МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ

****

ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №6

на тему:

**«Віртуальні таблиці в MS SQL Server»**

з курсу:

**"Бази даних"**

**ВИКОНАВ:**

слухач групи ПЗС-11

Гринчук Т.А.

**ПЕРЕВІРИЛА:**

доц. Павич Н.Я.

Львів – 2014

***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):

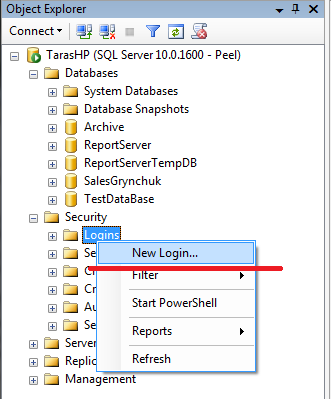


Рис. 1. Команда створення логіна

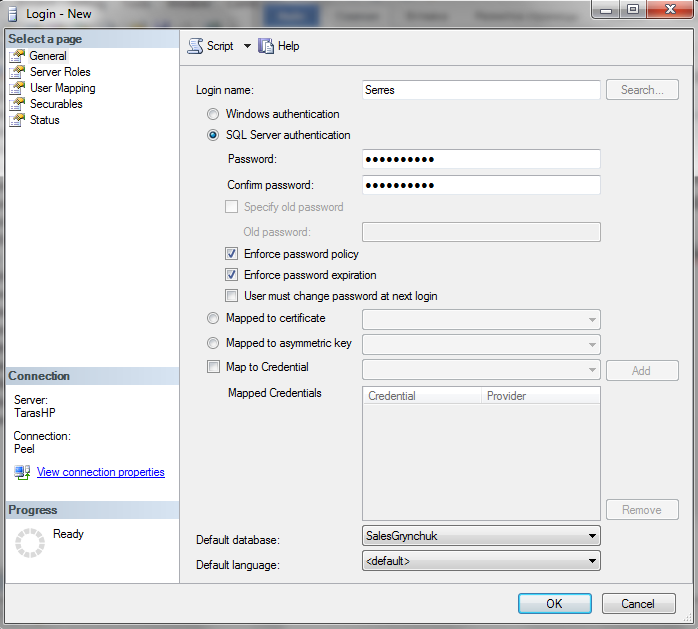


Рис. 2. Створення логіна

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

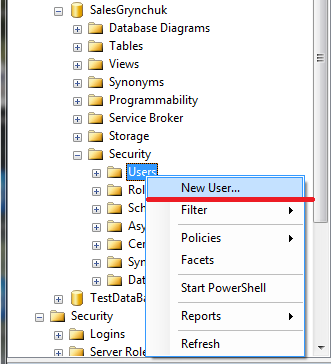


Рис. 3. Команда створення користувача БД

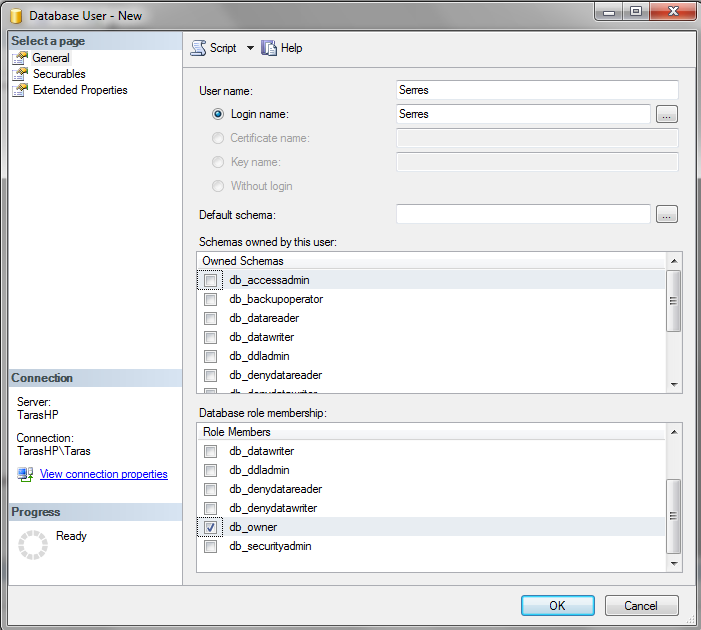


Рис. 4. Створення користувача БД

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()

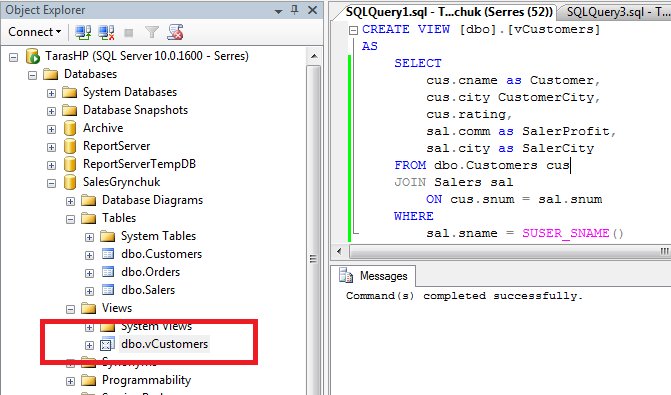


Рис. 5. Створення таблиці *vCustomers*

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

SELECT

\*

FROM vCustomers

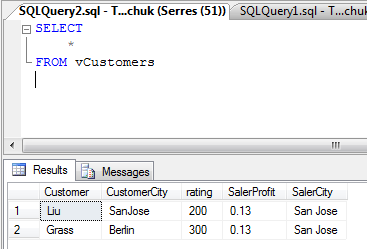


Рис. 6. Результат запиту до *vCustomers*

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

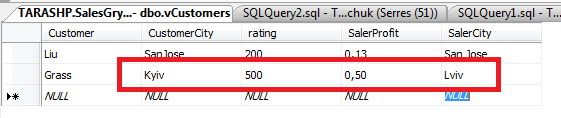


Рис. 7. *vCustomers* після редагування

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

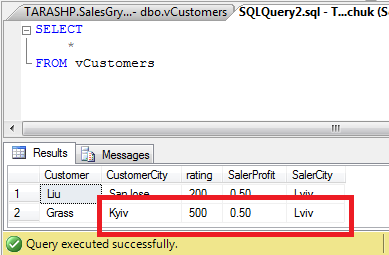


Рис. 8. Виконання запиту до *vCustomers*

***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()

)

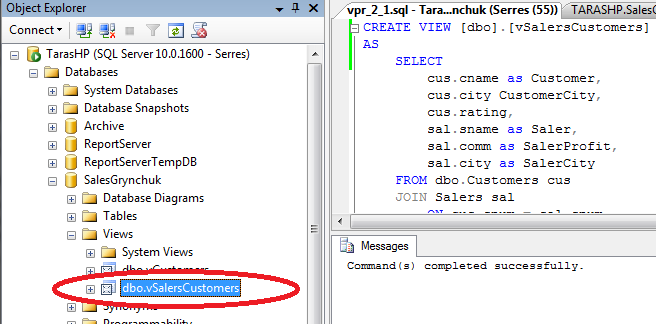


Рис. 10. Створення таблиці *vSalersCustomers*

Перевіримо результат наступним запитом (рис. 11):

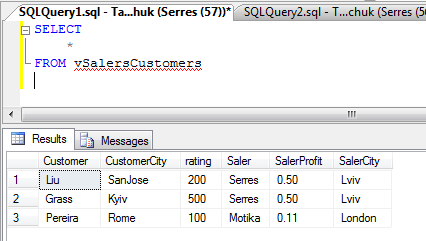


Рис. 11. Вибірка всіх даних з таблиці *vSalersCustomers*

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):

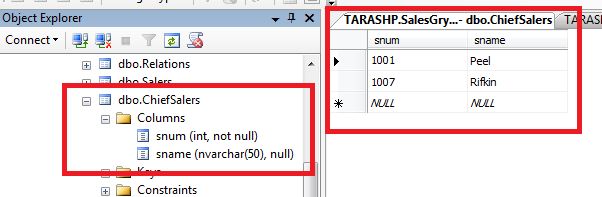


Рис.12. Таблиця *ChiefSalers*

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):

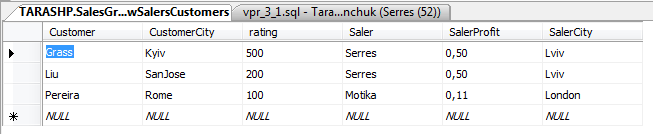


Рис. 13. Віртуальна таблиця для користувача Serres

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):

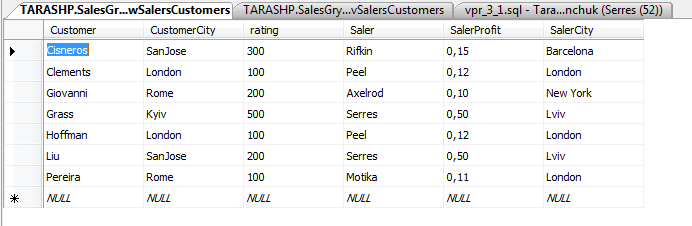


Рис. 14. Віртуальна таблиця для користувача Peel

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

**Висновки:**

На даній лабораторній роботі я вивчив синтаксис створення та модифікації віртуальних таблиць. Засвоїв принципи представлення даних для різних груп користувачів.