МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ

****

ЗВІТ ДО ЛАБОРАТОРНИХ РОБІТ №№ 1-6

з курсу:

**"Бази даних"**

**ВИКОНАВ:**

слухач групи ПЗС-11

Гринчук Т.А.

**ПЕРЕВІРИЛА:**

доц. Павич Н.Я.

Львів – 2014

**Лабораторна робота №1**

**Мета роботи:** Встановити та налаштувати SQL Server 2008, навчитися створювати бази даних.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Microsoft SQL Server — комерційна [система керування базами даних](http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85), що розповсюджується корпорацією[Microsoft](http://uk.wikipedia.org/wiki/Microsoft). Мова, що використовується для запитів — [Transact-SQL](http://uk.wikipedia.org/w/index.php?title=Transact-SQL&action=edit&redlink=1), створена спільно Microsoft та [Sybase](http://uk.wikipedia.org/wiki/Sybase). Transact-SQL є реалізацією стандарту [ANSI](http://uk.wikipedia.org/wiki/ANSI)/[ISO](http://uk.wikipedia.org/wiki/ISO) щодо структурованої мови запитів ([SQL](http://uk.wikipedia.org/wiki/SQL)) із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

Базовий код MS SQL Server (до версії 7.0) ґрунтувався на коді [Sybase SQL Server](http://uk.wikipedia.org/w/index.php?title=Sybase_SQL_Server&action=edit&redlink=1). Це дозволило [Microsoft](http://uk.wikipedia.org/wiki/Microsoft) вийти на ринок баз даних для підприємств, де конкурували [Oracle](http://uk.wikipedia.org/wiki/Oracle_Corporation), [IBM](http://uk.wikipedia.org/wiki/IBM), і, пізніше, сама Sybase. Microsoft, Sybase і [Ashton-Tate](http://uk.wikipedia.org/w/index.php?title=Ashton-Tate&action=edit&redlink=1) спочатку об'єдналися для створення і випуску на ринок першої версії програми, що отримала назву SQL Server 1.0 для [OS/2](http://uk.wikipedia.org/wiki/OS/2)(близько [1989 року](http://uk.wikipedia.org/wiki/1989)), яка фактично була еквівалентом Sybase SQL Server 3.0 для [Unix](http://uk.wikipedia.org/wiki/Unix), [VMS](http://uk.wikipedia.org/w/index.php?title=VMS&action=edit&redlink=1) та ін. Microsoft SQL Server 4.2 був випущений у [1992 році](http://uk.wikipedia.org/wiki/1992) та входив до складу операційної системи [Microsoft OS/2](http://uk.wikipedia.org/w/index.php?title=Microsoft_OS/2&action=edit&redlink=1) версії 1.3. Офіційний реліз Microsoft SQL Server версії 4.21 для ОС [Windows NT](http://uk.wikipedia.org/wiki/Windows_NT) відбувся одночасно з релізом самої [Windows NT](http://uk.wikipedia.org/wiki/Windows_NT) (версії 3.1). Microsoft SQL Server 6.0 був першою версією SQL Server, створеною виключно для архітектури NT і без участі в процесі розробки Sybase. До того часу, як вийшла на ринок ОС Windows NT, Sybase і Microsoft розійшлися та створювали вже власні моделі цього програмного продукту. Microsoft намагалася отримати виняткові права на всі версії SQL Server для [Windows](http://uk.wikipedia.org/wiki/Microsoft_Windows). Пізніше Sybase змінила назву свого продукту на [Adaptive Server Enterprise](http://uk.wikipedia.org/w/index.php?title=Adaptive_Server_Enterprise&action=edit&redlink=1) щоб уникнути плутанини з Microsoft SQL Server. До [1994 року](http://uk.wikipedia.org/wiki/1994) Microsoft отримала від [Sybase](http://uk.wikipedia.org/wiki/Sybase) три повідомлення про авторські права як натяк на походження Microsoft SQL Server.

Після розділення компанії зробили декілька самостійних релізів програм. SQL Server 7.0 був першим сервером баз даних зі справжнім [графічним інтерфейсом](http://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87%D0%B0) адміністрування. Для усунення претензій з боку Sybase у порушенні авторських прав, весь успадкований код в сьомій версії був переписаний. Це забезпечило також й успіх SQL Server 2000, який був першою редакцією, орієнтованою на архітектуру [IA-64](http://uk.wikipedia.org/wiki/IA-64). Протягом подальших шести років [корпорація Microsoft](http://uk.wikipedia.org/wiki/Microsoft) працювала над вдосконаленням вже існуючої версії SQL Server 2000 доки не збудувала зручнішу систему Microsoft SQL Server 2005. Були вдосконалені продуктивність, кліентські інструменти [інтегрованого середовища розробки](http://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%BE%D0%B2%D0%B8%D1%89%D0%B5_%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B8), а також у декількох додаткових системах, що встановлюються разом із SQL Server 2005. Змінено: інструментарій [процесів керування сховищами даних](http://uk.wikipedia.org/wiki/ETL) (SQL Server Integration Services або [SSIS](http://uk.wikipedia.org/wiki/SQL_Server_Integration_Services)), сервер звітів, сервер [OLAP](http://uk.wikipedia.org/wiki/OLAP) та [інтелекутального аналізу даних](http://uk.wikipedia.org/w/index.php?title=%D0%86%D0%BD%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D1%83%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85&action=edit&redlink=1) ([Analysis Services](http://uk.wikipedia.org/w/index.php?title=Microsoft_Analysis_Services&action=edit&redlink=1)), а також декілька технологій повідомлень, особливо Service Broker та Notification Services.

Microsoft SQL Server як мову запитів використовує версію [SQL](http://uk.wikipedia.org/wiki/SQL), що отримала назву [TRANSACT-SQL](http://uk.wikipedia.org/w/index.php?title=TRANSACT-SQL&action=edit&redlink=1) (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт [ISO](http://uk.wikipedia.org/wiki/ISO) для SQL) з багатьма розширеннями. T-SQL дозволяє використовувати додатковий синтаксис [процедур, що зберігаються](http://uk.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%B0,_%D1%89%D0%BE_%D0%B7%D0%B1%D0%B5%D1%80%D1%96%D0%B3%D0%B0%D1%94%D1%82%D1%8C%D1%81%D1%8F&action=edit&redlink=1) і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком). Microsoft SQL Server та Sybase ASE для взаємодії з мережею використовують протокол рівня застосунка під назвою [Tabular Data Stream](http://uk.wikipedia.org/w/index.php?title=Tabular_Data_Stream&action=edit&redlink=1) (TDS, протокол передачі табличних даних).

##### Хід роботи:

1. Встановити новий екземпляр SQL Server. Для сортування задати латиницю.
2. Налаштувати доступ до SQL Server з інших комп’ютерів в мережі.

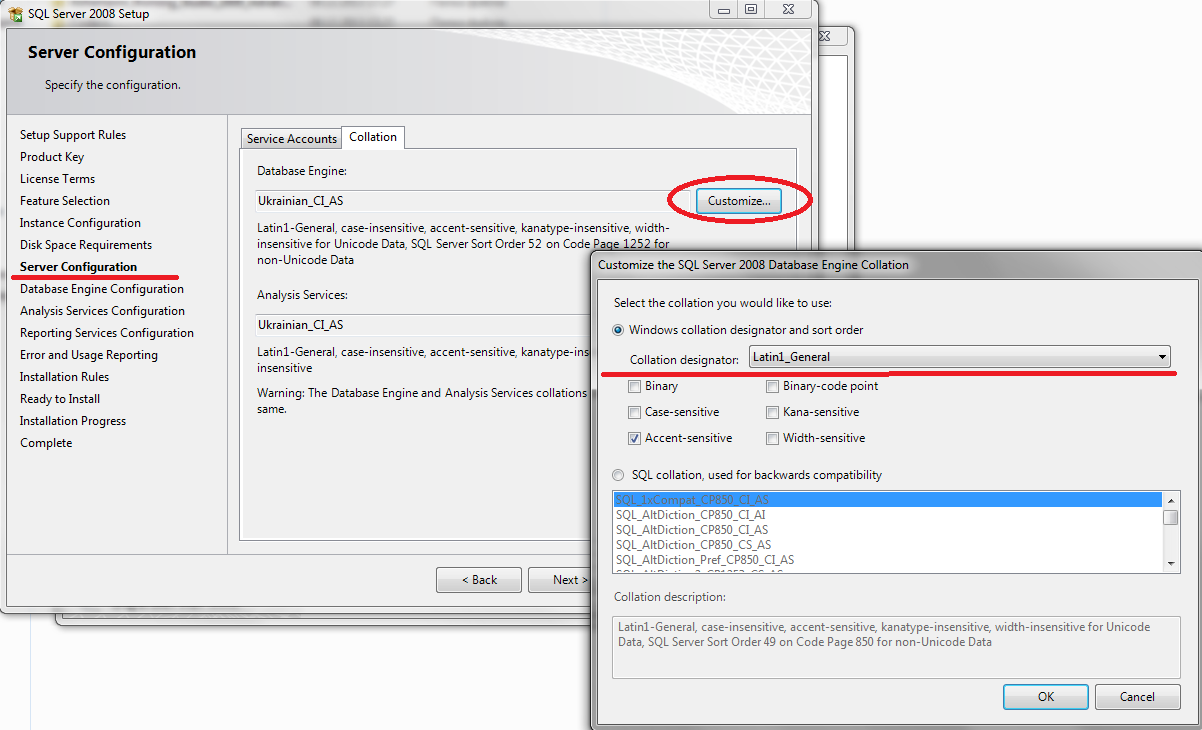
Для того щоб задати потрібний нам порядок сортування, на етапі **«Server Configuration»** установки *SQL Server 2008* на закладці **«Collation»**, змінимо порядок сортування для *Database Engine* та *Analysis Services*, натиснувши кнопку **«Customize..»** (рис. 1). 

Рис. 1. Зміна порядку сортування

У списку вибору виберемо: **Latin1\_General**. Вигляд вікна установки після вказаних дій зображено на рис. 2.

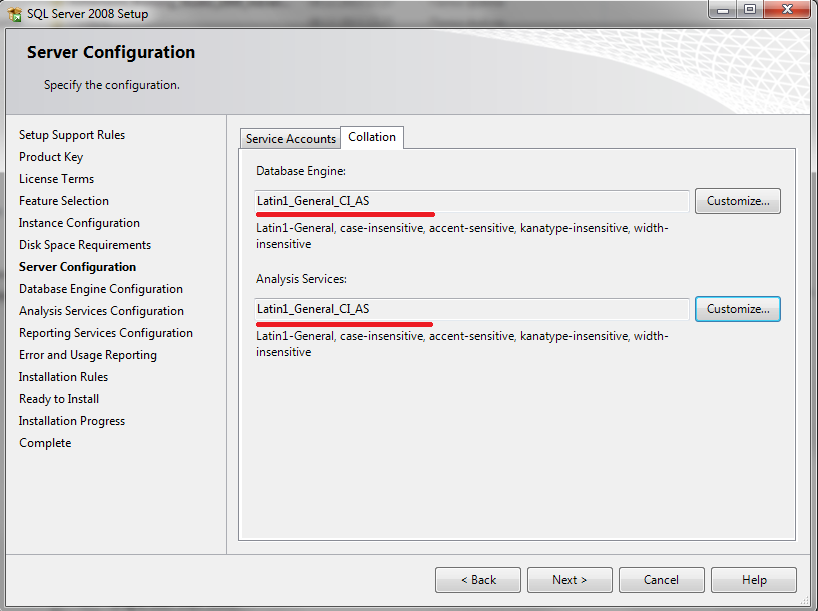


Рис. 2. Вікно установки після вибору сортування

Для налаштування доступу до SQL Server на хості 192.168.1.101 з інших комп’ютерів в мережі, виконаємо наступне**:** запуститимо на цьому комп’ютері утиліту *SQL Server Configuration Manager*. В розділі **SQL Server Network Configuration**–>**Protocols for MSSQLSERVER** включимо протоколи **TCP/IP** та **Named Pipes** за допомогою контекстного меню та його команди: **Enable** (рис. 3).

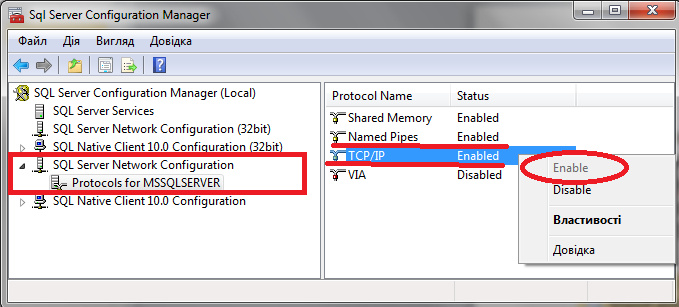


Рис. 3. Включення протоколів TCP/IP та Named Pipes

Зайдемо в гілку **SQL Server Services** та включимо за допомогою контекстного меню сервіс: **SQL Server Agent** (рис. 4).

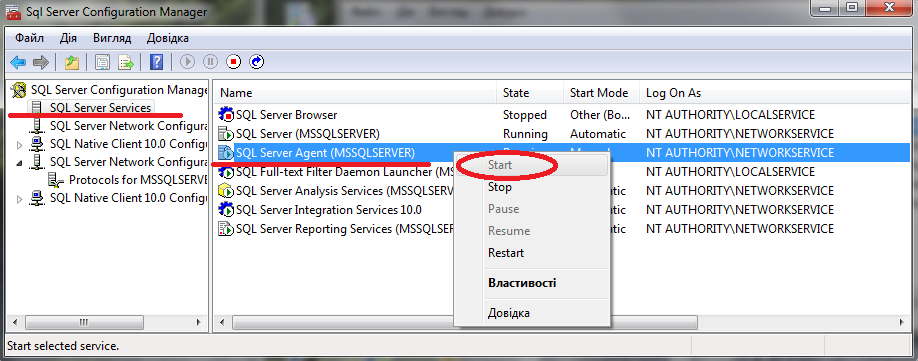


Рис. 4. Включення SQL Server Agent

Для того, щоб зміни вступили в силу, потрібно перезавантажити *SQL Server*. Для цього, скористаємось командою **Restart** контекстного меню пункту **SQL Server (MSSQL Server)** (рис. 5).

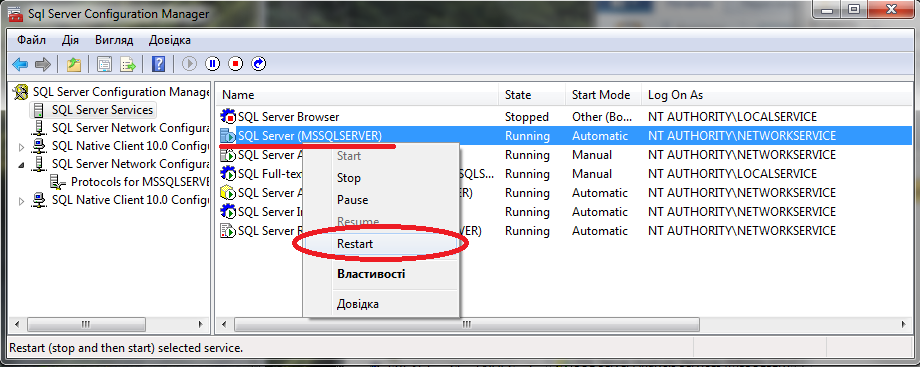


Рис. 5. Перезавантаження SQL Server

Спробуємо з іншої машини в мережі (TarasHP - 192.168.1.100) зайти на *SQL Server* комп’ютера 192.168.1.101. Для цього зайдемо в *Microsoft SQL Server Management Studio* на комп’ютері 192.168.1.100 і у вікні *Object Explorer* натиснемо на кнопку: *Connect Object Explorer*. У вікні з’єднання вкажемо ip-адресу сервера, логін та пароль на натискаємо **Connect** (рис. 6).

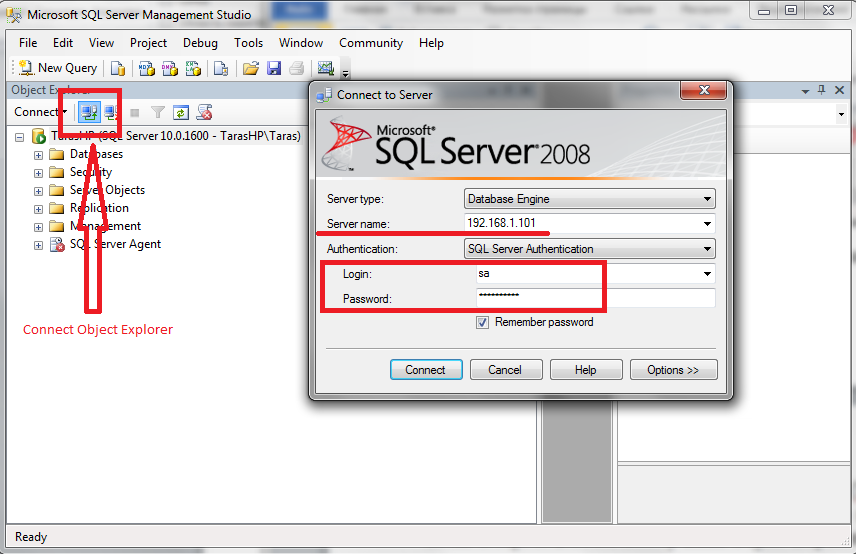


Рис. 6. Підключення до SQL Server на 192.168.1.101

У вікні *Object Explorer* з’явиться нове підключення (рис. 7)

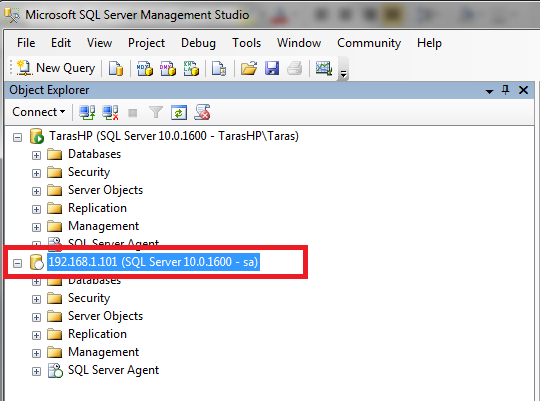


Рис. 7. Нове підключення до SQL Server на 192.168.1.101

**Лабораторна робота №2**

**Мета роботи:** Cтворення бази даних засобами графічного інтерфейсу програми Management Studio: створення таблиць, створення зв’язків між таблицями.

##### Хід роботи:

Завдання

1. В СУБД MS SQL Server створити навчальну базу даних «Операції купівлі-продажу». Назва бази даних повинна містити прізвище власника.
2. В СУБД MS SQL Server за допомогою графічного інтерфейсу створити таблиці бази даних «Операції купівлі-продажу» та базу даних індивідуального завдання.
3. В СУБД MS SQL Server за допомогою графічного інтерфейсу побудувати діаграму бази даних «Операції купівлі-продажу» та базу даних індивідуального завдання.
4. Оформити звіт.

Salers

|  |  |  |  |
| --- | --- | --- | --- |
| snum | sname | city | comm |
| 1001 | Peel | London | .12 |
| 1002 | Serres | San Jose | .13 |
| 1004 | Motika | London | .11 |
| 1007 | Rifkin | Barcelona | .15 |
| 1003 | Axelrod | New York | .10 |

Customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cnum | cname | city | rating | snum |
| 2001 | Hoffman | London | 100 | 1001 |
| 2002 | Giovanni | Rome | 200 | 1003 |
| 2003 | Liu | SanJose | 200 | 1002 |
| 2004 | Grass | Berlin | 300 | 1002 |
| 2006 | Clemens | London | 100 | 1001 |
| 2008 | Cisneros | SanJose | 300 | 1007 |
| 2007 | Pereira | Rome | 100 | 1004 |

Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| onum | amt | odate | cnum | snum |
| 3001 | 18.69 | 03-09-2013 | 2008 | 1007 |
| 3003 | 767.19 | 03-09-2013 | 2001 | 1001 |
| 3002 | 1900.10 | 03-09-2013 | 2007 | 1004 |
| 3005 | 5160.45 | 03-09-2013 | 2003 | 1002 |
| 3006 | 1098.16 | 03-09-2013 | 2008 | 1007 |
| 3009 | 1713.23 | 04-09-2013 | 2002 | 1003 |
| 3007 | 75.75 | 04-09-2013 | 2004 | 1002 |
| 3008 | 4723.00 | 05-09-2013 | 2006 | 1001 |

Створимо таблицю **SalesGrynchuk** використавши команду: *Databases\New Database…* (рис. 1-2) :

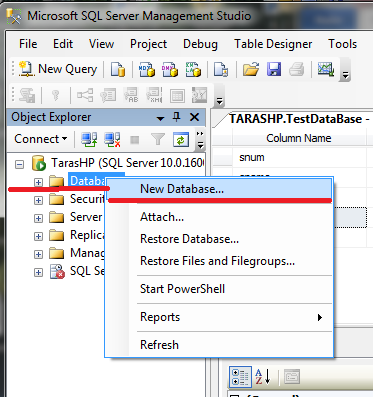


Рис. 1. Команда *Databases\New Database…*

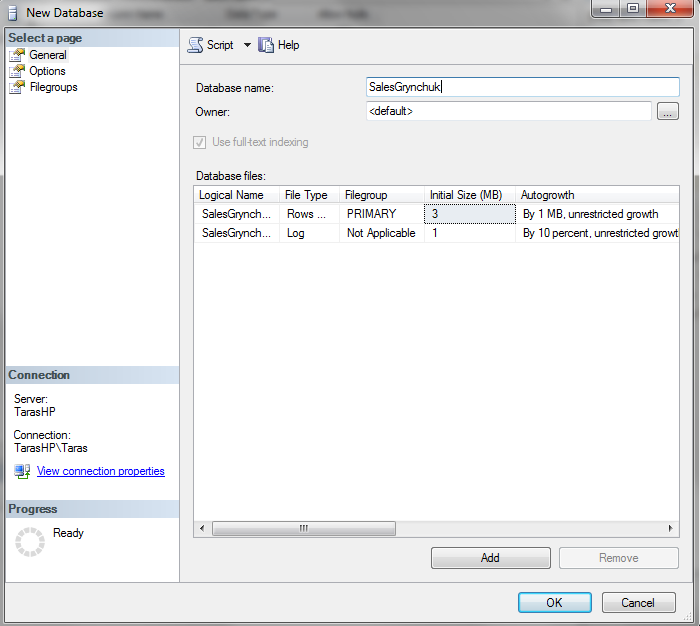


Рис. 2. Задання назви БД

Створимо таблицю Salers по команді: SalesGrynchuk\Tables\New Table… (рис. 3):

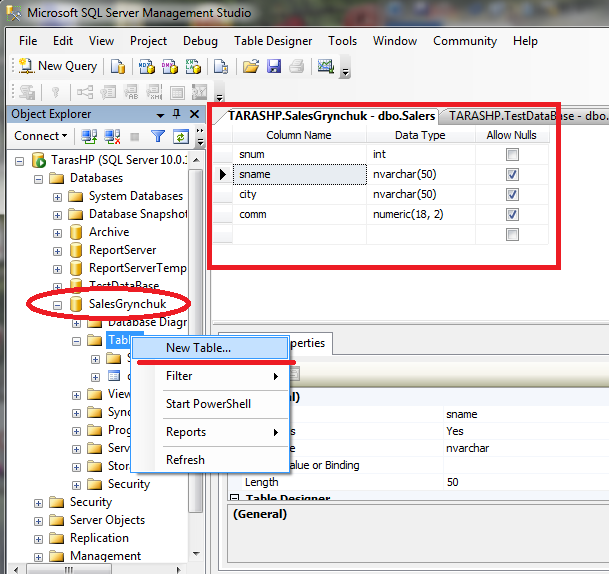


Рис. 3. Створення таблиці Salers

Створимо первинний ключ таблиці, вставши на поле **snum** натиснемо на кнопку **Set Primary Key** (рис. 4):

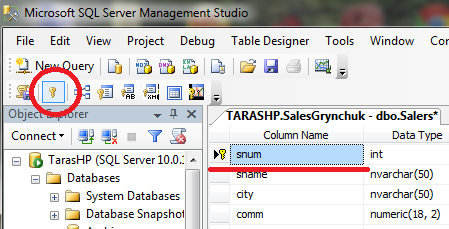


Рис. 4. Встановлення первинного ключа

За допомогою команди *Select Top 100 Rows* введемо рядки таблиці (рис. 5):

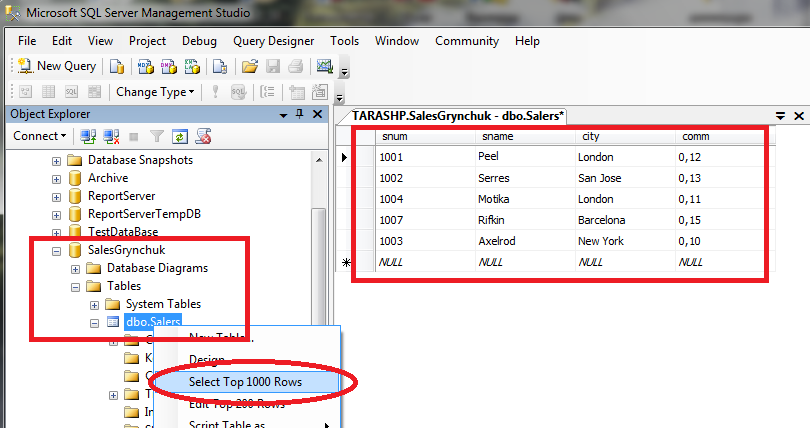


Рис. 5. Введення рядків таблиці **Salers**

Аналогічно створимо решту таблиць бази даних. Для того, щоб створити діаграму зв’язків між таблицями БД, командою контекстного меню Database Diagrams/New Database Diagram (рис. 6):

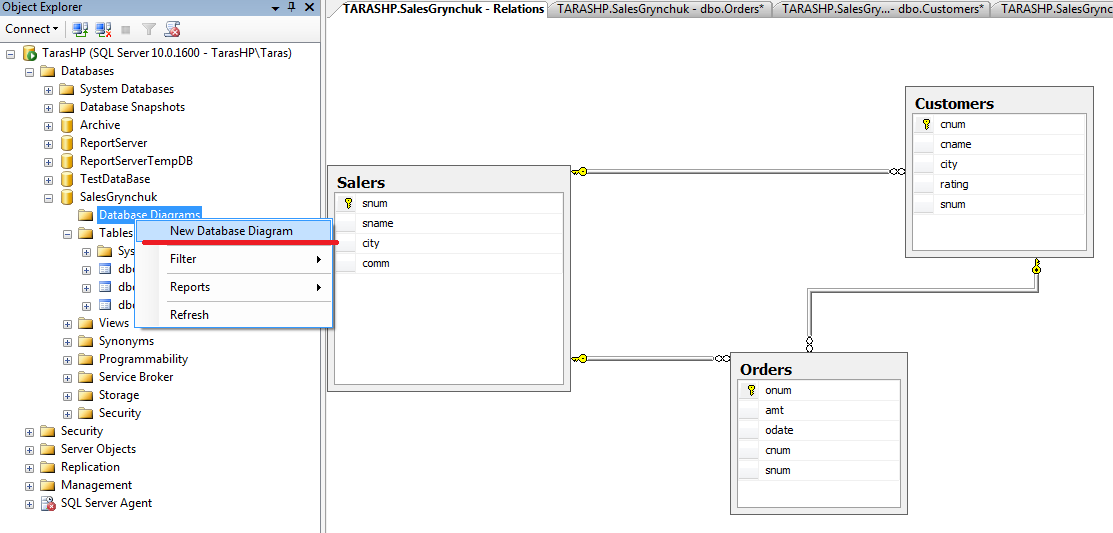


Рис. 6. Діаграма бази даних

**Лабораторна робота №3**

***Мета:*** Вивчення синтаксису інструкції SELECT, отримання практичних навиків написання однотабличних запитів.

**Хід роботи**

**Вправа 1.** Модифікуйте запит з прикладу 1, щоб результатом було повернення лише даних про ім’я замовника та його розміщення.

SELECT

\*

FROM Customers

Результатом буде запит:

SELECT

cname, city

FROM Customers

Виконання зображено на рис. 1.

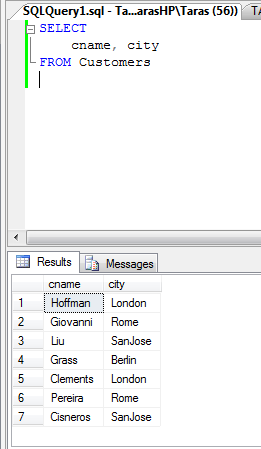


Рис. 1. Запит до *вправи 1*

**Вправа 2.** Модифікуйте запит з прикладу 2, так щоб результатом було повернення списку замовників розміщених у Лондоні або Римі.

SELECT

cname,

city,

rating

FROM Customers

WHERE

city = 'London'

Результатом буде запит:

SELECT

cname,

city,

rating

FROM Customers

WHERE

city = 'London' or city = 'Rome'

Виконання зображено на рис. 2.

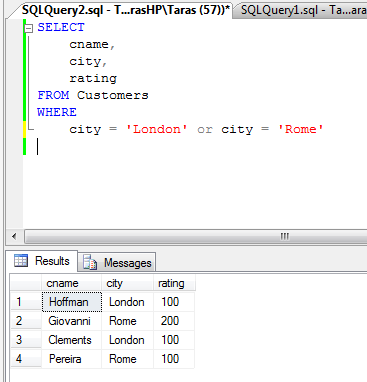
****

Рис. 2. Запит до *вправи 2*

**Вправа 3.** Модифікуйте запит з прикладу 2, так щоб результатом було повернення списку замовників розміщених за межами Лондона та із рівнем переваги понад 200.

Результатом буде запит:

SELECT

cname,

city,

rating

FROM Customers

WHERE

city <> 'London' and rating > 200

Виконання зображено на рис. 3.

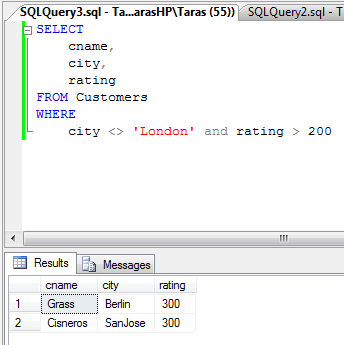


Рис. 3. Запит до *вправи 3*

**Вправа 4.** Модифікуйте запит з прикладу 2, так щоб результатом було повернення списку замовників імена, яких починаються на “Gr” .

Результатом буде запит:

SELECT

cname,

city,

rating

FROM Customers

WHERE

cname like 'Gr%'

Виконання зображено на рис. 4.

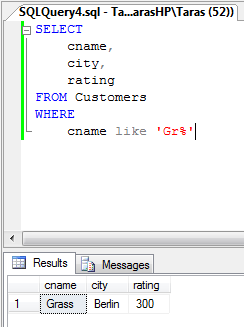


Рис. 4. Запит до *вправи 4*

**Вправа 5.** Модифікуйте запит з прикладу 2, так щоб результатом було повернення списку замовників у іменах, яких більше 5-ти букв.

Результатом буде запит:

SELECT

cname,

city,

rating

FROM Customers

WHERE

LEN(cname) > 5

Виконання зображено на рис. 5.

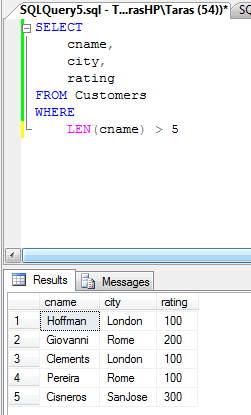


Рис. 5. Запит до *вправи 5*

**Вправа 6.** Модифікуйте запит з прикладу 3, так щоб результатом було повернення списку замовників, що не співпрацюють із продавцем Peel, а також із продавцями, у яких комісія перевищує 0,13 .

SELECT cname,city,rating

FROM Customers

WHERE snum NOT IN

(

SELECT snum

FROM Salers

WHERE sname = 'Peel')

Результатом буде запит:

SELECT

cname,

city,

rating

FROM Customers

WHERE

snum NOT IN (

SELECT

snum

FROM Salers

WHERE

sname = 'Peel' or comm > 0.13

)

Виконання зображено на рис. 6.

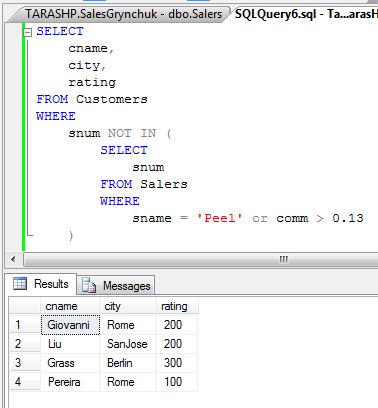


Рис. 6. Запит до *вправи 6*

**Вправа 7.** Модифікуйте запит з прикладу 3, так щоб результатом було повернення списку замовників, що не здійснювали операцію купівлі продажу після 04-09-2013.

Результатом буде запит:

SELECT

cname,

city,

rating

FROM Customers

WHERE

cnum NOT IN (

SELECT

cnum

FROM Orders

WHERE

odate > '2013-09-04'

)

Виконання зображено на рис. 7.

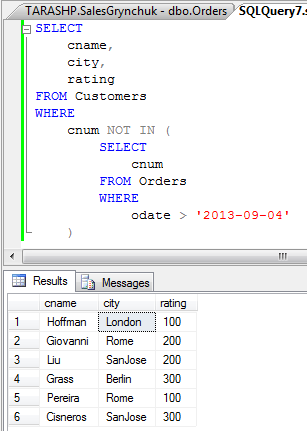


Рис. 7. Запит до *вправи 7*

**Вправа 8.** Реалізуйте запит, результатом якого є об’єднаний список замовників із вправи 6 та вправи 7 (підказка: для об’єднання результатів двох запитів застосуйте оператор UNION).

Результатом буде запит:

SELECT

cname,

city,

rating

FROM Customers

WHERE

snum NOT IN (

SELECT

snum

FROM Salers

WHERE

sname = 'Peel' or comm > 0.13

)

union

SELECT

cname,

city,

rating

FROM Customers

WHERE

cnum NOT IN (

SELECT

cnum

FROM Orders

WHERE

odate > '2013-09-04'

)

Виконання зображено на рис. 8.

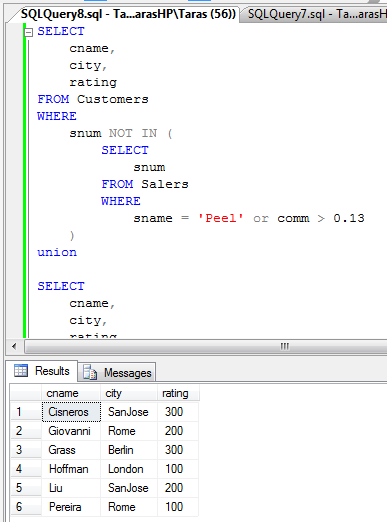


Рис. 8. Запит до *вправи 8*

**Вправа 9.** Спробуйте виконати запиит, поясніть помилку, що виникла.

SELECT

snum,

cname,

COUNT(cnum)

FROM Customers

GROUP BY

Snum

При виконанні виникла помилка: **Column 'Customers.cname' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.** Це пояснюється тим, що ми не вказали назву агрегатної функції. В запит потрібно внести наступні зміни:

SELECT

snum,

cname,

COUNT(cnum) as N

FROM Customers

GROUP BY

snum,

cname

Виконання зображено на рис. 9.

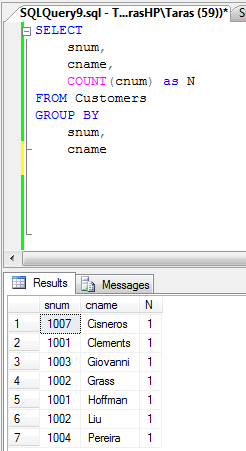


Рис. 9. Запит до *вправи 9*

**Вправа 10.** Реалізуйте запит, котрий підрахує кількість операцій купівлі продажу за кожен з днів.

Результатом буде запит:

SELECT

odate,

COUNT(odate) as N

FROM Orders

GROUP BY

Odate

Виконання зображено на рис. 10.

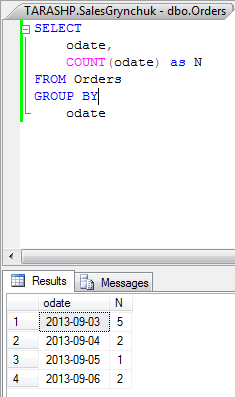


Рис. 10. Запит до *вправи 10*

**Вправа 11.** Розширте запит із вправи 10, так щоб було враховано лише продавців із комісійними неменшими, аніж 0,12.

Результатом буде запит:

SELECT

odate,

COUNT(odate) as N

FROM Orders

WHERE snum in

(

SELECT snum

FROM Salers

WHERE comm >= 0.12

)

GROUP BY

Odate

Виконання зображено на рис. 11.

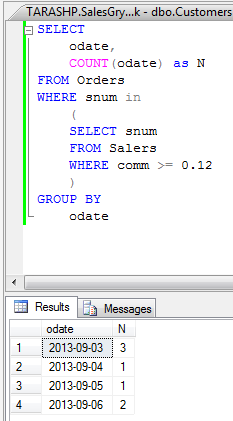


Рис. 11. Запит до *вправи 11*

**Вправа 12.** Модифікуйте запит з прикладу 6, так щоб при підрахунку кількості замовників були враховані лише замовники із кодом переваги вищим за 200.

SELECT

snum,

COUNT(cnum)

FROM Customers

GROUP BY

snum

HAVING

COUNT(cnum) > 2

Результатом буде запит:

SELECT

rating,

COUNT(cnum) AS N

FROM Customers

GROUP BY

rating

HAVING (rating) > 200

Виконання зображено на рис. 12.

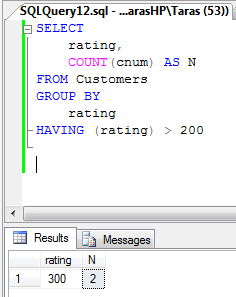


Рис. 12. Запит до *вправи 12*

**Вправа 13.** Реалізуйте запит, що видасть результати про сумарну суму усіх операцій купівлі продажу для кожного дня і по кожному продавцю. У результуючий набір повинні увійти лише ті дні, коли сума операцій перевищила 3000, а до розгляду було прийнято лише операції купівлі-продажу на суму вищу за 100.

Результатом буде запит:

SELECT

odate,

snum,

SUM(amt) as SumOfSales

FROM Orders

WHERE amt > 100

GROUP BY

odate,

snum

HAVING SUM(amt) > 3000

Виконання зображено на рис. 13.

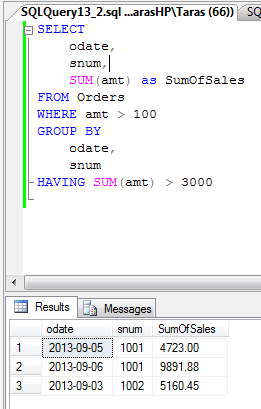


Рис. 13. Запит до *вправи 13*

**Лабораторна робота №4**

***Мета:*** Вивчення основних принципів та типів об’єднання у багатотабличних запитах на вибірку.

**Хід роботи**

***Приклад 1.*** Запит до навчальної бази даних, який повертає для усіх операцій купівлі-продажу на суму біль 1000, ім’я замовника, розміщення замовника, суму операції купівлі-продажу та її дату.

SELECT

cus.cname,

cus.city,

ord.amt,

ord.odate

FROM Customers cus

JOIN Orders ord

ON cus.cnum = ord.cnum

WHERE

ord.amt > 1000

Результатом зображено на рис. 1:

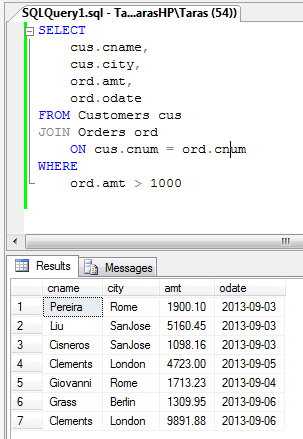


Рис. 1. Запит до *прикладу 1*

***Приклад 2.*** Запит до навчальної бази даних, який повертає для усіх операцій купівлі-продажу на суму біль 1000, ім’я замовника, розміщення замовника, суму операції купівлі-продажу та її дату. Умову об’єднання вказано у виразі WHERE.

SELECT

cus.cname,

cus.city,

ord.amt,

ord.odate

FROM Customers cus, Orders ord

WHERE

ord.amt > 1000

AND cus.cnum = ord.cnum

Результатом зображено на рис. 2:

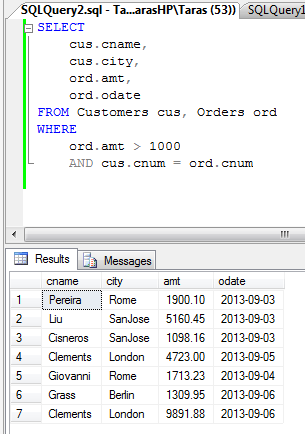


Рис. 2. Запит до *прикладу 2*

***Приклад 3.*** Запит до навчальної бази даних, який повертає дані про усіх продавців та проведені ними операції-купівлі продажу. Продавці, котрі не провели жодної операції купівлі продажу теж потраплять у список вибірки. (Для того, щоб спостерігати відмінності між результатами наведеного нижче скрипта та скрипта із застосуванням операції внутрішнього об’єднання рекомендується у таблицю **Salers** додати нового продавця).

SELECT

sal.sname,

sal.comm,

ord.amt,

ord.odate

FROM Salers sal

LEFT JOIN Orders ord

ON sal.snum = ord.snum

Результатом зображено на рис. 3:

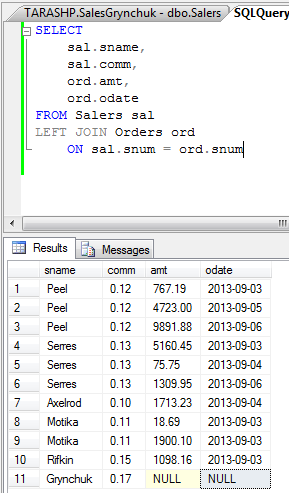


Рис. 3. Запит до *прикладу 3*

***Приклад 4.*** Запит до навчальної бази даних, який повертає дані про операції купівлі-продажу проведені продавцем із замовниками, які йому не призначені.

SELECT

cus.cname,

ord.amt,

ord.odate

FROM Customers cus

JOIN Orders ord

ON cus.cnum = ord.cnum

AND cus.snum != ord.snum

Результатом зображено на рис. 4:

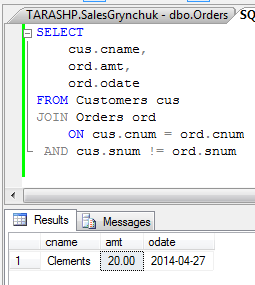


Рис. 4. Запит до *прикладу 4*

***Приклад 5.*** Запит до навчальної бази даних, що для усіх проведених операцій купівлі-продажу повертає дані про продавця, покупця та суму операції.

SELECT

sal.sname,

ord.amt,

cus.cname

FROM Salers sal

JOIN Orders ord

ON sal.snum = ord.snum

JOIN Customers cus

ON ord.cnum = cus.cnum

Результатом зображено на рис. 5:

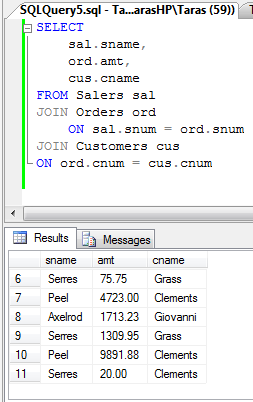


Рис. 5. Запит до *прикладу 5*

**Лабораторна робота №6**

***Мета:*** Вивчення синтаксису створення та модифікації віртуальних таблиць. Засвоєння принципів представлення даних для різних груп користувачів.

**Хід роботи**

Завдання:

1. Виконати усі вправи вказані у вказівках до лабораторної роботи (у звіті вказати усі зроблені при цьому зміни у структурі навчальної бази даних).
2. Реалізувати віртуальні таблиці для бази індивідуального завдання, орієнтовані на спрощення сприйняття інформації користувачем (багатотабличні запити).
3. Реалізувати підсистему розмежування прав доступу до даних у базі індивідуального завдання на основі віртуальних таблиць. Врахувати, що різні користувачі можуть мати різні рівні повноважень (див. вправу 3).
4. Оформити звіт (у звіті вказати структуру бази даних індивідуального завдання до і після реалізації розмежування доступу до даних).

***Вправа 1.*** Доповніть перелік полів віртуальної таблиці полями з таблиці Salers. Дослідіть можливість вставки та модифікації даних засобами такої віртуальної таблиці.

Створимо новий логін: “Serres” (рис. 1-2):

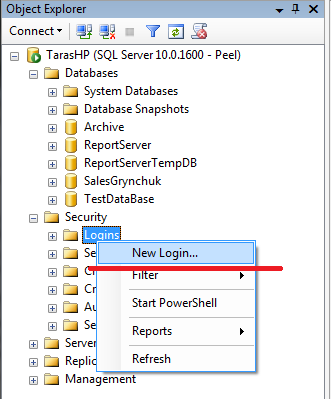


Рис. 1. Команда створення логіна

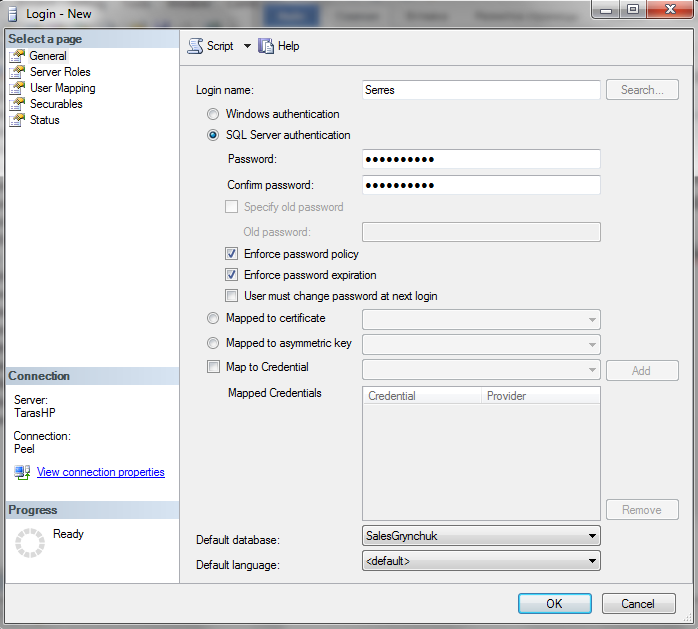


Рис. 2. Створення логіна

Тепер створимо відповідного користувача для нашої бази даних (рис. 3-4):

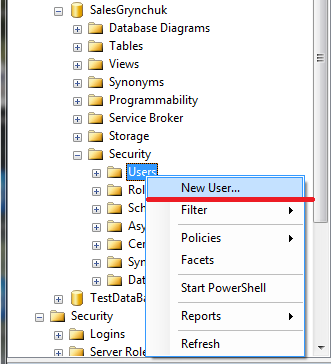


Рис. 3. Команда створення користувача БД

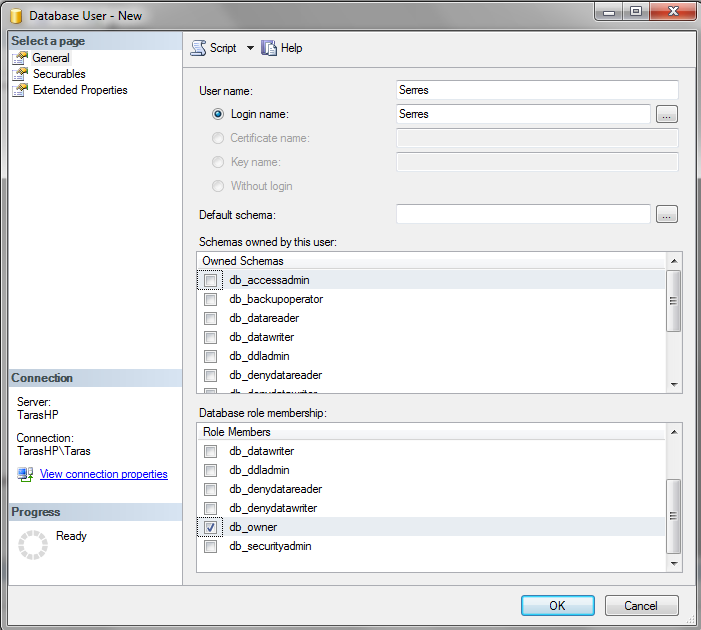


Рис. 4. Створення користувача БД

Підключимося до БД за під даним користувачем і наступним запитом створимо для нього віртуальну таблицю *vCustomers* (рис. 5):

CREATE VIEW [dbo].[vCustomers]

AS

SELECT

cus.cname as Customer,

cus.city CustomerCity,

cus.rating,

sal.comm as SalerProfit,

sal.city as SalerCity

FROM dbo.Customers cus

JOIN Salers sal

ON cus.snum = sal.snum

WHERE

sal.sname = SUSER\_SNAME()

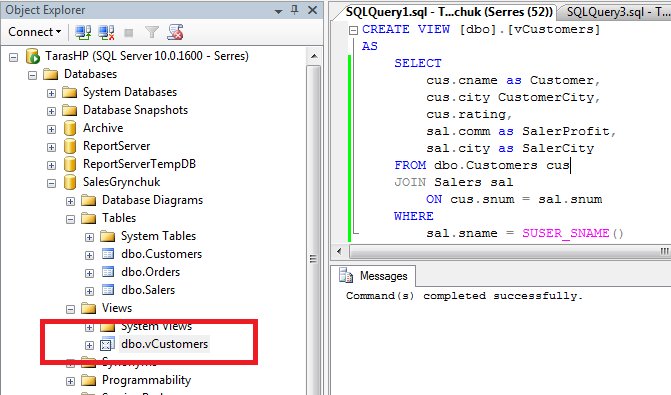


Рис. 5. Створення таблиці *vCustomers*

Для перевірки результату виконаємо простий запит до цієї таблиці (рис. 6):

SELECT

\*

FROM vCustomers

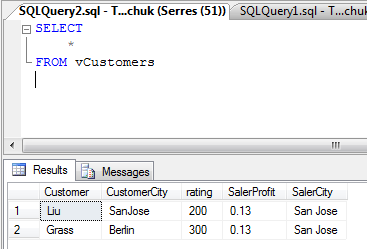


Рис. 6. Результат запиту до *vCustomers*

Відкриємо дану таблицю для редагування. Зауважимо, що дані можна змінювати, але додати нові колонки не вдасться (рис. 7):

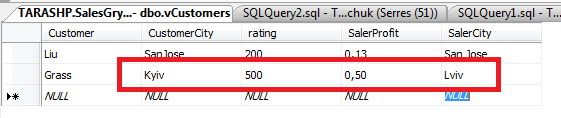


Рис. 7. *vCustomers* після редагування

Виконаємо попередній запит до модифікованих даних (рис. 8):

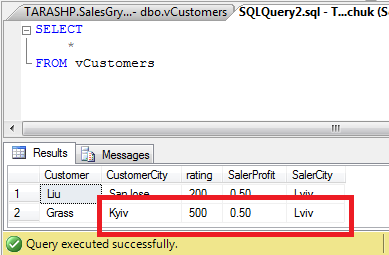


Рис. 8. Виконання запиту до *vCustomers*

***Вправа 2.*** Запропонуйте та реалізуйте модифікації (можливо кілька варіантів) навчальної бази даних, щоб кожному замовнику можливо було призначити кілька альтернативних продавців. Для модифікованої бази даних створіть аналогічну до vCustomers віртуальну таблицю, що забезпечить для продавця можливість роботи з усіма замовниками, яким його призначено. Врахуйте, що кількість продавців, які призначені замовнику, може бути довільною.

Створимо ще одну таблицю *Relations*, для проставлення відношень продавців до покупців (рис. 9):



Рис. 9. Таблиця *Relations*

Створимо користувача *Serres* та побудуємо для нього віртуальну таблицю *vSalersCustomers*. Для цього підключимося під цим користувачем до бази і виконаємо наступний скрипт (рис. 10):

CREATE VIEW [dbo].[vSalersCustomers]

AS

SELECT

cus.cname as Customer,

cus.city CustomerCity,

cus.rating,

sal.sname as Saler,

sal.comm as SalerProfit,

sal.city as SalerCity

FROM dbo.Customers cus

JOIN Salers sal

ON cus.snum = sal.snum

WHERE

cus.cnum in (

SELECT

cnum

FROM Relations

WHERE

sname = SUSER\_SNAME()

)

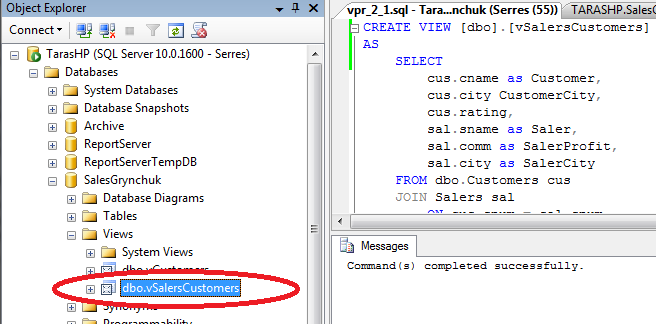


Рис. 10. Створення таблиці *vSalersCustomers*

Перевіримо результат наступним запитом (рис. 11):

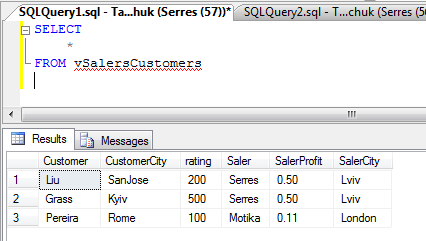


Рис. 11. Вибірка всіх даних з таблиці *vSalersCustomers*

Таким чином продавець *Serres* може працювати з цією віртуальною таблицею.

***Вправа 3.*** Запропонуйте та реалізуйте модифікації навчальної бази даних (можливо кілька варіантів), щоб для ряду продавців було можливим задати особливий статус (керівника підприємства), а саме забезпечити для них доступ до інформації про усіх замовників. Для модифікованої бази даних створіть аналогічну до vCustomers віртуальну таблицю, що забезпечить виконання поставленої задачі.

Створимо таблицю керівників *ChiefSalers* (рис. 12):

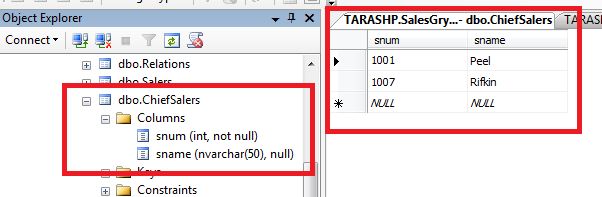


Рис.12. Таблиця *ChiefSalers*

Доповнимо попередній запит, умовою, якщо користувач в списку керівників то йому формується таблиця зі всіма записами, цей запит матиме вигляд:

CREATE VIEW [dbo].[vNewSalersCustomers]

AS

SELECT

cus.cname as Customer,

cus.city CustomerCity,

cus.rating,

sal.sname as Saler,

sal.comm as SalerProfit,

sal.city as SalerCity

FROM dbo.Customers cus

JOIN Salers sal

ON cus.snum = sal.snum

WHERE

cus.cnum in (

SELECT

cnum

FROM Relations

WHERE

sname = SUSER\_SNAME()

)

union

SELECT

cus.cname as Customer,

cus.city CustomerCity,

cus.rating,

sal.sname as Saler,

sal.comm as SalerProfit,

sal.city as SalerCity

FROM dbo.Customers cus

JOIN Salers sal

ON cus.snum = sal.snum

WHERE

SUSER\_SNAME() in (

SELECT

sname

FROM ChiefSalers

)

Зайдемо в базу під користувачем *Serres* та запустимо його, віртуальна таблиця матиме вигляд (рис. 13):

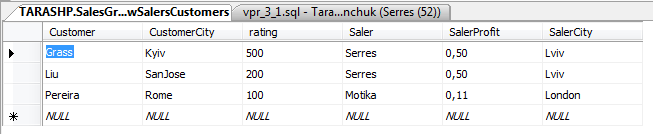


Рис. 13. Віртуальна таблиця для користувача Serres

Як бачимо вона містить лише 3 записи, як і в попередній вправі, оскільки цей користувач не є керівником. Зайдемо під користувачем *Peel* і сформуємо таблицю для нього (рис. 14):

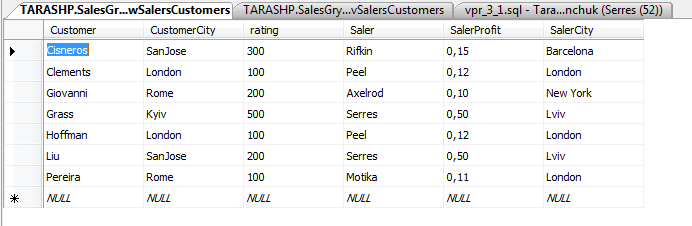


Рис. 14. Віртуальна таблиця для користувача Peel

Як бачимо таблиця містить всі можливі записи, оскільки користувач є в списку керівників.