

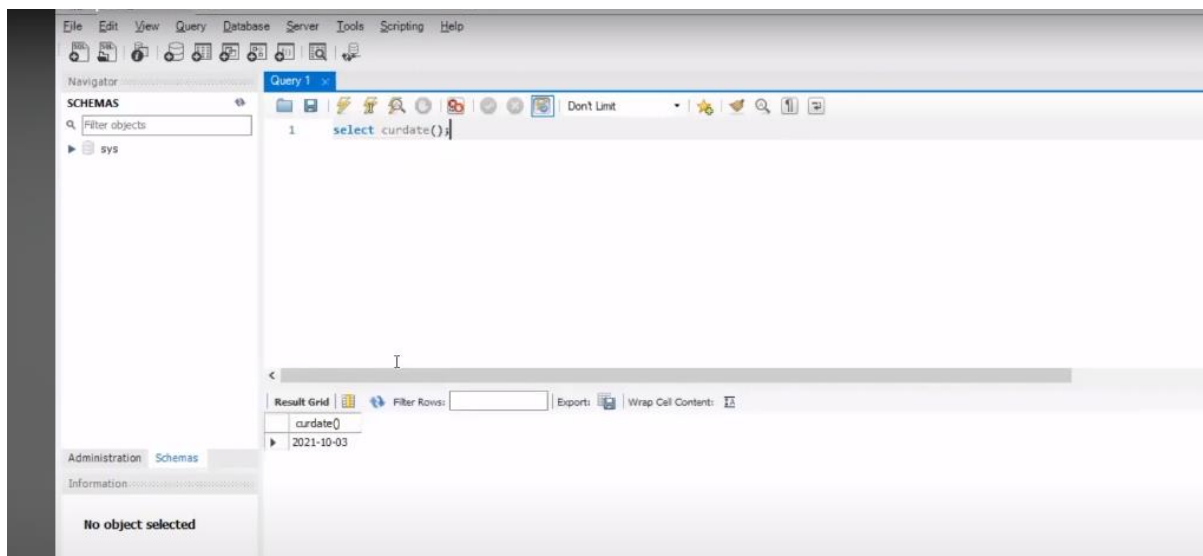


Рис. 1.7. Вікно MySQL Workbench

Для засвоєння навичок роботи з базами даних треба підключитись до існуючої БД, що знаходиться у відкритих джерелах, або створити самостійно БД. Розберемо підключення або завантаження існуючої БД в MySQL Workbench, для цього із головного меню треба обрати *File – Open SQL Script...* (Ctrl+Shift+O) та обрати файл БД (рис. 1.8).

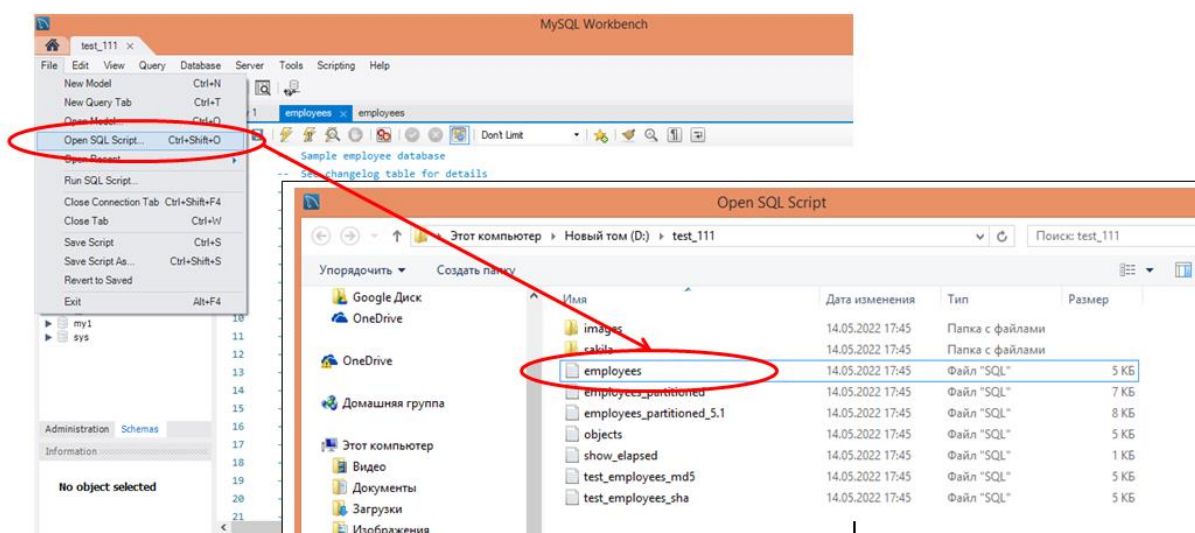


Рис. 1.8. Завантаження файлу БД в MySQL Workbench

Після успішного завантаження БД MySQL видає відповідне повідомлення і у вікні Navigator MySQL Workbench з'явиться БД зі всіма таблицями (рис. 1.9).

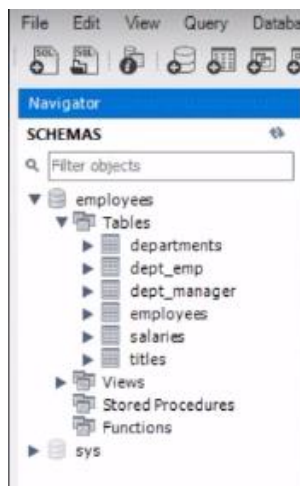


Рис. 1.9. Підключена БД Employees в MySQL Workbench

Завдання до практичної роботи 1

1. Інсталювати MySQL Workbench 8.0 і вище (безкоштовна версія).
2. Підключитися до БД Employees.
3. Ознайомитися з даною БД: таблиці, їх вміст, зв'язки між таблицями.
4. Намалювати ER-модель та реляційну схему БД.
5. Визначити primary key, foreign key, типи даних в таблицях.

Критерії оцінювання практичної роботи 1

За всі правильно виконані завдання можна отримати 100 балів (за кожне завдання по 20 балів).

Перелік рекомендованих джерел

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.

Практична робота № 2. Створення запитів мовою SQL

Мета: закріпити теоретичні знання і розвинути практичні навички написання запитів за заданими критеріями до БД.

Теоретичні відомості

Для відбору даних з БД використовується наступний оператор [1, 3, 15]:

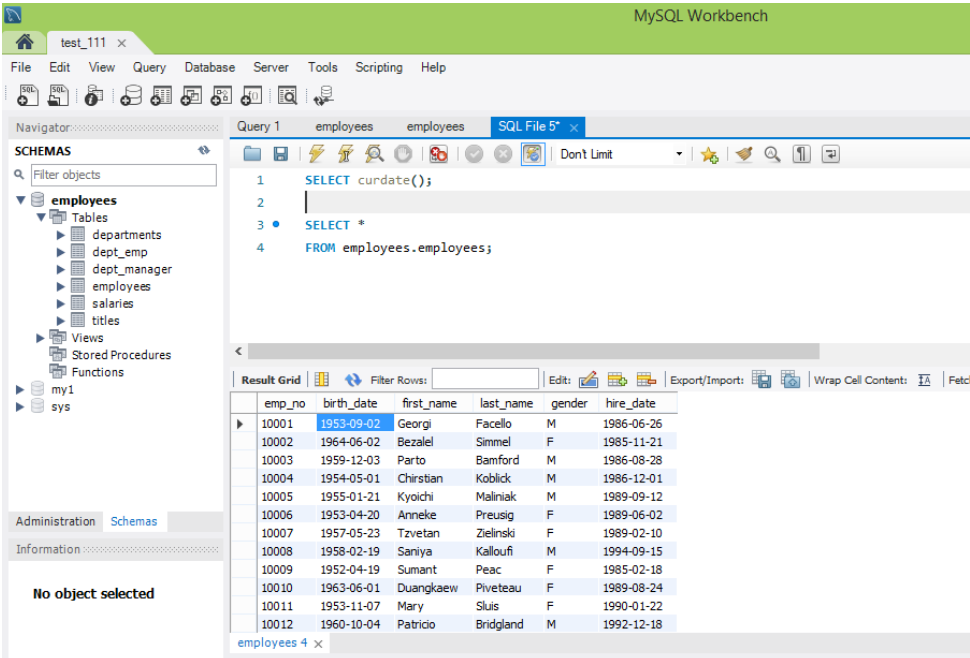
```
SELECT column1, column2, ...
FROM table_name;
```

де *column1, column2* – стовбці таблиці, які треба отримати,
table_name - таблиця, з якої відбираються дані.

Якщо з таблиці треба обрати всі стовбці, то використовують наступний синтаксис [19]:

```
SELECT * FROM table_name;
```

Приклад 2.1. Обрати всіх співробітників фірми.



The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'employees' database selected. The 'Query 1' window contains the following SQL code:

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

The 'Result Grid' at the bottom displays the results of the query. The first row shows the current date: 1953-09-02. The subsequent rows show the data from the 'employees' table:

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	Christban	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	Skuis	F	1990-01-22
10012	1960-10-04	Patricio	Bridgland	M	1992-12-18

Для задання критерію відбору використовується наступний синтаксис [19]:

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

де *condition* – умова відбору (дата, кількість, назва тощо). Якщо в умові треба поєднати декілька критеріїв, то потрібно використовувати логічні оператори AND, OR, NOT (можливо їх комбінацію):

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

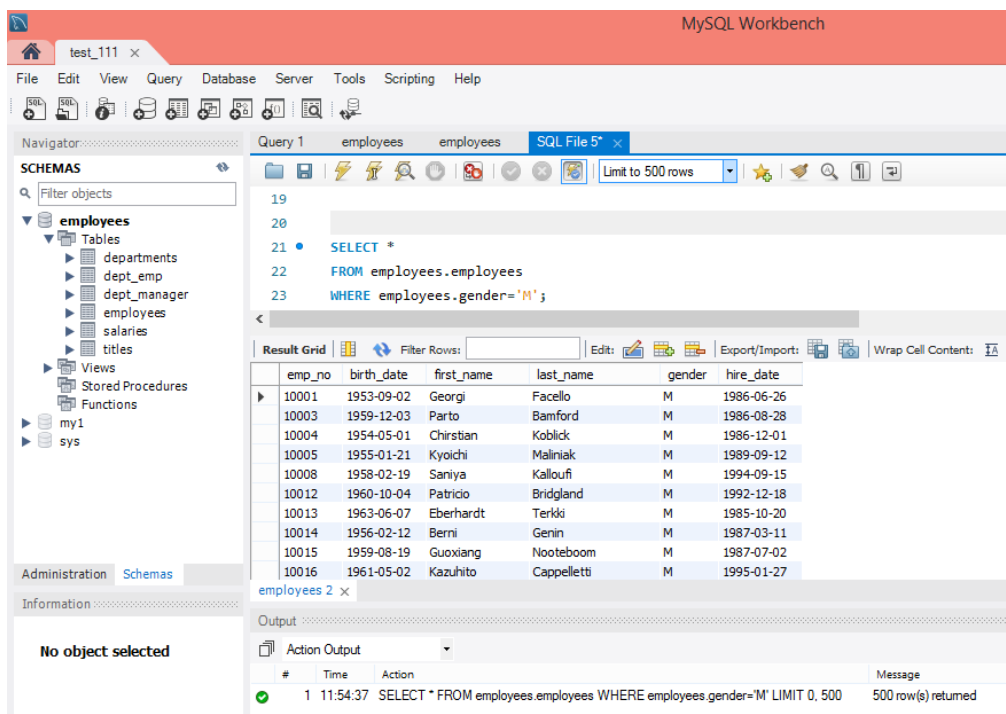
або

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

або

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

Приклад 2.2. Обрати співробітників фірми чоловічої статі.



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```
SELECT *
FROM employees.employees
WHERE employees.gender='M';
```

The result grid displays the following data:

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	Nooeteboom	M	1987-07-02
10016	1961-05-02	Kazuhiro	Cappelletti	M	1995-01-27

The output pane at the bottom shows the execution message: "1 11:54:37 SELECT * FROM employees.employees WHERE employees.gender='M' LIMIT 0, 500 500 row(s) returned".

Приклад 2.3. Обрати співробітників фірми, які були прийняті на роботу з будь-якої дати крім 28.08.1986.

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

```

34
35
36 • SELECT *
37 FROM employees.employees
38 WHERE employees.hire_date != '1986-08-28';

```

The 'Result Grid' below the query shows a list of employee records with columns: emp_no, birth_date, first_name, last_name, gender, and hire_date. The records shown are:

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	Bezalel	Simmel	F	1985-11-21
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01
10005	1955-01-21	Kyoichi	Maliniak	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

Приклад 2.4. Обрати співробітників фірми жіночої статі та прийнятих на роботу в 1987 році, у яких ім'я Parto або прізвище Plotkin.

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

```

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);

```

The 'Result Grid' below the query shows a list of employee records with columns: emp_no, birth_date, first_name, last_name, gender, and hire_date. The records shown are:

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

Для формування умови запиту зручно використовувати оператор IN при заданні множини, елементи якої поєднані умовою «або»:

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

або

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (SELECT STATEMENT);
```

Для пошуку за шаблоном використовують оператор LIKE [17, 19]:

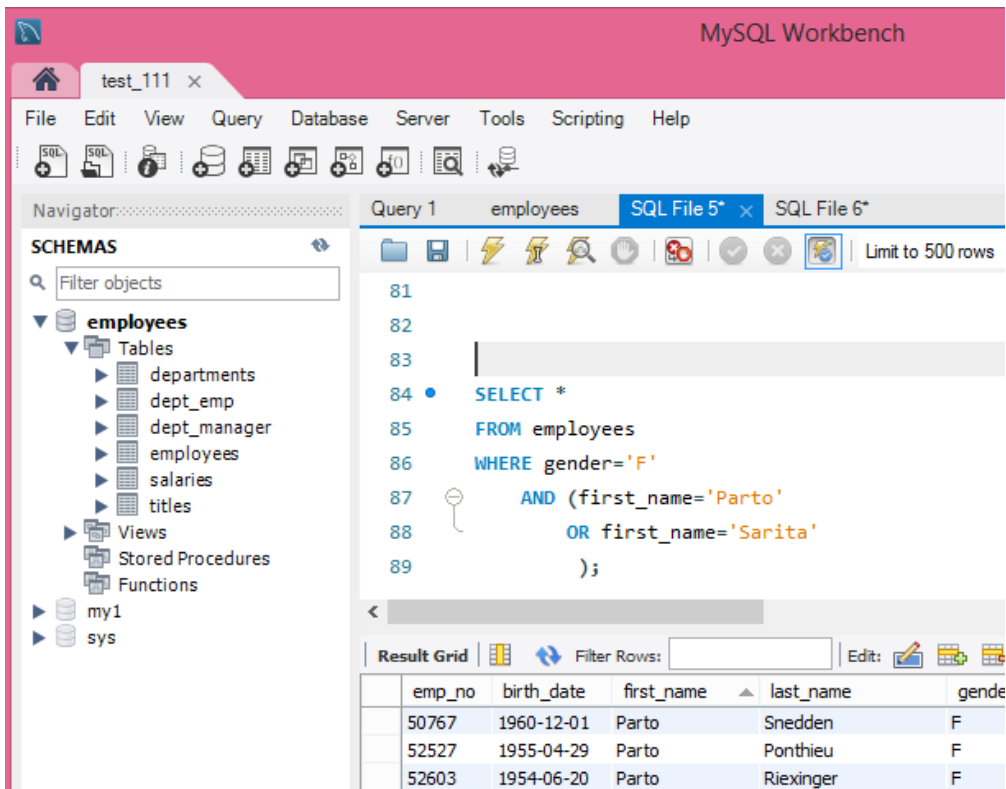
```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

де *pattern* – це заданий шаблон, в якому використовуються символи «%»-будь-яка кількість будь-яких символів та символ «_»-один будь-який символ [10].

Для задання діапазону значень в запиті зручно використовувати оператор BETWEEN:

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

Приклад 2.5. Обрати співробітників фірми жіночої статі, на ім'я Parto або Sarita.



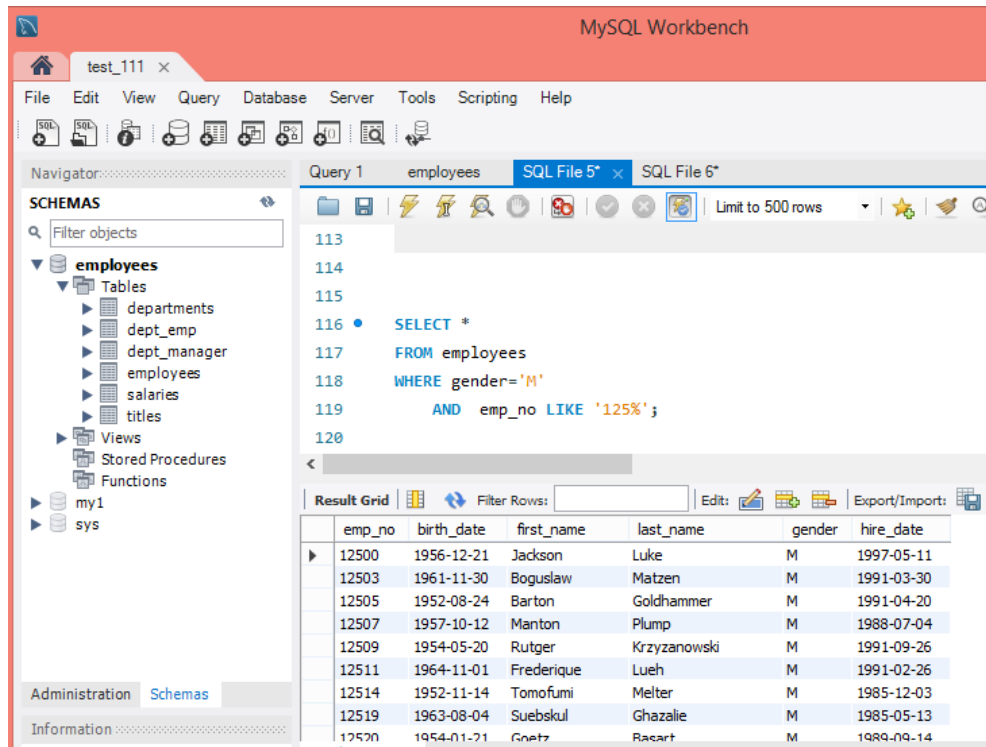
The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employees' database selected. The main window shows a SQL query in 'Query 1' with the following text:

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

Below the query editor, the 'Result Grid' is visible, showing the following data:

emp_no	birth_date	first_name	last_name	gender
50767	1960-12-01	Parto	Snedden	F
52527	1955-04-29	Parto	Ponthieu	F
52603	1954-06-20	Parto	Riexinger	F

Приклад 2.6. Обрати співробітників фірми чоловічої статі за номером, що починається з «125».



Функції агрегації в запитах дозволяють розрахувати кількість значень в стовпці – COUNT(), максимальне / мінімальне значення – MAX() / MIN(), середнє арифметичне – AVG(), суми – SUM() [10]. Синтаксис їх написання:

```

SELECT MIN(column_name)
FROM table_name
WHERE condition;

```

```

SELECT MAX(column_name)
FROM table_name
WHERE condition;

```

```

SELECT COUNT(column_name)
FROM table_name
WHERE condition;

```

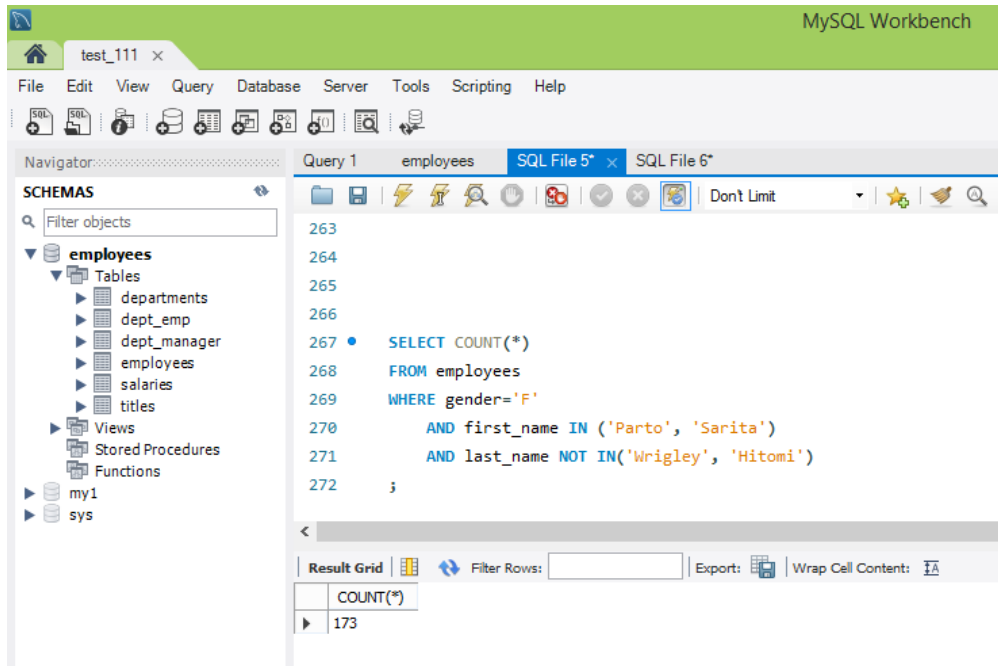
```

SELECT AVG(column_name)
FROM table_name
WHERE condition;

```

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

Приклад 2.7. Порахувати кількість співробітників фірми за відповідним критерієм відбору.



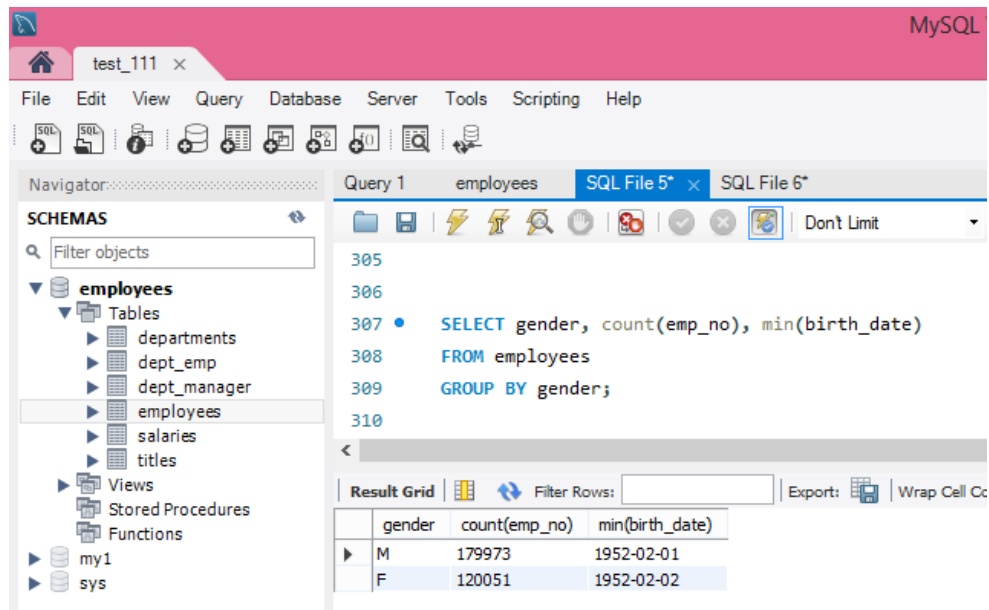
Одночасно з наведеними функціями агрегації є можливість використовувати оператор GROUP BY (), якщо треба згрупувати дані за умовою [10]. Синтаксис написання:

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

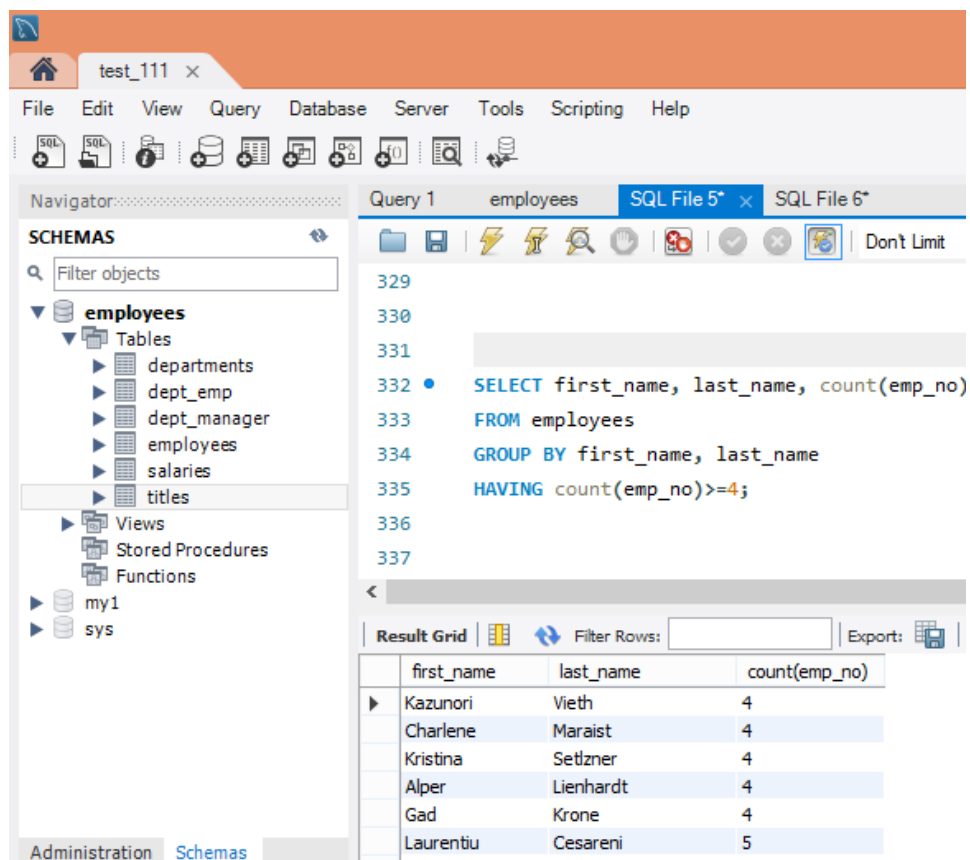
Оператор HAVING () дозволяє додати умову на розраховане значення за допомогою функції агрегації [10, 15]. Синтаксис:

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

Приклад 2.8. Порахувати кількість чоловіків та жінок на фірми, а також мінімальну дату народження за кожною статтю.



Приклад 2.9. Знайти співробітників фірми в кількості більше або дорівнює 4 з однаковими прізвищами та іменами.



У разі потреба відсортувати за спаданням або зростанням відібрані дані – використовують оператор ORDER BY() [10].Синтаксис:


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

...

Приклад 2.10. Знайти інформацію про зарплату співробітників фірми та вивести за зростанням її розміру.

The screenshot shows a SQL Enterprise Manager window titled 'test_111'. The 'Schemas' pane on the left shows a database named 'employees' with several tables, including 'salaries'. The 'Query 1' window contains the following SQL code:

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

Below the query window is a 'Result Grid' showing the results of the query. The grid has four columns: 'emp_no', 'salary', 'from_date', and 'to_date'. The data is sorted by salary in descending order.

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

У випадку, коли в запиті треба відобразити дані з декількох таблиць, необхідно використовувати оператор об'єднання таблиць JOIN (рис. 2.1). Синтаксис [10, 17]:

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

```

SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;

```

```

SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;

```

```

SELECT column_name(s)
FROM table1
CROSS JOIN table2;

```

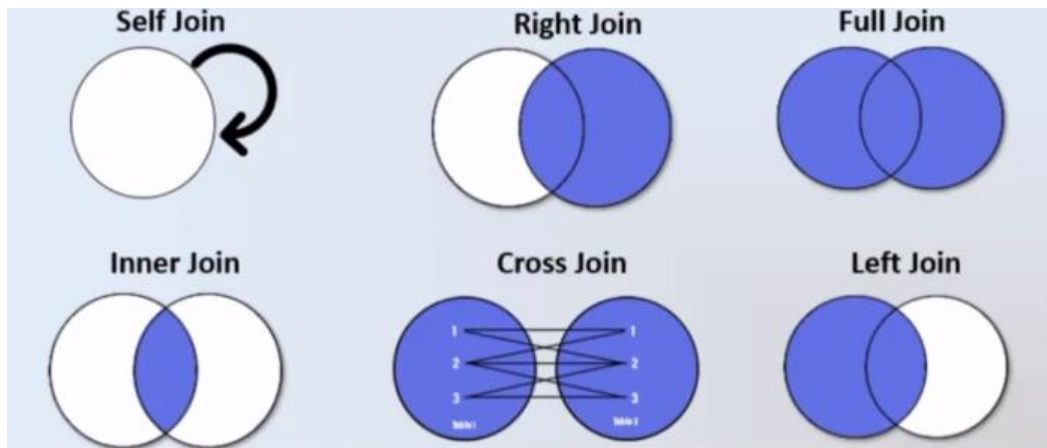


Рис. 2.1. Типи оператора JOIN і і результат

Приклад 2.11. Відібрати всіх співробітників та керівників відділів.

The screenshot shows MySQL Workbench with a query window open. The query is:

```

SELECT *
FROM employees as em
LEFT JOIN dept_manager as dm
ON em.emp_no = dm.emp_no;

```

The result grid displays the following data:

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

Завдання до практичної роботи 2

1. Отримайте список з працівниками чоловіками з іменем 'Mark'.
2. Отримайте список з працівниками жінками, хто був прийнятий на роботу 1995-11-29.
3. Отримайте список з працівниками чоловіками, хто був прийнятий на роботу 1996-06-19.
4. Отримайте список з усіма працівниками жінками з іменами Kellie та Beshir.
5. Отримайте список з працівниками жінками, хто був прийнятий на роботу 1995-11-29 та усіма співробітниками чоловіками хто був прийнятий на роботу 1996-06-19.
6. Отримайте список з працівниками жінками, хто був прийнятий на роботу в дати: 1997-12-07, 1995-11-29, 1994-11-11, 1995-04-27, 1997-01-03.
7. Визначити кількість працівників жінок, які були прийняті на роботу у 2000 році.
8. Визначити кількість працівників, які були прийняті на роботу 1989-09-12 та чиє ім'я не містить символ 'а'.
9. Визначити максимальну зарплату працівників, які були прийняті на роботу у 2000 році.
10. Визначити середню зарплату всіх працівників.

Критерії оцінювання практичної роботи 2

За всі правильно виконані завдання можна отримати 100 балів (за кожне завдання по 10 балів).

Перелік рекомендованих джерел

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. SQL Підручник <https://w3schoolsua.github.io/sql/index.html#gsc.tab=0>.
4. 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.

5. Дяченко О.Ф. Математичні основи баз даних : навч. посіб. Маріуполь : МДУ, – Вінниця : Твори, 2020, 136 с.
6. Мулеса О.Ю. Основи мови запитів SQL. Ужгород, 2015, 48 с.

Практична робота № 3. Застосування віконних функцій

Мета: закріпити теоретичні знання і розвинути практичні навички використання віконних функцій.

Теоретичні відомості

Віконні функції – це операції SQL, які виконують обчислення для набору рядків, що пов’язані із поточним рядком. На відміну від агрегаційних функцій, вони не призводять до того, що рядки об’єднуються в один вихідний ряд - рядки зберігають свої окремі ідентичності (рис. 3.1). Їх називають віконними функціями, оскільки вони виконують обчислення для “вікна” рядків. Наприклад, можна розрахувати накопичену суму продажів або визначити найвищий бал в кожній групі [10, 15, 17].

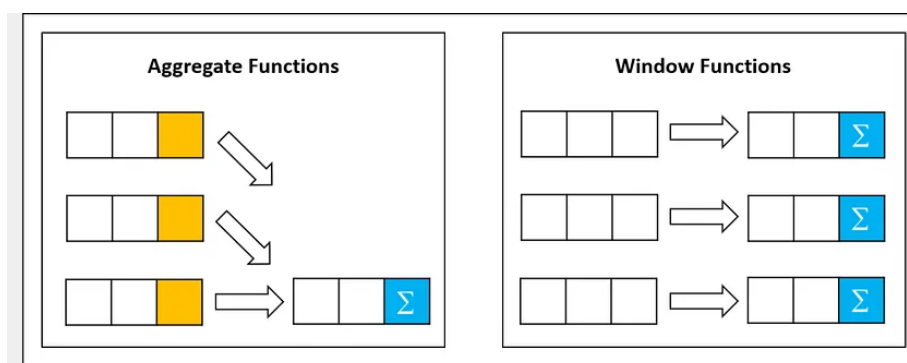


Рис. 3.1. Різниця між агрегаційними функціями та віконними функціями

Синтаксис використання віконних функцій наступний:

```
SELECT column_name,
       WINDOW_FUNCTION(column_name) OVER (
         PARTITION BY column_name
         ORDER BY column_name
         RANGE/ROWS BETWEEN ... AND ...
       )
FROM table_name;
```

Виділяють такі основні типи віконних функцій:

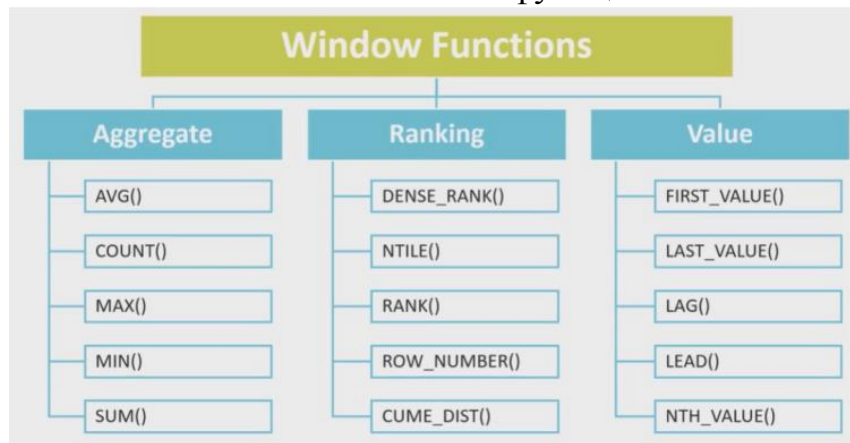


Рис. 3.2. Типи віконних функцій

Приклад 3.1. Сформуванати стовпець з максимальним значенням зарплати для кожного співробітника.

MySQL Workbench interface showing a query and its result grid. The query is:

```

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;
  
```

The result grid shows the following data:

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
10003	43636	1998-12-02	1999-12-02	43699
10003	43478	1999-12-02	2000-12-01	43699
10003	43699	2000-12-01	2001-12-01	43699

В функціях **ранжування** після ключового слова **OVER** обов'язковим має бути задання умови **ORDER BY**, за яким має буде 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 3.2. Проранжувати оцінки за кожним учнем.

```
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;
```

name	subject	grade	row_number	rank	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»

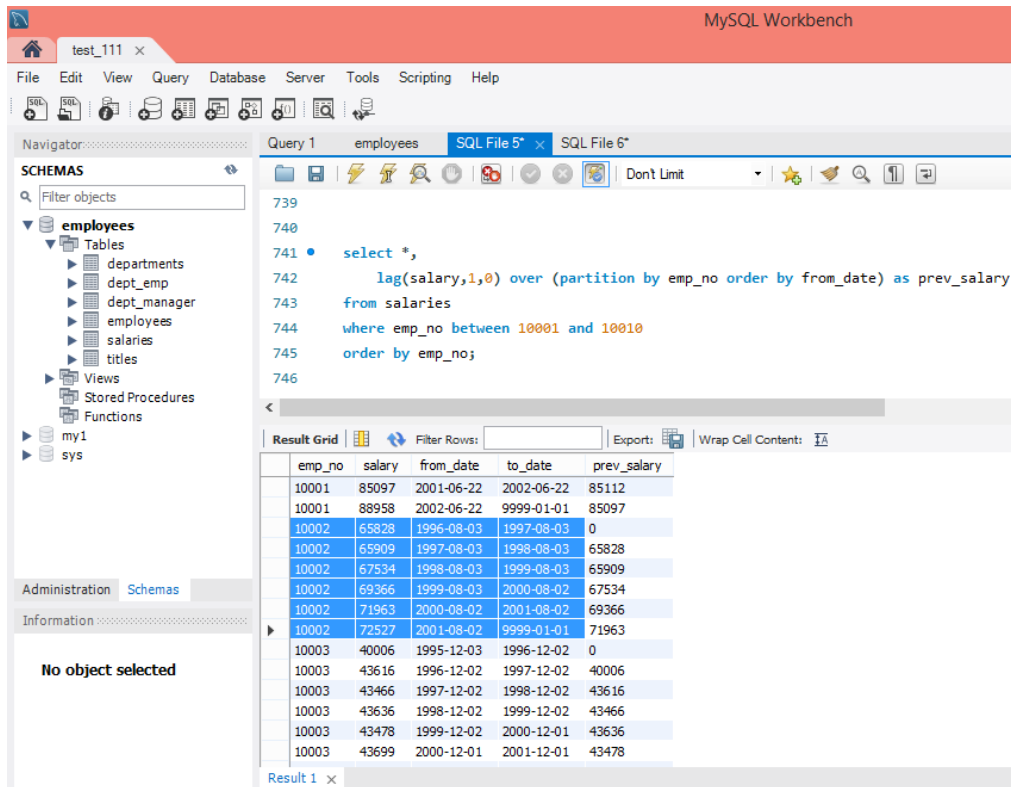
Функції зміщення - це функції, які дозволяють переміщуючись по виділеній партиції таблиці, звертатися до попереднього значення рядка або крайніх значень рядків у партиції.

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.

Приклад 3.3. Сформуванати стовпець із значенням зарплати з сувом на один період для кожного співробітника.



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

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

```

The result grid displays the following data:

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	43636	1998-12-02	1999-12-02	43466
10003	43478	1999-12-02	2000-12-01	43636
10003	43699	2000-12-01	2001-12-01	43478

Завдання до практичної роботи 3

1. Створити стовпець для розрахунку накопичувальної суми заробітної плати за кожним співробітником.
2. Розрахувати, як відрізняється поточна отримана зарплата від максимальної за кожним співробітником.
3. Розрахувати ранг за отриманою зарплатою для кожного співробітника.
4. Знайти різницю в отриманій зарплаті співробітником в порівнянні за попередній період.

Критерії оцінювання практичної роботи 3

За всі правильно виконані завдання можна отримати 100 балів (за кожне завдання по 25 балів).

Перелік рекомендованих джерел

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.

Практична робота № 4. Створення візуалізації в Power BI

Мета: закріпити теоретичні знання і розвинути практичні навички із створення візуальних елементів в Power BI Desktop.

Теоретичні відомості

Вирішення прикладних задач математики [16] та професійних задач аналітики, зокрема аналіз та оптимізація бізнес-процесів, моделювання процесів, прогноз показників бізнес-процесів пов'язані з візуалізацією [4, 7, 9-13]. При проведенні візуалізації розрахунків, отриманих в результаті запиту, можна використати Power BI Desktop (хмарну версію – рис. 4.1 або інсталювану через портал) [18].

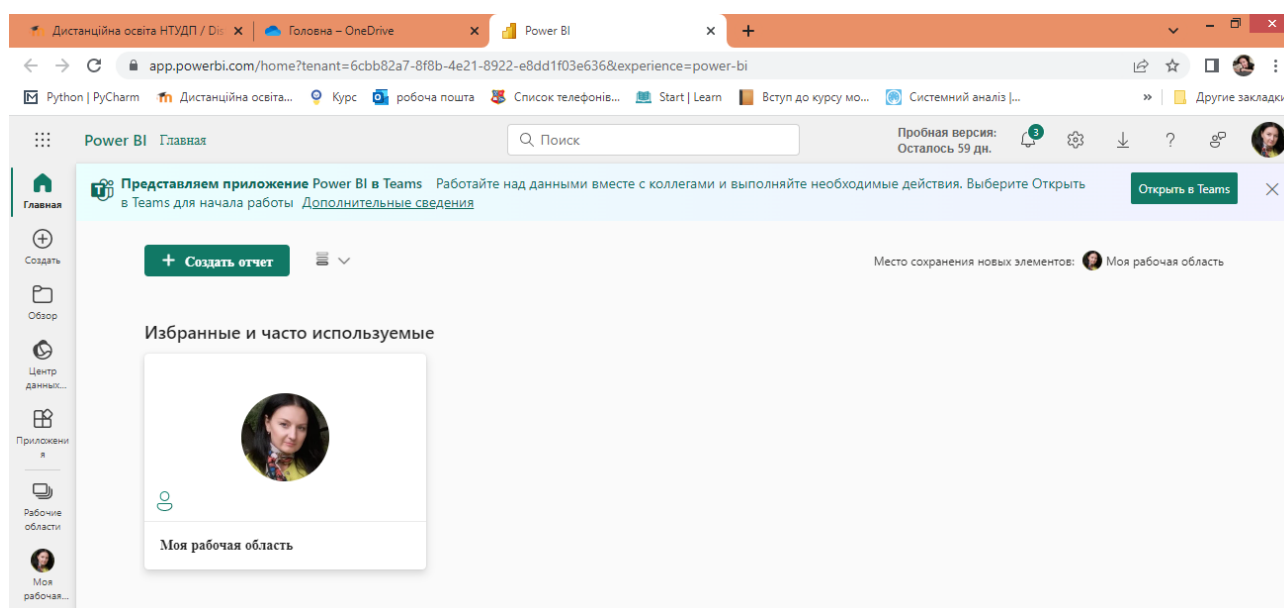


Рис. 4.1. Хмарна версія Power BI Desktop.

Для цього спочатку результат розрахунку або запити із БД можна експортувати в csv файл (рис. 4.2) [10].

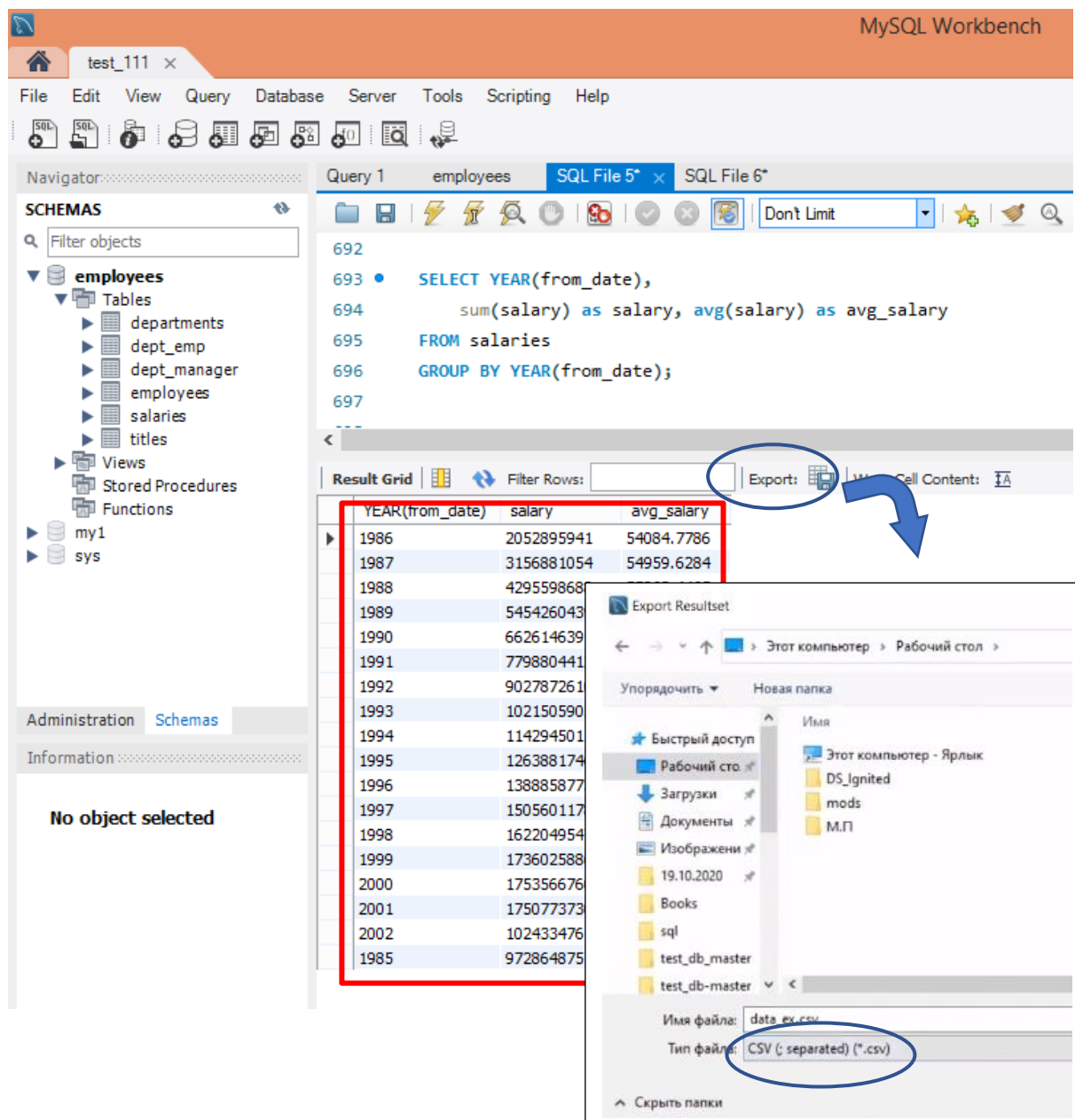


Рис. 4.2. Експорт результату запиту в csv файл

Наступним кроком можна відкрити середовище Power BI через One Drive використовуючи акаунт студента (співробітника) університету НТУ «Дніпровська політехніка», як показано на рис. 4.1. Або інша можливість – з найбільшим доступним функціоналом, зареєструватись на порталі Power BI, а потім інстальювати версію Power BI Desktop [2, 18].

Наступним кроком є обрання робочої області та створення набору даних (можна обрати файл csv), на основі якого буде побудовано візуальні елементи, на рис. 4.3 продемонстровано цей крок.

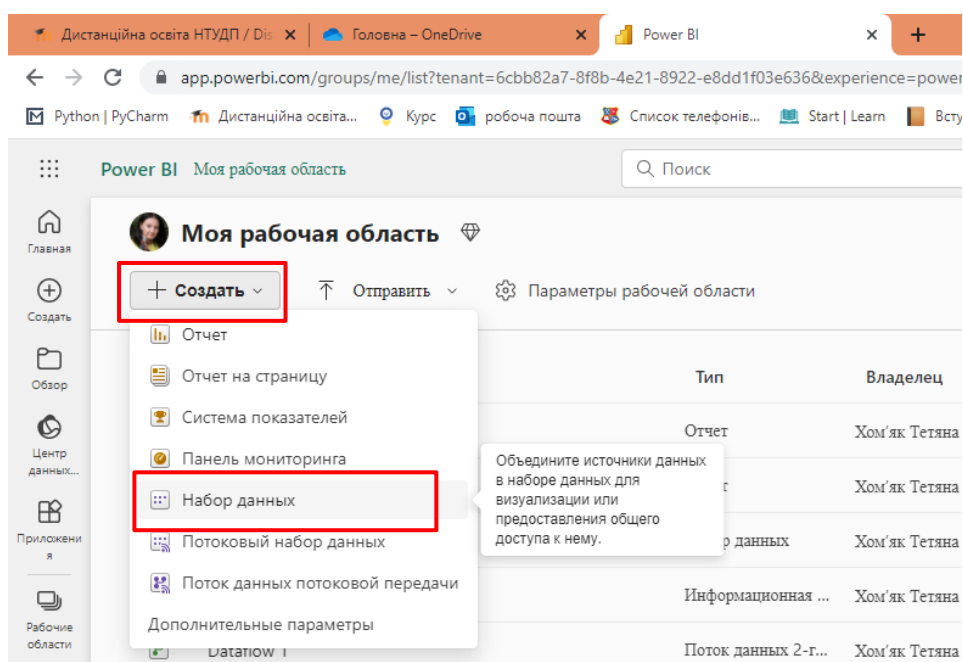


Рис. 4.3. Створення набору даних в робочій області Power BI

Для побудови візуальних елементів (стовпчатих діаграм, графіків, кругових діаграм, тощо) необхідно скористатись панелью *Візуалізація* та обрати елемент, який потрібно пов'язати з відповідними даними панелі *Дані* (рис.4.4).

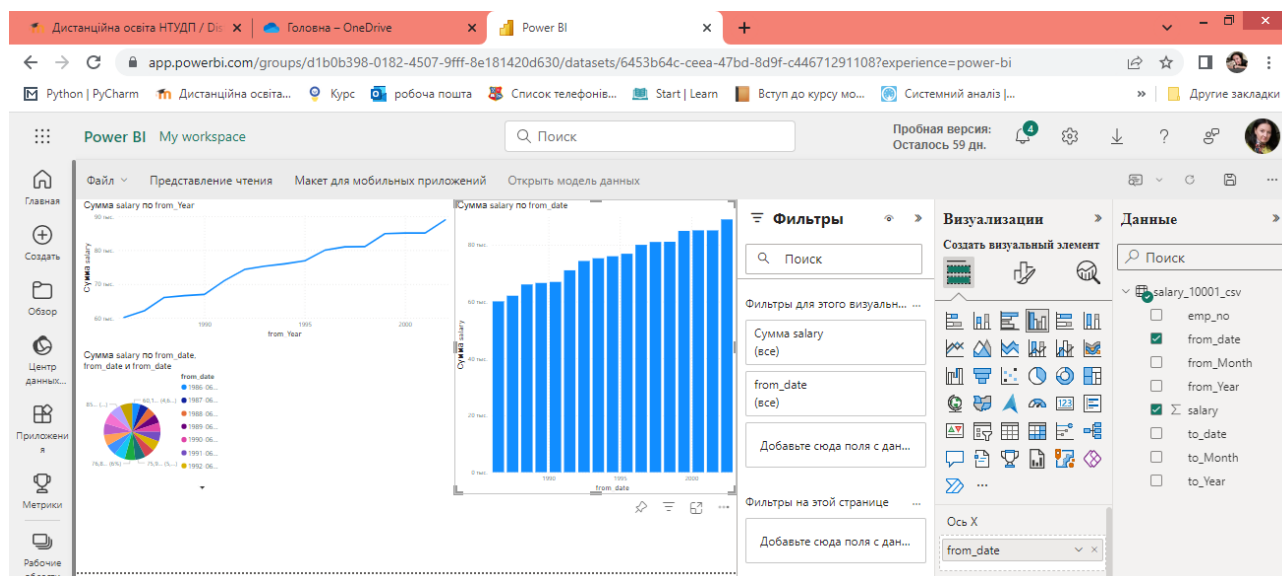


Рис. 4.4. Приклади використання різних візуальних елементів в Power BI для відображення зарплати співробітника за кожен рік

Завдання до практичної роботи 4

1. Знайти з БД Employees всіх працівників, зарплата яких перевищує 150 000 доларів США на рік.
2. За знайденими співробітниками зробити візуалізацію в Power BI залежності зарплати працівників.

Критерії оцінювання практичної роботи 4

За всі правильно виконані завдання можна отримати 100 балів (за кожне завдання по 50 балів).

Перелік рекомендованих джерел

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.

Практична робота № 5. Створення OLAP-кубу

Мета: закріпити теоретичні знання і розвинути практичні навички із алгоритму побудови OLAP-кубу та створенню аналітичних зрізів куба.

Теоретичні відомості

Для створення OLAP-кубу буде використано БД AWTO [6], в якій створено запит з обчислення вартості продажів автомобілів та його результати записано в таблицю SHOW, яку додано в діаграму та об'єднано з іншими таблицями БД. Таблиця SHOW виступає таблицею фактів в сховищі даних. Структура таблиці SHOW наступна (рис. 5.1).

Имя столбца	Тип данных	Разрешит...
Код_региона	numeric(2, 0)	<input type="checkbox"/>
Код_філії	numeric(3, 0)	<input type="checkbox"/>
Код_менеджера	numeric(3, 0)	<input type="checkbox"/>
Код_авто	numeric(3, 0)	<input type="checkbox"/>
Код_моделі	numeric(2, 0)	<input type="checkbox"/>
Модель	char(10)	<input checked="" type="checkbox"/>
Дата_продажі	date	<input checked="" type="checkbox"/>
Кількість_продано	numeric(2, 0)	<input checked="" type="checkbox"/>
[Вартість продаж]	numeric(9, 0)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Рис. 5.1. Структура таблиці SHOW

Діаграма БД після додавання таблиці SHOW представлена на рис. 5.2.

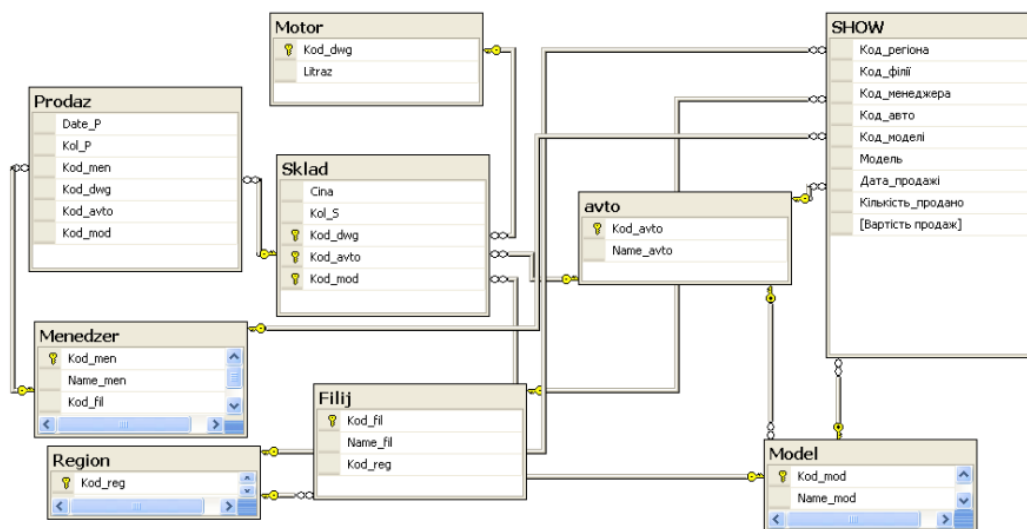


Рис. 5.2. Діаграма бази даних

Алгоритм створення OLAP-кубу полягає у виконанні наступних кроків:

1. Завантажити SQL Server Business Intelligence Development Studio.
2. Створити OLAP-проект, вказуючи його ім'я OLAP_2025.
3. Створити джерела даних на основі БД АВТО.
4. Створити подання джерел даних, вибираючи з БД таблицю фактів та таблиці вимірів.
5. Створити куб, вибираючи спочатку таблицю фактів, а потім таблицю вимірювань.
6. Робота з кубом даних та створення аналітичних зрізів куба.

1. Завантаження Business Intelligence Development Studio

Microsoft SQL Server включає в себе відокремлений додаток, що має назву Microsoft SQL Server Analysis Services для виконання OLAP аналізу. Для роботи з цим додатком використовують Business Intelligence Development Studio (BIDS), що також включається в Microsoft SQL Server.

Робота з OLAP аналізу в Microsoft SQL Server починається із завантаження Business Intelligence Development Studio програми [14]:

Пуск — Всі програми — Microsoft SQL Server — SQL Server Business Intelligence Development Studio (рис. 5.3)

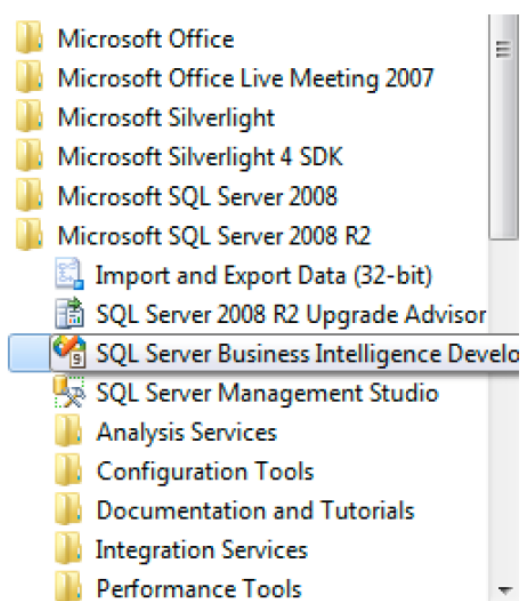


Рис. 5.3. Вибір пункту меню для завантаження BIDS

2. Створення Analysis Services проекту

Після відкриття BIDS необхідно створити новий Analysis Services проект
Файл — Создать Проект (File — New — Project).

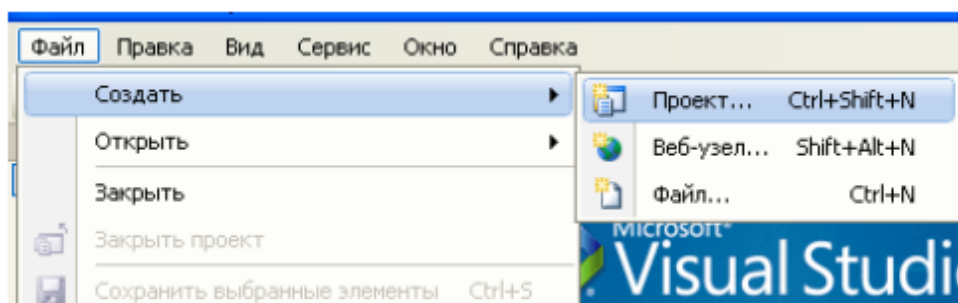


Рис. 5.4. Вікно меню для вибору створення Analysis Services-проекту

У вікні, що відкриється (рис. 5.4), необхідно обрати *Создать проект служб Analysis Services*, як тип проекту, що буде використовуватися, а

також вказати ідентифікуючі параметри проекту, а саме: назву проекту, місце фізичного розташування файлів проекту та назву рішення проекту (рис. 5.5).

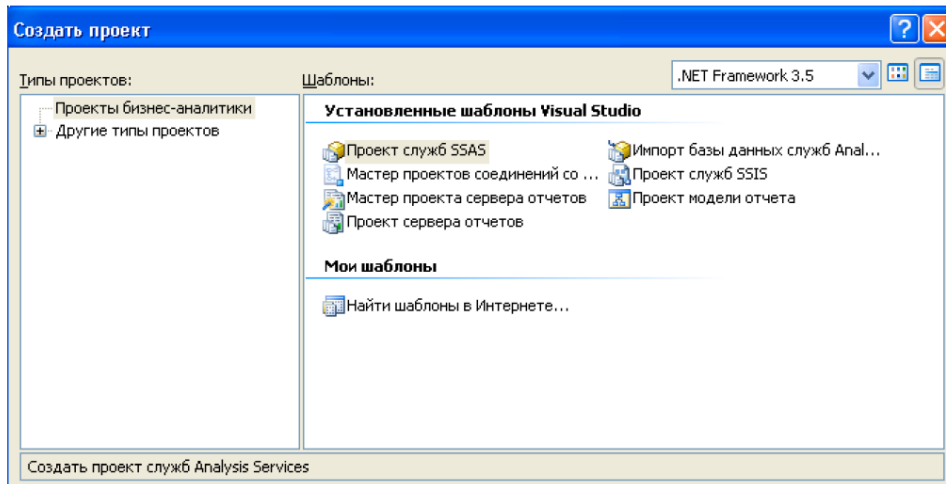


Рис. 5.5. Вікно меню для визначення ідентифікаційних параметрів Analysis Services-проекту

Після введення даних та натискання кнопки ОК система створює проект в середовищі Business Intelligence Development Studio (BIDS), назва якого з'явиться у вікні Оглядача рішень (Solution Explorer). В нашому випадку це проект OLAP_2025.

Оглядач рішень можна активізувати натиснувши кнопку головного меню.

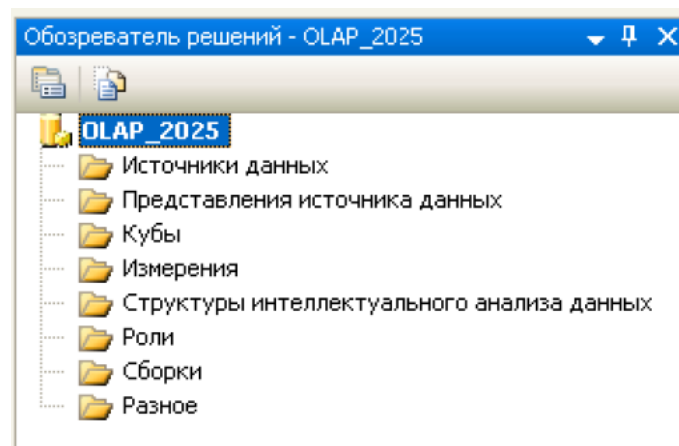


Рис. 5.6. Вікно оглядача рішень з назвою створеного проекту

3. Створення джерел даних (Data Source)

Після створення проекту, необхідно створити джерело даних (Data Source) до проекту. Для цього у вікні Оглядача рішень (Solution Explorer) необхідно знайти папку Джерела даних (Data Sources) та через контекстне меню потрібно обрати Створити джерело даних (New Data Source...) (рис. 5.7).

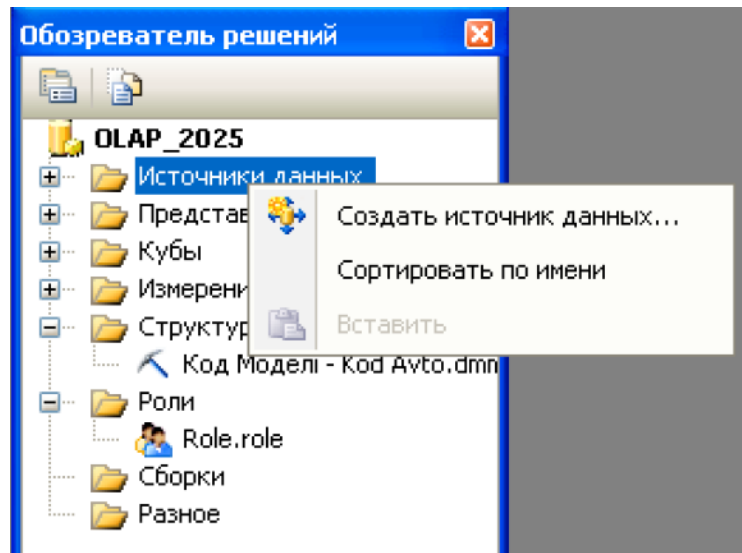


Рис. 5.7. Вікно для активізації створення джерела даних

Після чого відкриється Майстер Джерел Даних (Data Source Wizard), в якому треба натиснути кнопку Далі (Next).

На наступному кроці необхідно вказати з'єднання з даними (Data Connection), що і визначить джерело даних (Data Source). Для цього треба обрати кнопку Створити (New) (рис. 5.8) для відкриття Менеджера (диспетчера) з'єднання (Connection Manager).

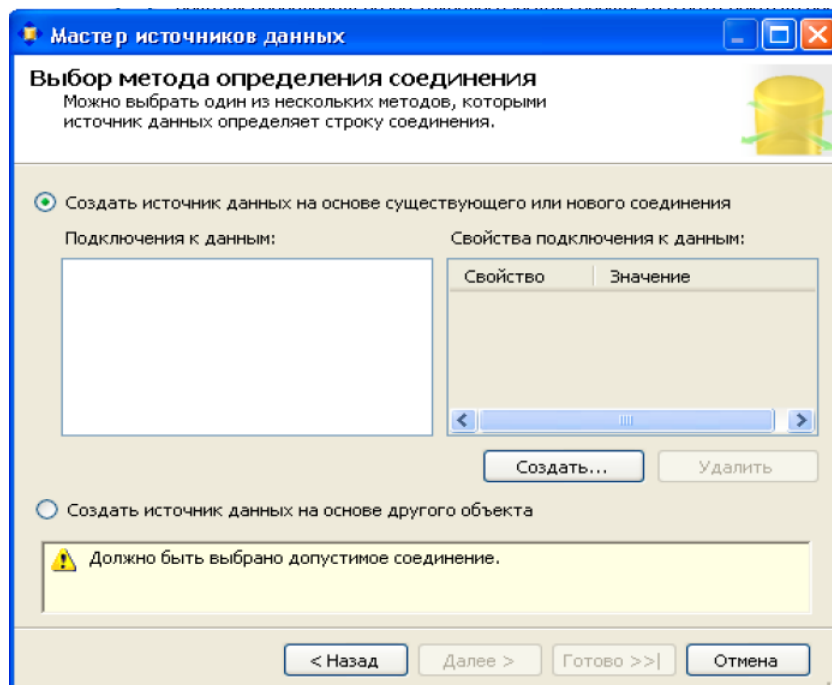


Рис. 5.8. Вікно для відкриття Менеджера (диспетчера) з'єднання (Connection Manager)

У вікні, що відкриється (рис. 5.9) в полі Ім'я сервера (Server Name) вказуємо адресу (ім'я) сервера, де знаходиться SQL Server. Так як в нашому

випадку SQL Server знаходиться на поточному комп'ютері, то в полі Server Name вказуємо localhost (рис. 5.10).

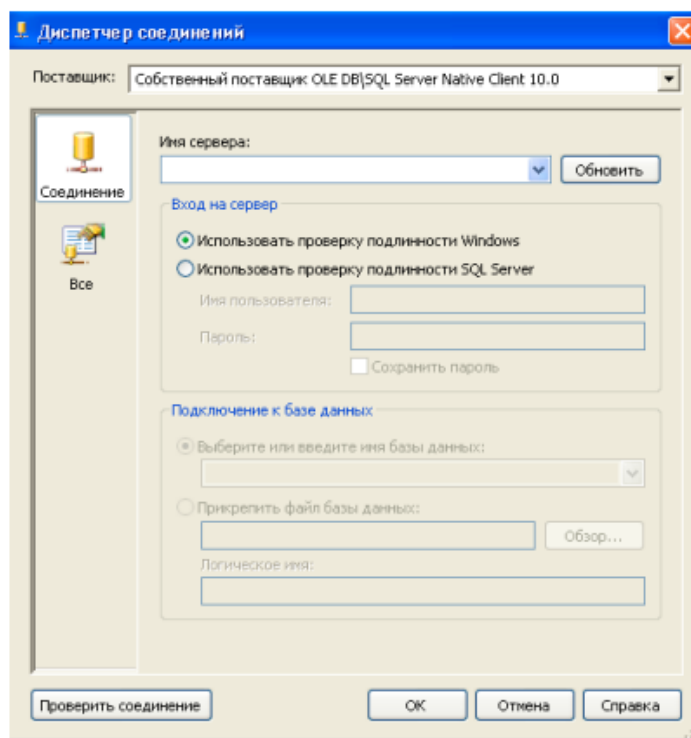


Рис. 5.9. Вікно диспетчера з'єднання

В полі Вхід на сервер (Log on to the server) необхідно вказати тип авторизації на SQL Server. Параметр Use Windows Authentication означає, що система буде намагатися з'єднатися із SQL сервером використовуючи логін та пароль поточного користувача Windows. Параметр Use SQL Server Authentication означає, що система буде використовувати логін та пароль, що задаються та зберігаються виключно на SQL сервері.

Для виконання практичної роботи можна обирати Use Windows Authentication. В групі Підключення до бази даних (Connect to database) необхідно обрати базу даних, яка і буде джерело даних. Заповнюючи параметр *Виберіть чи введіть ім'я бази даних* (Select or enter a database name) користувач має можливість обрати базу даних, що зберігається на сервері, вказаному в полі Server Name.

В полі Прикріпити файл до бази даних (Attach a database file) користувач має можливість обрати базу даних у вигляді файлу бази даних (розширення .mdf) — на випадок, якщо база даних від'єднана від SQL сервера.

Обираємо параметр *Виберіть чи введіть ім'я бази даних* (Select or enter a database name) та заповнюємо його значенням ім'ям бази даних AWTO. Заповнене вікно диспетчера з'єднань наведено на рис. 5.10.

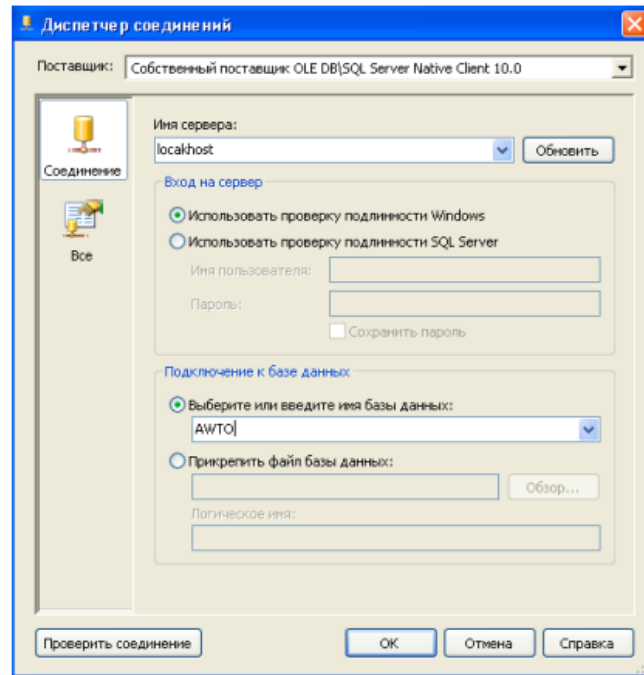


Рис. 5.10. Вікно диспетчера з'єднання після заповнення відповідними даними.

Для перевірки правильності з'єднання із джерелом даних - натискаємо *Перевірити з'єднання* (Test Connection). У випадку коректності з'єднання система відобразить відповідне повідомлення. Якщо ж система не змогла встановити з'єднання із базою даних, то відповідне повідомлення буде показано користувачу, або у вигляді неіснуючого імені SQL серверу, або задання імені користувача, яке не існує на SQL сервері.

Після вдалої перевірки з'єднання із базою даних у вікні *Менеджера З'єднання* натискаємо ОК, після чого система автоматично заповнить необхідні поля у вікні Майстра Джерел Даних (Data Source Wizard) (рис. 5.11).

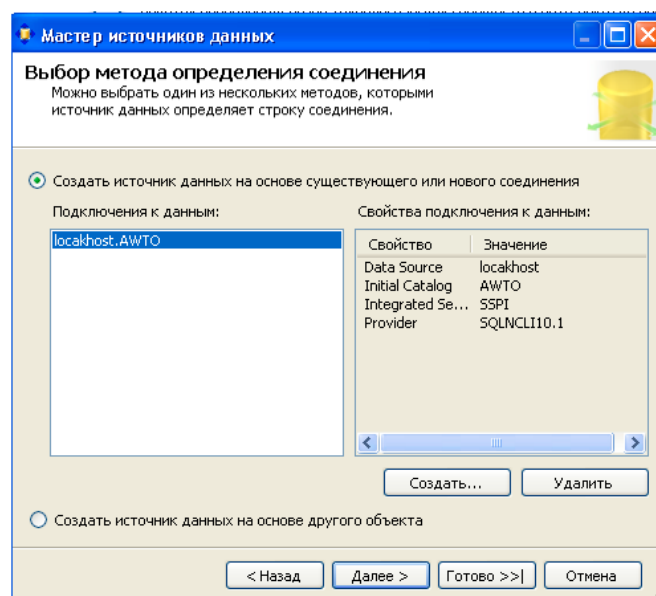


Рис. 5.11. Вікно Майстра джерел даних

Натискаємо Далі (Next), з'явиться вікно *Олицетворения*, в якому вибираємо параметр *Успадкування* (Inherit) і знову натискаємо Далі (Next).

В наступному вікні необхідно вказати користувача, який SQL Server Analysis Services буде використовуватися для з'єднання з джерелом даних. Після чого з'явиться наступне вікно (рис. 5.12), яке завершує роботу майстра джерела даних, в якому буде вказано ім'я бази даних AWTO - джерела даних для сховища даних.

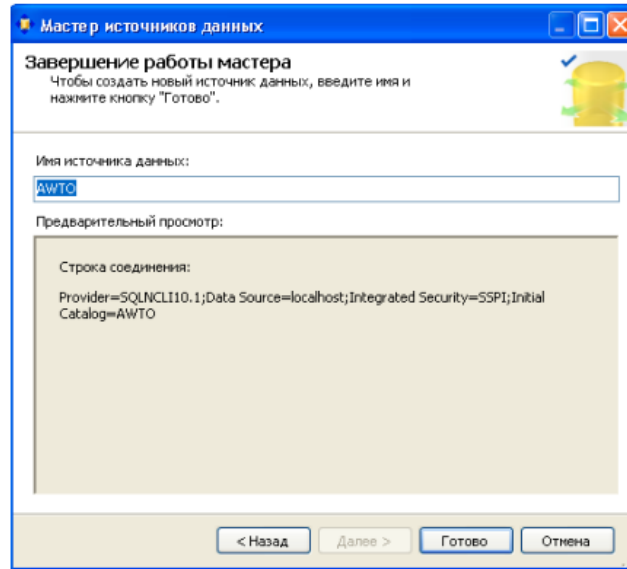


Рис. 5.12. Вікно завершення роботи майстра для вибору джерела даних

У вікні (рис. 5.12) останнього кроку необхідно обрати *Готово* (Finish) і система створить джерело даних, яке з'явиться у вікні Оглядач рішень (Solution Explorer) (рис. 5.13).

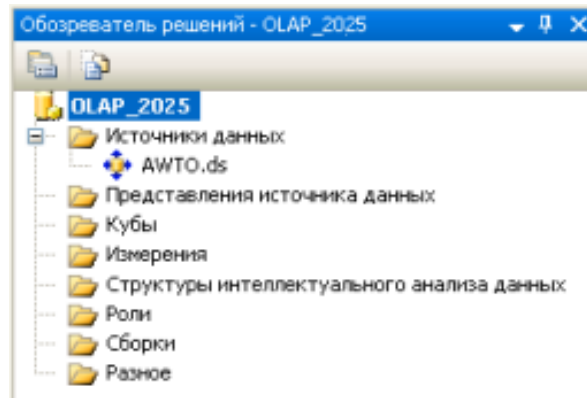


Рис. 5.13. Вікно оглядача рішень з вибраним джерелом даних

4. Створення подання джерел даних (Data Source View)

Після створення джерела даних необхідно створити подання джерел даних (Data Source Views). Для цього у вікні *Оглядач рішень* (Solution Explorer) необхідно знайти папку *Представлення джерела даних* (Data Source Views) та з контекстного меню потрібно обрати *Створити представлення джерела даних* (New Data Source View...) (рис. 5.14).

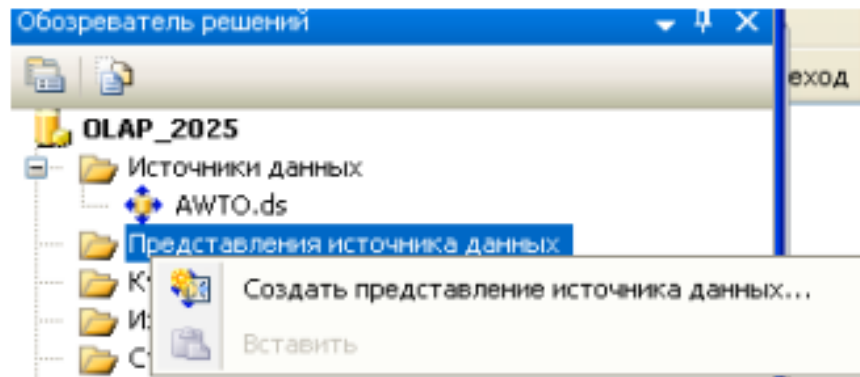


Рис. 5.14. Вікно для створення представлення джерела даних

Далі відкриється Майстер Подання Джерел Даних (Data Source View Wizard), після чого треба обрати джерело реляційних даних. (рис. 5.15).

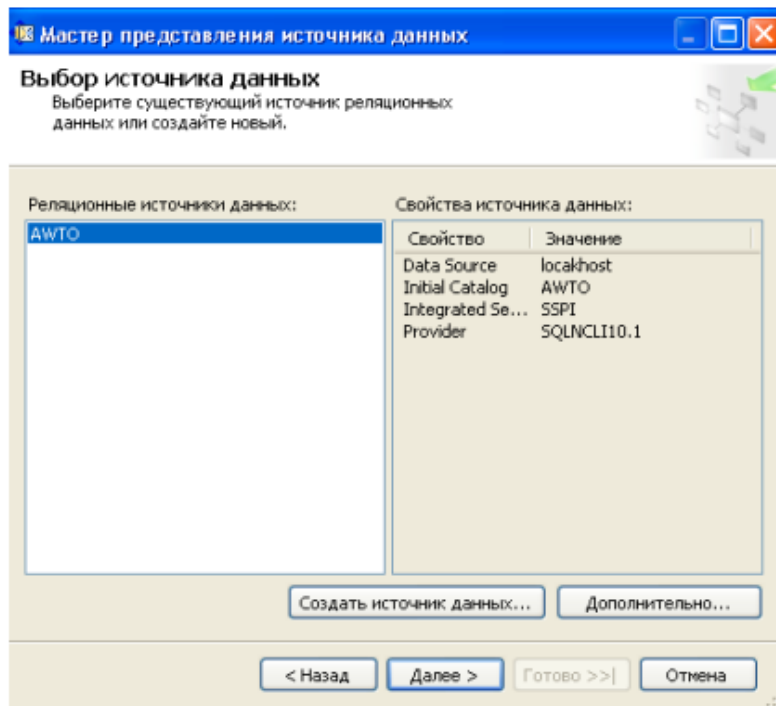


Рис. 5.15. Вікно для вибору джерела даних

На цьому кроці необхідно вказати джерело даних. Обираємо раніше створене джерело даних - базу даних AWTO та після з вікна вибору таблиць (tables) та представлень (view) необхідно обрати таблиці із обраного джерела даних, що будуть приймати участь у створенні куба (рис. 5.16).

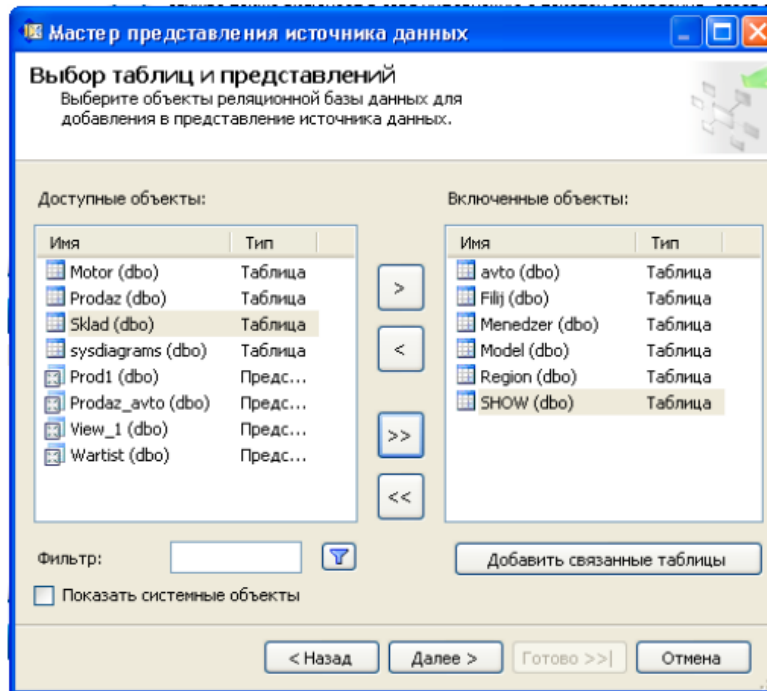


Рис. 5.16. Вікно для вибору таблиц і представлень

Система створить джерело даних, яке з'явиться у вікні *Оглядача рішень* (Solution Explorer), та відповідну діаграму сховища даних (рис. 5.17).

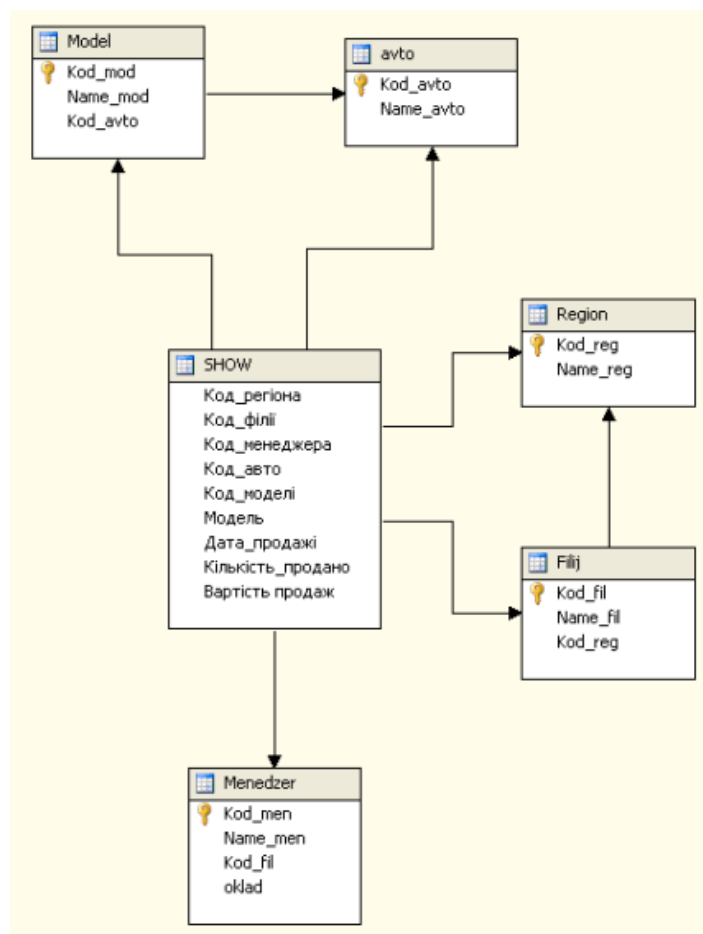


Рис. 5.17. Діаграма сховища даних

5. Створення кубу даних (Cube)

Після створення подання джерела даних необхідно створити куб даних Куб (Cubes). Для цього у вікні *Оглядач рішень* (Solution Explorer) в папці *Куб* (Cubes) із контекстного меню потрібно обрати *Створити куб* (New Cubes) (рис. 5.18).

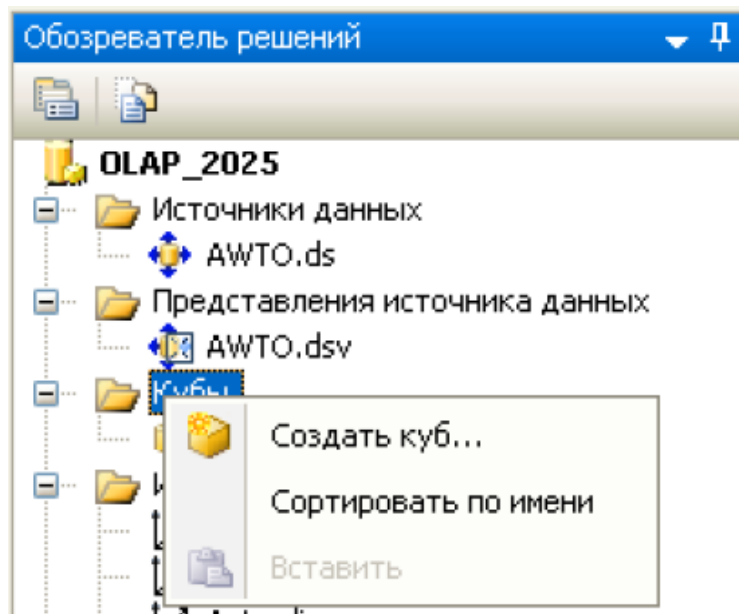


Рис. 5.18. Вікно для створення кубу даних.

Відкриється Майстер Кубів (Data Source View Wizard), в якому обираємо опцію *Використовувати існуючі таблиці* (Use existing tables), тобто параметр для створення кубу на основі вже існуючих таблиць із створеного подання джерела даних. У вікні (рис. 5.19) *Вибір таблиць груп мір* (Select Measure Group Tables) треба обрати таблицю фактів SHOW, як таку, що містить *Показники* (measures) для кубу даних (рис. 5.19).

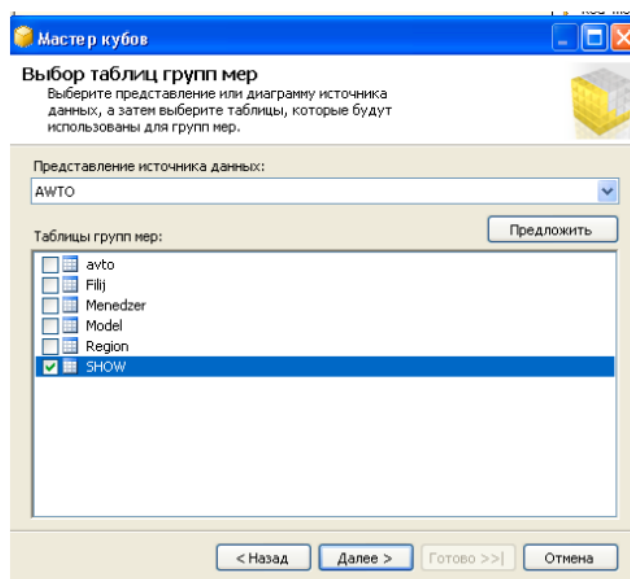


Рис. 5.19. Вікно вибору таблиці фактів для куба

В наступному вікні (рис. 5.20) безпосередньо обираємо *показники* (measures) із таблиці обраної на попередньому кроці: Кількість продано та Вартість Продаж.

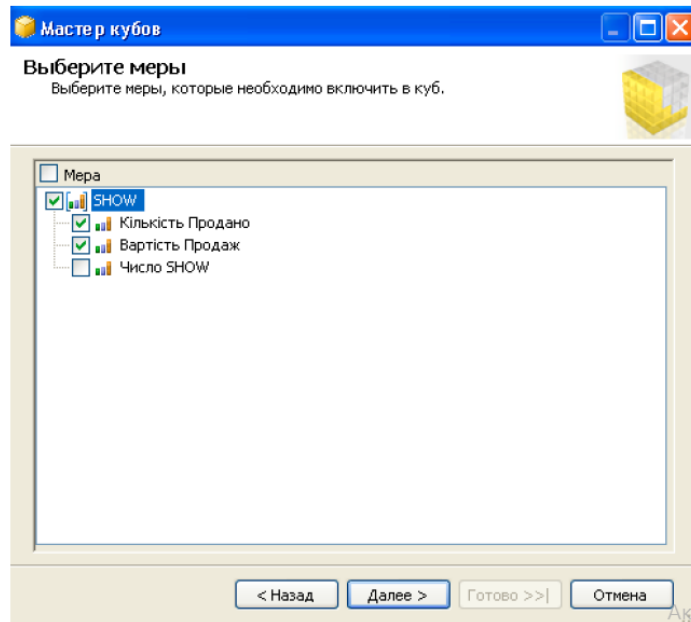


Рис. 5.20. Вікно вибору показників (змінних) куба

Після обрання показників (measures), що будуть аналізуватись, система пропонує обрати виміри (ознаки), за якими буде виконуватись аналіз кубу даних (рис. 5.21, 5.22).

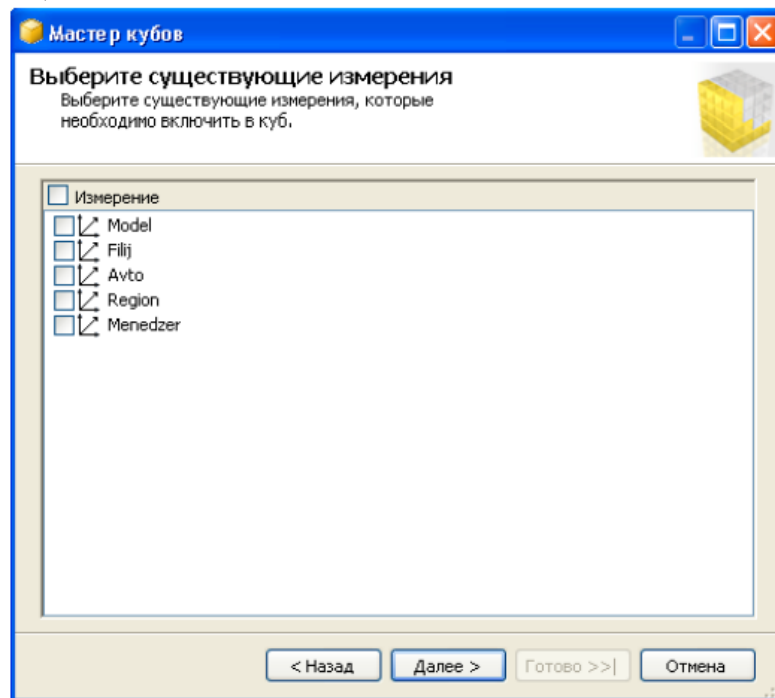


Рис. 5.21. Вікно вибору вимірів куба

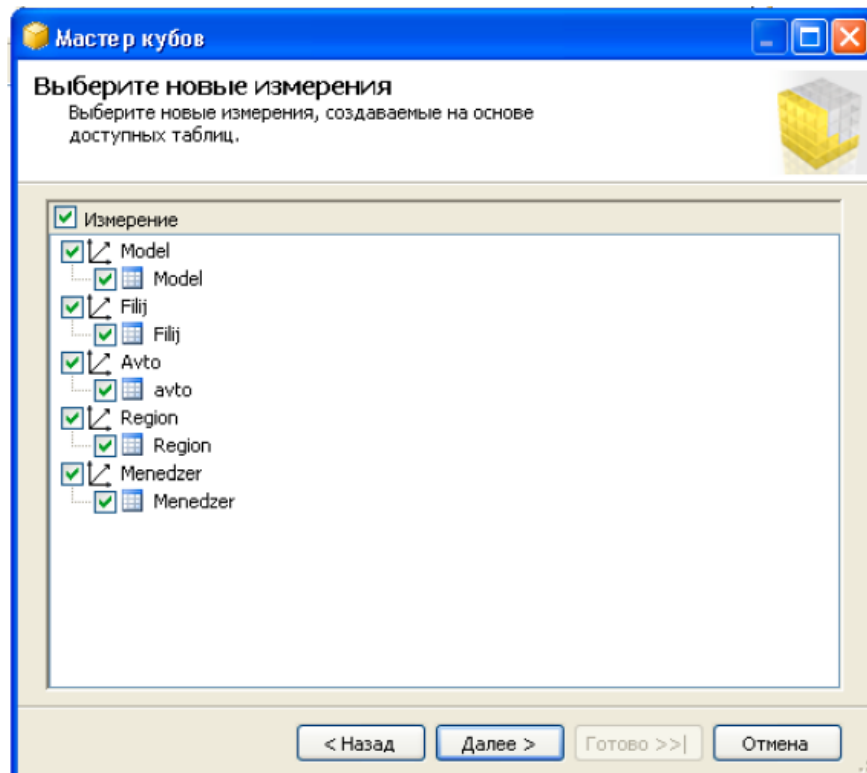


Рис. 5.22. Вікно вибору вимірів куба

На останньому кроці треба задати ім'я кубу даних (рис. 5.23).

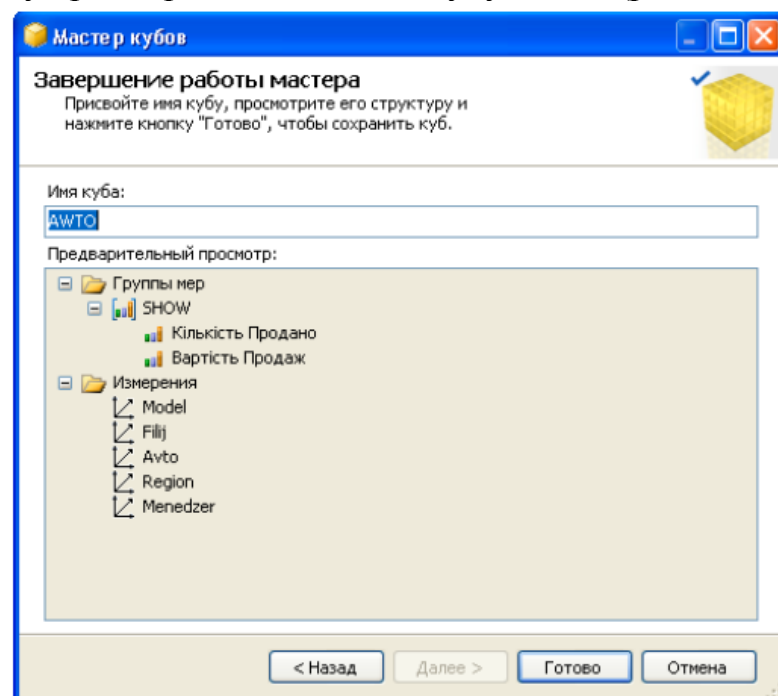


Рис.5.23. Вікно для задання ім'я куба

Система створює куб даних, який з'явиться у вікні *Оглядач рішень* (Solution Explorer) (рис. 5.24).

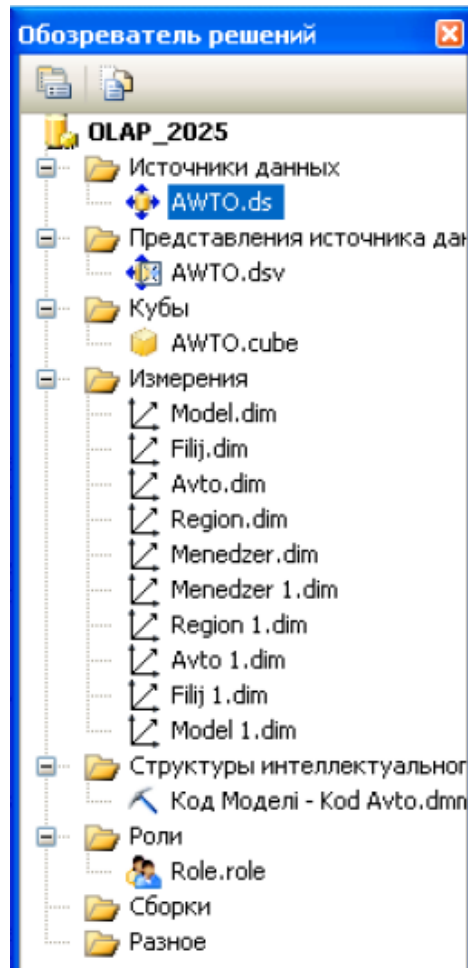


Рис. 5.24. Вікно зі створеним кубом AWTO.

5. Работа з кубом даних

Перед безпосередньою роботою з кубом необхідно перенести його структуру та дані до SQL Server Analysis Services. Для цього необхідно натиснути правою кнопкою на назві проекту та спочатку обрати Побудова (Build), а потім Розгортання (Deploy) (рис. 5.25).

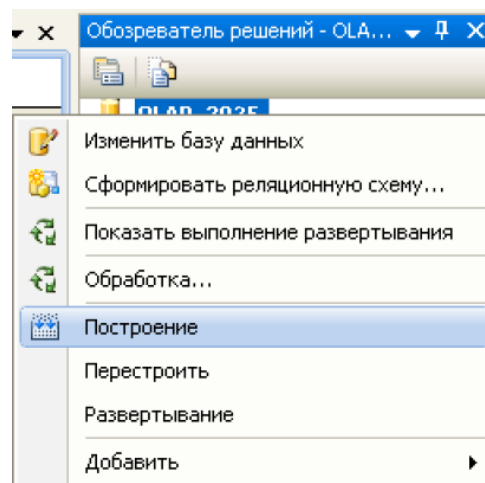


Рис. 5.25. Вікно побудови та розгортання куба

Про виконання оброблення БД та розгортання куба буде повідомлено у вікні *Виконання розгортання куба* (рис. 5.26). Якщо під час цієї процедури виникнуть помилки, то вони будуть вказані також в цьому вікні.

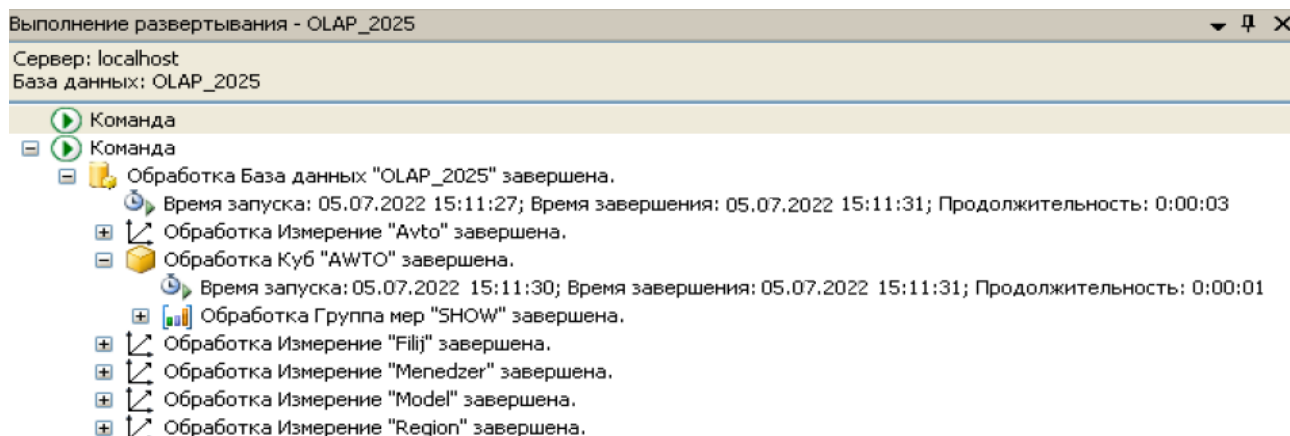
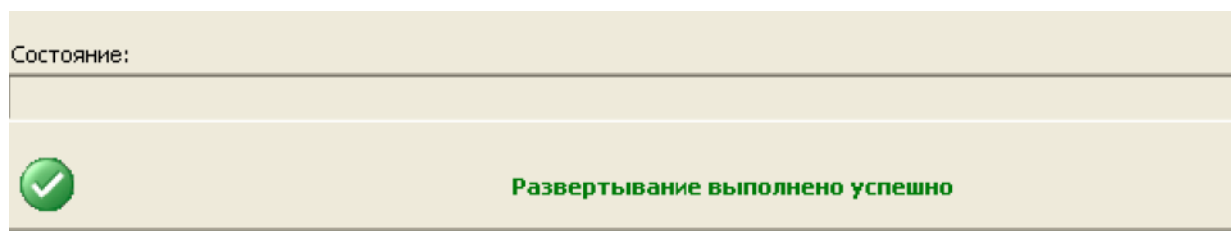


Рис. 5.26. Вікно діагностики про стан розгортання куба

У випадку вдалої компіляції проекту, вдалого з'єднання з базою даних та коректного відтворення структури в SQL Server Analysis Services система видасть відповідне повідомлення у вкладці Стан (Status):



Для безпосереднього перегляду створеного кубу, необхідно подвійним натисканням лівої кнопки мишки по назві проекту відкрити вікно схеми кубу та натиснути на кнопку Browser на верхній панелі меню. Відкриється наступне вікно (рис. 5.27), в якому можна вибирати можливі зрізи куба

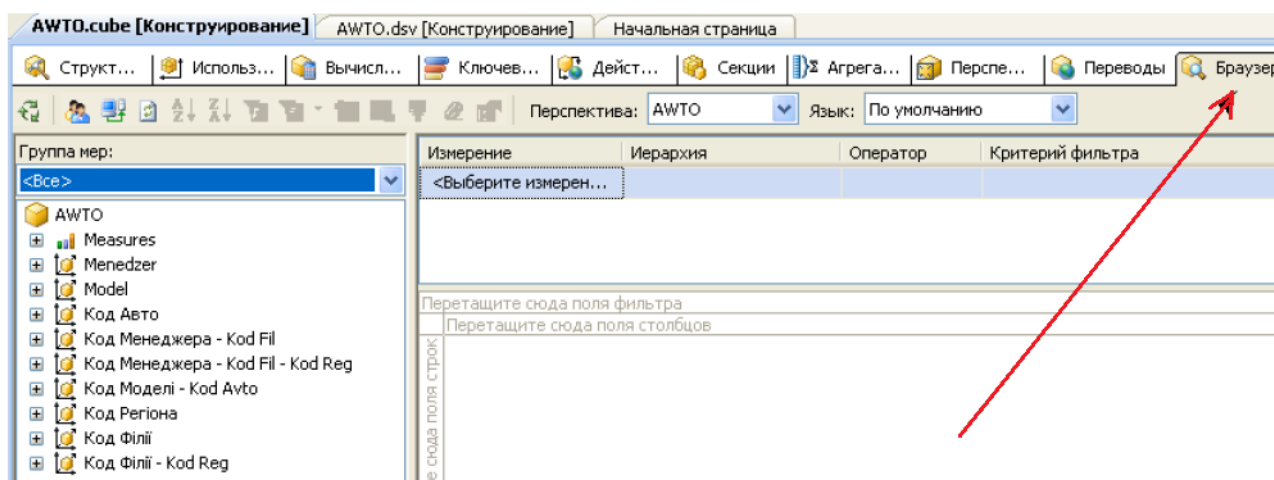


Рис. 5.27. Вікно для вибору варіантів зрізу куба

В цьому вікні можна відображати різні зрізи даних куба, тобто виконується безпосередня робота з кубом даних. Наприклад, можна задати зріз куба, який показуватиме кількість і вартість продаж автомобілів у розрізі менеджерів та філій (рис. 5.28). Для цього необхідно перетягнути відповідні поля з лівої частини вікна у відповідні колонки та рядки таблиці праворуч так, щоб у рядках знаходилися такі виміри (ознаки за якими виконується аналіз): Kod Filii, а в стовпчиках, як вимір, використовуємо: Kod Avto, на перетині відповідного рядка і стовпчика буде відображатися поле — Кількість Продано, Вартість Продаж. Комбінацію вимірів (ознак) можна змінювати відповідно до потреб бізнес-аналізу.

The screenshot shows a software interface for data cube analysis. On the left is a tree view of dimensions: 'AWTO' (Measures: SHOW, Вартість Продаж, Кількість Продано), 'Menedzer', 'Model', 'Kod Avto', 'Kod Менеджера - Kod Fil', 'Kod Менеджера - Kod Fil - Kod R', 'Kod Моделі - Kod Avto', 'Kod Регіона', 'Kod Філії', and 'Kod Філії - Kod Reg'. The main area displays a pivot table with 'Kod Avto' as the column dimension and 'Kod Fil' as the row dimension. The table shows sales volume and value for two different car models across two branches.

		Kod Avto		Общие итоги	
		1	7		
Kod Fil		Кількість Продано	Вартість Продаж	Кількість Продано	Вартість Продаж
1		5	500000	2	1000000
2		1	100000	1	500000
	Общие итоги	6	600000	3	1500000

Рис. 5.28. Приклад зрізу куба

Завдання для практичної роботи 5

1. В базі даних створіть запит, який би створював таблицю фактів, що буде виступати джерелом показників (змінних) для бізнес-аналізу. Цю таблицю долучіть до діаграми (схеми) даних, встановивши відповідні зв'язки. Створена таблиця буде основним джерелом даних сховища даних.

2. Визначити ознаки (виміри), згідно яких буде виконуватися аналіз показників (змінних) та перевірити їх наявність в базі даних.

3. Реалізувати OLAP-куб в середовищі Microsoft SQL Server Analysis Services.

4. Опрацювати різні варіанти створення зрізів кубу у відповідності до потреб бізнес-аналізу.

Критерії оцінювання практичної роботи 5

За всі правильно виконані завдання можна отримати 100 балів (за завдання 1 та 2 – по 20 балів, за 3 та 4 завдання - по 30 балів).

Перелік рекомендованих джерел

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. проєкт магістра. Запоріжжя : Запорізька політехніка, 2023, 87 с.

Список використаних джерел

1. 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.
2. Документація Microsoft Power BI. Режим доступу: <https://learn.microsoft.com/ru-ru/power-bi/fundamentals/power-bi-overview>
3. Дяченко О.Ф. Математичні основи баз даних : навч. посіб. Маріуполь : МДУ, – Вінниця : Твори, 2020, 136 с.
4. Моделювання та реінжиніринг бізнес-процесів: підручн. / С.В. Козир, В.В. Слесарев, С.А. Ус, Т.В. Хом'як; М-во освіти і науки України; Нац. техн. ун-т «Дніпровська політехніка». Дніпро: НТУ «ДП», 2022. 163 с.
5. Мулеса О.Ю. Основи мови запитів SQL. Ужгород, 2015. 48 с.
6. Ситник Н. В. Організація баз та сховищ даних [Електронний ресурс] : практикум. Київ : Київський нац. екон. ун-т, 2017, 148 с.
7. Прус О. В., Хом'як Т. В. Аналіз якості освіти у сільських та міських школах: проблеми та шляхи їх вирішення // Матеріали I (VII) міжнародної науково-практичної конференції здобувачів вищої освіти і молодих учених «Інформаційні технології: теорія і практика» 20 – 22 березня 2024 р, НТУ «ДП», 2024, с. 468-472.
8. Положення про порядок видання в світ інформаційно-методичного забезпечення освітнього процесу в Національному технічному університеті «Дніпровська політехніка» / М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». Дніпро : НТУ «ДП», 2024, 31 с.
9. Хом'як Т.В. Аналіз та моделювання діяльності додаткових курсів на кафедрі: дипл. проєкт магістра. Запоріжжя : Запорізька політехніка, 2023, 87 с.
10. Хом'як Т. В. 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, 192 с.
11. Хом'як Т.В., Коханчик Н.С., Малієнко А.В. Вирішення задачі маршрутизації транспорту на підприємстві // Збірник наукових праць НГУ. 2020. №63, с. 145-155.
12. Хом'як Т.В., Малієнко А.В. Симонець Г.В. Застосування методів згладжування для прогнозування обсягу виробництва // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології Збірник наукових праць. № 1'2019, С 8 -12.
13. Хом'як Т.В., Малієнко А.В., Фішбах К.К. Розробка системи підтримки прийняття рішень для вибору виду діяльності // Збірник наукових праць НГУ. 2019. № 59, с. 132-142 (doi.org/10.33271/crpnmu/59.132).
14. Gerardus B. OLAP cube. Toronto : 5STARCOOKS, 2022, 307 p.
15. Korotkevitch D. SQL Server Advanced Troubleshooting and Performance Tuning. Best Practices and Techniques, USA : O'Reilly Media, 2022, 500 p.

16. Khomyak T. Stabilization of the Unstable Spinning of a Lagrange Top Filled with a Fluid / T. Khomyak // International Applied Mechanics, 2015, V.51, № 6, P. 702-709.

17. 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.

18. Microsoft Power BI Cookbook: Creating Business Intelligence Solutions of Analytical Data Models, Reports, and Dashboards. Birmingham : Packt Publishing Ltd, 2017, 802 p.

19. SQL Підручник <https://w3schoolsua.github.io/sql/index.html#gsc.tab=0>.

Навчальне видання

Коряшкіна Лариса Сергіївна
Хом'як Тетяна Валеріївна
Малієнко Андрій Вікторович
Гаранжа Дмитро Миколайович

БАЗИ ДАНИХ В ПРОФЕСІЙНИХ ЗАДАЧАХ АНАЛІТИКИ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня магістра
освітньо-професійної програми «Системний аналіз»
зі спеціальності 124 Системний аналіз

Видано в авторській редакції.

Електронний ресурс.
Підписано до видання 16.07.2024. Авт. арк. 3,6.

Національний технічний університет «Дніпровська політехніка».
49005, м. Дніпро, просп. Дмитра Яворницького, 19.