

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Методичні вказівки
до виконання лабораторної роботи за темою «Створення та
використання представлень (view) засобами СУБД Microsoft SQL
Server»

для студентів, що навчаються за спеціальностями
«121 Інженерія програмного забезпечення»
«122 Комп'ютерні науки»
«126 Інформаційні системи та технології»

Затверджено
редакційно-видавничою
радою університету,
протокол № 2 від 16.06.2023 р.

Харків
НТУ ХПІ
2023

Методичні вказівки до виконання лабораторної роботи за темою «Створення та використання представлень (view) засобами СУБД Microsoft SQL Server» для студентів, що навчаються за спеціальностями 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 126 «Інформаційні системи та технології» / Уклад. Орловський Д.Л., Копп А.М. – Харків: НТУ «ХПІ», 2023. – 24 с.

Укладачі Д.Л. Орловський,
А.М. Копп

Рецензент Гринченко М.А.

Кафедра програмної інженерії та інтелектуальних технологій управління

ЗМІСТ

Вступ.....	4
1 Теоретичні відомості.....	5
1.1 Поняття представлення.....	5
1.2 Оновлення даних в представленнях.....	7
1.3 Переваги і недоліки представлень.....	8
2 Виконання роботи	13
2.1 Побудова та використання представлення для перегляду даних.....	13
2.2 Оновлення даних за допомогою представлень	16
2.2.1 Використання параметру WITH CHECK OPTION.....	16
2.2.2 Використання триггеру INSTEAD OF	19
2.3 Збереження результатів роботи	21
3 Вимоги до звіту.....	22
4 Питання для самоперевірки.....	23
Список літератури	24

ВСТУП

Сучасні інформаційні системи ґрунтуються на використанні баз даних, в яких накопичується різна інформація. Тому зараз розробляються і значно поширюються методи і засоби роботи з базами даних з метою підвищення ефективності роботи людини у різних галузях діяльності. Ці засоби та методи пов'язані з узагальненням і різними додатковими способами обробки даних. Основні ідеї сучасної інформаційної технології базуються на концепції, згідно з якою дані повинні бути організовані в бази даних з метою адекватного відображення реального світу, що змінюється, і задоволення інформаційних потреб користувачів. Ці бази даних створюються і функціонують під управлінням спеціальних програмних комплексів, які називають системами управління базами даних (СУБД).

Методичні вказівки до лабораторної роботи за темою «Створення та використання представлень (view) засобами СУБД Microsoft SQL Server» призначені для студентів, що навчаються за спеціальностями «121 Інженерія програмного забезпечення», «122 Комп'ютерні науки», «126 Інформаційні системи та технології».

Виконання лабораторної роботи повинно забезпечити закріплення теоретичних знань і практичних навичок, отриманих при вивченні лекційної частини дисциплін, пов'язаних із проектуванням, розробкою та застосуванням баз даних.

В методичних вказівках розглянуті основні питання, пов'язані з теоретичним обґрунтуванням та безпосереднім виконанням лабораторної роботи.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Поняття представлення

Представлення, або уявлення, або перегляди (VIEW), є тимчасовими, похідними (інакше – віртуальні) таблицями і є об'єктами бази даних, інформація в яких не зберігається постійно, як у базових таблицях, а формується динамічно при зверненні до них. Звичайні таблиці відносяться до базових, тобто що містить дані і постійно знаходиться на пристрої зберігання інформації. Представлення не може існувати само по собі, а визначається тільки в термінах однієї або декількох таблиць. Застосування представлень дозволяє розробникові бази даних забезпечити кожному користувачеві або групі користувачів найбільш відповідні способи роботи з даними, що вирішує проблему простоти їх використання і безпеки. Вміст представлень вибирається з інших таблиць за допомогою виконання запиту, причому при зміні значень в таблицях дані в представленні автоматично міняються. Представлення - це фактично той же запит, який виконується всякий раз при участі в якій-небудь команді. Результат виконання цього запиту в кожен момент часу стає змістом представлення. У користувача створюється враження, що він працює із справжньою, реально існуючою таблицею.

У СУБД є дві можливості реалізації представлень. Якщо його визначення просте, то система формує кожен запис представлення в міру необхідності, поступово прочитуючи початкові дані з базових таблиць. У разі складного визначення СУБД доводиться спочатку виконати таку операцію, як матеріалізація представлення, тобто зберегти інформацію, з якої складається представлення, в тимчасовій таблиці. Потім система приступає до виконання призначеної для користувача команди і формування її результатів, після чого тимчасова таблиця видаляється.

Представлення – це зумовлений запит, що зберігається у базі даних, який виглядає подібно до звичайної таблиці і не вимагає для свого зберігання дискової пам'яті. Для зберігання представлення

використовується тільки оперативна пам'ять. На відміну від інших об'єктів бази даних представлення не займає дискової пам'яті за винятком пам'яті, необхідної для зберігання визначення самого представлення.

Створення і зміни представлень в стандарті мови і реалізації в MS SQL Server співпадають і представлені наступною командою:

```
<визначення_представлення> ::=  
    { CREATE| ALTER} VIEW ім'я_представлення  
    [(ім'я_стовпця [...n])]  
    [WITH ENCRYPTION]  
    AS SELECT_оператор  
    [WITH CHECK OPTION]
```

Розглянемо призначення основних параметрів.

За умовчанням імена стовпців в представленні відповідають іменам стовпців в початкових таблицях. Явну вказівку імені стовпця потрібно для обчислюваних стовпців або при об'єднанні декількох таблиць, що мають стовпці з однаковими іменами. Імена стовпців перераховуються через кому, відповідно до порядку їх дотримання в представленні.

Параметр WITH ENCRYPTION наказує серверу шифрувати SQL-код запиту, що гарантує неможливість його несанкціонованого перегляду і використання. Якщо при визначенні представлення необхідно приховати імена початкових таблиць і стовпців, а також алгоритм об'єднання даних, необхідно застосувати цей аргумент.

Параметр WITH CHECK OPTION наказує серверу виконувати перевірку змін, вироблюваних через представлення, на відповідність критеріям, визначеним в операторові SELECT. Це означає, що не допускається виконання змін, які приведуть до зникнення рядка з представлення. Таке трапляється, якщо для представлення встановлений горизонтальний фільтр і зміна даних призводить до невідповідності рядка встановленим фільтрам. Використання аргументу WITH CHECK OPTION

гарантує, що зроблені зміни будуть відображені в представленні. Якщо користувач намагається виконати зміни, що призводять до виключення рядка із представлення, при заданому аргументі WITH CHECK OPTION сервером буде сформовано повідомлення про помилку і усі зміни будуть відхилені.

Представлення можна використовувати в команді SQL так само, як і будь-яку іншу таблицю. До представлення можна будувати запит, модифікувати його (якщо воно відповідає певним вимогам), сполучати з іншими таблицями. Зміст представлення не фіксований і оновлюється кожного разу, коли на нього посилаються в команді. Представлення значно розширюють можливості управління даними. Зокрема, це прекрасний спосіб дозволити доступ до інформації в таблиці, приховавши частину даних.

Таким чином, представлення може змінюватися командами модифікації DML, але фактично модифікація впливає не на само представлення, а на базову таблицю.

Представлення видаляється командою:

```
DROP VIEW ім'я_представлення [,..n]
```

1.2 Оновлення даних в представленнях

Не усі представлення в SQL можуть бути модифіковані. Представлення, що модифікується, визначається наступними критеріями:

- ґрунтується тільки на одній базовій таблиці;
- містить первинний ключ цієї таблиці;
- не містить DISTINCT у своєму визначенні;
- не використовує GROUP BY або HAVING у своєму визначенні;
- по можливості не застосовує у своєму визначенні підзапити;
- не використовує константи або вираження значень серед вибраних полів виводу;

- у представлення має бути включений кожен стовпець таблиці, що має атрибут NOT NULL;
- оператор SELECT перегляду не використовує агрегуючі (підсумкові) функції, з'єднання таблиць, Збережені процедури, і функції, визначені користувачем;
- ґрунтується на поодиноким запиту, тому об'єднання (UNION) не дозволене.

Якщо перегляд задовольняє цим умовам, до нього можуть застосовуватися оператори INSERT, UPDATE, DELETE. Відмінності між представленнями, що модифікуються, і представленнями, призначеними тільки для читання, не випадкові. Цілі, для яких їх використовують, різні. З представленнями, що модифікуються, в основному обходяться точно так, як і з базовими таблицями. Фактично, користувачі не можуть навіть усвідомити, чи являється об'єкт, з яким вони працюють, базовою таблицею або представленням, тобто передусім це засіб захисту для приховання конфіденційних частин таблиці або таких частин таблиці, що не відносяться до потреб цього користувача. Представлення в режимі «тільки для читання» дозволяють отримувати і формувати дані раціональніше. Вони створюють цілий арсенал складних запитів, які можна виконати і повторити знову, зберігаючи отриману інформацію. Результати цих запитів можуть потім використовуватися в інших запитах, що дозволить уникнути складних предикатів і понизити вірогідність помилкових дій.

1.3 Переваги і недоліки представлень

Механізм представлення – потужний засіб СУБД, що дозволяє приховати реальну структуру бази даних від деяких користувачів шляхом визначення представлень. Будь-яка реалізація представлення повинна гарантувати, що стан відношення, що представляється, точно відповідає стану даних, на яких визначено це представлення. Звичайне обчислення представлення робиться кожного разу при його використанні. Коли представлення створюється, інформація про нього записується в каталог

БД під власним ім'ям. Будь-які зміни в даних адекватно відобразяться в представленні - в цьому його відмінність від дуже схожого на нього запиту до БД. В той же час запит є як би <миттєвою фотографією> даних і при зміні останніх запит до БД необхідно повторити. Наявність представлень у БД потрібна для забезпечення логічної незалежності даних. Якщо система забезпечує фізичну незалежність даних, то зміни у фізичній структурі БД не впливають на роботу призначених для користувача програм. Логічна незалежність має на увазі той факт, що при зміні логічної структури даних вплив на призначені для користувача програми також не виявляється, а значить, система повинна уміти вирішувати проблеми, пов'язані з ростом і реструктуризацією БД. Очевидно, що зі збільшенням кількості даних, що зберігаються у БД, виникає необхідність її розширення за рахунок додавання нових атрибутів або стосунків – це називається ростом БД. Реструктуризація даних має на увазі збереження тієї ж самої інформації, але змінюється її розташування, наприклад, за рахунок перегруповання атрибутів в стосунках. Припустимо, деяке відношення через які-небудь причини необхідно розділити на два. З'єднання отриманих стосунків в представленні відтворює початкове відношення, а у користувача складається враження, що ніякої реструктуризації не робилася. Окрім вирішення проблеми реструктуризації представлення можна застосовувати для перегляду одних і тих же даних різними користувачами і в різних варіантах. За допомогою представлень користувач має можливість обмежити об'єм даних для зручності роботи. Нарешті, механізм уявлень дозволяє приховати службові дані, не цікаві користувачам.

У разі виконання СУБД на персональному комп'ютері, що працює автономно, використання представлень зазвичай має на меті лише спрощення структури запитів до бази даних. Проте у разі розрахованої на багато користувачів мережевої СУБД представлення грають ключову роль у визначенні структури бази даних і організації захисту інформації. Розглянемо основні **переваги** застосування представлень в подібному середовищі.

1. Незалежність від даних.

За допомогою представлень можна створити погоджену, незмінну картину структури бази даних, яка залишатиметься стабільною навіть у разі зміни формату початкових таблиць (наприклад, додавання або видалення стовпців, зміни зв'язків, розподілу таблиць, їх реструктуризації або перейменування). Якщо в таблицю додаються або з неї видаляються не використовувані в представленні стовпці, то змінювати визначення цього представлення буде не потрібно. Якщо структура початкової таблиці перевпорядковується або таблиця розділяється, можна створити представлення, що дозволяє працювати з віртуальною таблицею колишнього формату. У разі розподілу початкової таблиці, колишній формат може бути віртуально відтворений за допомогою представлення, побудованого на основі з'єднання знову створених таблиць, - звичайно, якщо це виявиться можливим. Останню умову можна забезпечити за допомогою приміщення в усі знову створені таблиці первинного ключа колишньої таблиці.

2. Актуальність.

Зміни даних у будь-якій з таблиць бази даних, вказаних у визначальному запиті, негайно відображається на вмісті представлення.

3. Підвищення захищеності даних.

Права доступу до даних можуть бути надані виключно через обмежений набір представлень, що містять тільки ту підмножину даних, яка потрібна користувачеві. Подібний підхід дозволяє істотно посилити контроль за доступом окремих категорій користувачів до інформації у базі даних.

4. Зниження складності.

Представлення дозволяють спростити структуру запитів за рахунок об'єднання даних з декількох таблиць в єдину віртуальну таблицю. В результаті багатотабличні запити зводяться до простих запитів до одного представлення.

5. Додаткові зручності.

Створення представлень може забезпечувати користувачів додатковими зручностями – наприклад, можливістю роботи тільки з дійсно потрібною частиною даних. В результаті можна добитися максимального спрощення тієї моделі даних, яка знадобиться кожному кінцевому користувачеві.

6. Можливість налаштування.

Представлення є зручним засобом налаштування індивідуального образу бази даних. В результаті одні і ті ж таблиці можуть бути пред'явлені користувачам в абсолютно різному вигляді.

7. Забезпечення цілісності даних.

Якщо в операторові CREATE VIEW буде вказана фраза WITH CHECK OPTION, то СУБД стане здійснювати контроль за тим, щоб в початкові таблиці бази даних не був введений жоден з рядків, що не задовольняють пропозиції WHERE у визначальному запиті. Цей механізм гарантує цілісність даних в представленні.

Практика обмеження доступу деяких користувачів до даних за допомогою створення спеціалізованих представлень, безумовно, має значні переваги перед наданням їм прямого доступу до таблиць бази даних.

Проте використання представлень в середовищі SQL не позбавлене **недоліків**.

1. Обмежені можливості оновлення.

В деяких випадках представлення не дозволяють вносити зміни в дані, що містяться в них.

2. Структурні обмеження.

Структура представлення встановлюється у момент його створення. Якщо визначальний запит представлений у формі SELECT * FROM , то символ * посилається на усі стовпці, існуючі в вихідній таблиці на момент створення представлення. Якщо згодом в вихідну таблицю бази даних

додадуться нові стовпці, то вони не з'являться в цьому представленні до тих пір, поки це представлення не буде видалено і знову створене.

3. Зниження продуктивності.

Використання представлень пов'язане з певним зниженням продуктивності. У одних випадках вплив цього чинника абсолютно тривиальний, тоді як в інших воно може послужити джерелом істотних проблем. Наприклад, представлення, визначене за допомогою складного багатотабличного запиту, може зажадати значних витрат часу на обробку, оскільки при його формуванні потрібно буде виконувати з'єднання таблиць всякий раз, коли знадобиться доступ до даного представлення. Виконання формування представлень пов'язане з використанням додаткових обчислювальних ресурсів.

2 ВИКОНАННЯ РОБОТИ

При виконанні лабораторної роботи передбачається використання застосунку SQL Server Management Studio, який входить до складу інтегрованої платформи Microsoft SQL Server. При цьому передбачається використання СУБД Microsoft SQL Server версії 2008 або вищої. У зв'язку із цим елементи інтерфейсу можуть мати відмінності порівняно із тими, що наведено далі у вигляді рисунків (особливо у випадку використання версії, локалізованої для використання певної національної мови). Ці відмінності не є принциповими та не впливають на виконання роботи та отримані результати.

Для початку виконання треба підключити базу даних (delivery або dlvr), з якою працювали при виконанні лабораторної роботи за темою «Створення та використання програмних об'єктів бази даних засобами СУБД Microsoft SQL Server».

Увага! Результати використання програмних об'єктів, що розглядаються далі, повинні бути результативними (тобто в результаті виконання запиту повинні бути виведені один або кілька записів). Відсутність результату запиту є ознакою помилок під час побудови запиту, невідповідності запиту наявним даним тощо. Такий запит потребує аналізу та перевірки. Також передбачається, що результати запитів ґрунтуються на тих даних, які були введені до бази даних при виконанні попередньої лабораторної роботи.

2.1 Побудова та використання представлення для перегляду даних

Як відомо, при нормалізації відношень виконується декомпозиція відношення, яке потребує нормалізації. При цьому замість одного вихідного відношення створюється два або більше відношень, які зв'язуються між собою за допомогою первинних та зовнішніх ключів. Нормалізовані відношення не мають таких недоліків, як, наприклад,

аномалії оновлення, але їх зміст для кінцевого користувача не є зручним для сприйняття. Наприклад, відношення (тобто) таблиця «Contracts» містить посилання на таблицю «Suppliers» у вигляді коду постачальника. Уся інша інформація про постачальників зберігається у відповідних окремих таблицях. Припустимо, що у користувача цієї бази даних виникла потреба при перегляді списку договорів бачити більш докладну інформацію щодо постачальників, а саме назву постачальника, прізвище, ім'я та по батькові для фізичних осіб або податковий номер для юридичних осіб. Для вирішення цієї задачі можна застосувати представлення.

Для створення представлення слід виконати таку послідовність дій.

1. Клацнути правою кнопкою миші по пункту Views і в меню вибрати пункт New View...

2. У списку таблиць вибрати таблиці «Contracts», «Suppliers», «IndividualEntrepreneurs», «LegalEntities». Список таблиць закрити. В результаті з'явиться графічне зображення таблиць, що використовуються як джерело даних для представлення, та зв'язків між ними (рисунок 2.1)

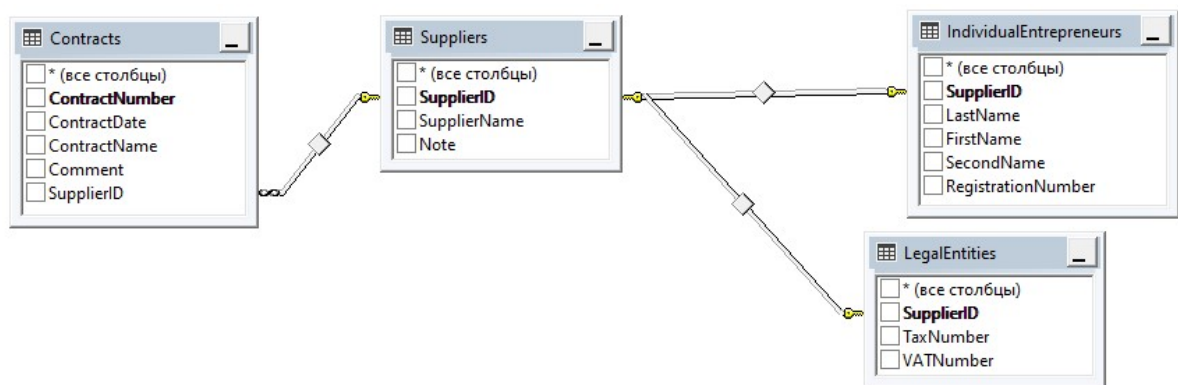


Рисунок 2.1

3. Клацнути правою кнопкою миші по зв'язку між таблицями «Suppliers» та «IndividualEntrepreneurs» та вибрати пункт Select All Rows from Suppliers. Клацнути правою кнопкою миші по зв'язку між таблицями «Suppliers» та «LegalEntities» та вибрати пункт Select All Rows from

Suppliers. В результаті зв'язки набудуть вигляду, наведеного на рисунку 2.2.

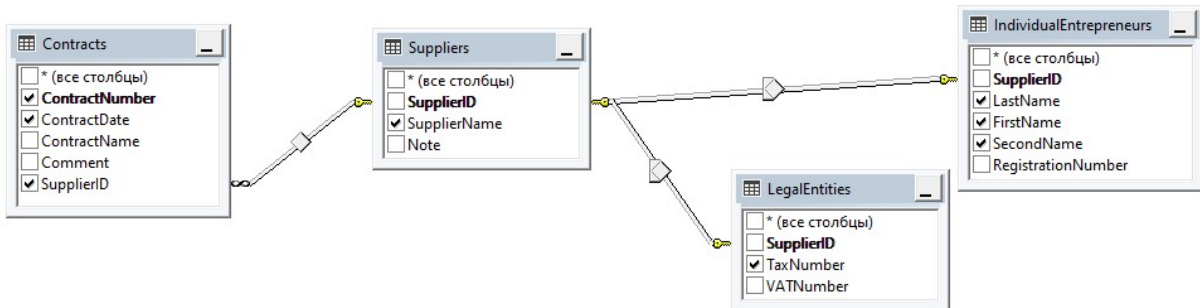


Рисунок 2.2

4. Вибрати поля таблиць, які включаються до результату запиту, поставивши позначки для відповідних полів (рисунк 2.2). У результаті текст запиту представлення матиме вигляд, наведений рисунку 2.3. Натиснувши на панелі інструментів кнопку Execute SQL, можна отримати результат запиту. Цей результат має певний недолік – дані постачальників як суб'єктів господарювання (тобто, як юридичних та фізичних осіб) перебувають у різних полях (рисунк 2.4). Цей недолік можна виправити, змінивши текст запиту (рисунк 2.5).

```
SELECT  dbo.Contracts.ContractNumber, dbo.Contracts.ContractDate, dbo.Contracts.SupplierID, dbo.Suppliers.SupplierName, dbo.LegalEntities.TaxNumber,
FROM    dbo.Contracts INNER JOIN
        dbo.Suppliers ON dbo.Contracts.SupplierID = dbo.Suppliers.SupplierID LEFT OUTER JOIN
        dbo.LegalEntities ON dbo.Suppliers.SupplierID = dbo.LegalEntities.SupplierID LEFT OUTER JOIN
        dbo.IndividualEntrepreneurs ON dbo.Suppliers.SupplierID = dbo.IndividualEntrepreneurs.SupplierID
```

Рисунок 2.3

	ContractNum...	ContractDate	SupplierID	SupplierName	TaxNumber	LastName	FirstName	SecondName
►	1	1999-09-01 00:0...	1	ПП Іваненко І.І	NULL	Іваненко	Ілля	Іванович
	2	1999-09-10 00:0...	1	ПП Іваненко І.І	NULL	Іваненко	Ілля	Іванович
	3	1999-09-10 00:0...	3	ПП Петренко ...	NULL	Петренко	Павло	Петрович
	4	1999-09-23 00:0...	3	ПП Петренко ...	NULL	Петренко	Павло	Петрович
	5	1999-09-24 00:0...	2	ТОВ «Інтерфрут»	00123987	NULL	NULL	NULL
	6	1999-10-01 00:0...	1	ПП Іваненко І.І	NULL	Іваненко	Ілля	Іванович
	7	1999-10-02 00:0...	2	ТОВ «Інтерфрут»	00123987	NULL	NULL	NULL

Рисунок 2.4


```

SELECT  dbo.Contracts.ContractNumber, dbo.Contracts.ContractDate, dbo.Contracts.SupplierID, dbo.Suppliers.SupplierName,
        ISNULL(dbo.LegalEntities.TaxNumber + SPACE(30), RTRIM(dbo.IndividualEntrepreneurs.LastName)
        + ' ' + RTRIM(dbo.IndividualEntrepreneurs.FirstName) + ' ' + RTRIM(dbo.IndividualEntrepreneurs.SecondName)) AS Supplier
FROM      dbo.Contracts INNER JOIN
        dbo.Suppliers ON dbo.Contracts.SupplierID = dbo.Suppliers.SupplierID LEFT OUTER JOIN
        dbo.LegalEntities ON dbo.Suppliers.SupplierID = dbo.LegalEntities.SupplierID LEFT OUTER JOIN
        dbo.IndividualEntrepreneurs ON dbo.Suppliers.SupplierID = dbo.IndividualEntrepreneurs.SupplierID

```

Рисунок 2.5

5. Змінений результат виконання запиту наведений на рисунку 2.6. Можна вважати, що цей результат вже задовольняє умовам, які висувалися для перегляду списку договорів. Зверніть увагу на появу у тексті запиту функцій ISNULL, SPACE, RTRIM.

	ContractNum...	ContractDate	SupplierID	SupplierName	Supplier
►	1	1999-09-01 00:0...	1	ПП Іваненко І.І	Іваненко Ілля Іванович
	2	1999-09-10 00:0...	1	ПП Іваненко І.І	Іваненко Ілля Іванович
	3	1999-09-10 00:0...	3	ПП Петренко П.П	Петренко Павло Петрович
	4	1999-09-23 00:0...	3	ПП Петренко П.П	Петренко Павло Петрович
	5	1999-09-24 00:0...	2	ТОВ «Інтерфрут»	00123987
	6	1999-10-01 00:0...	1	ПП Іваненко І.І	Іваненко Ілля Іванович
	7	1999-10-02 00:0...	2	ТОВ «Інтерфрут»	00123987

Рисунок 2.6

6. Зберегти представлення з ім'ям View_1.

7. Перевірити роботу представлення, для чого клацнути правою кнопкою миші на ім'я представлення і у меню вибрати команду Open View.

2.2 Оновлення даних за допомогою представлень

2.2.1 Використання параметру WITH CHECK OPTION

Припустимо, що виникла потреба роботи з договорами, які будо укладено з певним постачальником. Для отримання такого списку договорів можна застосувати представлення. Запит, за допомогою якого можна створити таке представлення, наведений на рисунку 2.7.

```

CREATE VIEW [dbo].[View_11]
AS
SELECT      ContractNumber, ContractDate, SupplierID, Comment
FROM        dbo.Contracts
WHERE       (SupplierID = 3)
WITH CHECK OPTION

```

Рисунок 2.7

За допомогою такого представлення можна оновлювати дані. Спробуємо це зробити за допомогою запиту, який наведено на рисунку 2.8. Як видно з повідомлення, що наведено на рисунку 2.9, дані у жодному запису у результаті виконання цього запиту змінено не буде.

```

USE delivery
UPDATE View_11 SET SupplierID=2 WHERE ContractNumber=1

```

Рисунок 2.8

Messages

(0 row(s) affected)

Рисунок 2.9

Наступний приклад оновлення даних за допомогою цього представлення наведено на рисунку 2.10. При спробі виконання цього запиту буде виведено повідомлення, яке наведено на рисунку 2.11. Поява цього повідомлення обумовлена саме застосуванням параметру WITH CHECK OPTION при визначенні представлення.

```

USE delivery
UPDATE View_11 SET SupplierID=2 WHERE ContractNumber=3

```

Рисунок 2.10

Messages

Msg 550, Level 16, State 1, Line 2
The attempted insert or update failed because the target view either specifies WITH CHECK OPTION or spans a view that specifies WITH CHECK OPTION and one or more rows resulting from the operation did not qualify under the CHECK OPTION constraint.
The statement has been terminated.

Рисунок 2.11

Для того, щоб з'ясувати, що зміниться у цьому представленні, якщо прибрати параметр WITH CHECK OPTION, змінимо попереднє представлення. Запит, за допомогою якого можна створити таке представлення, наведений на рисунку 2.12.

```
CREATE VIEW [dbo].[View_2]
AS
SELECT      ContractNumber, ContractDate, SupplierID, Comment
FROM        dbo.Contracts
WHERE       (SupplierID = 3)
```

Рисунок 2.12

Перевіримо роботу цього представлення, застосувавши такі ж запити, як і для попереднього представлення. Результат першого запиту (рисунок 2.13) буде таким, як і у попередньому випадку (рисунок 2.14).

```
USE delivery
UPDATE View_2 SET SupplierID=2 WHERE ContractNumber=1
```

Рисунок 2.13



Messages

(0 row(s) affected)

Рисунок 2.14

А у другому випадку (рисунок 2.15) оновлення даних, на відміну від попередньої спроби, буде результативним (рисунок 2.16). Також, перевібивши дані у таблиці «Contracts», можна побачити, що код постачальника у договорі 3 змінено з 3 на 2. Це стало можливим саме завдяки відсутності параметра WITH CHECK OPTION.

```
USE delivery
UPDATE View_2 SET SupplierID=2 WHERE ContractNumber=3
```

Рисунок 2.15

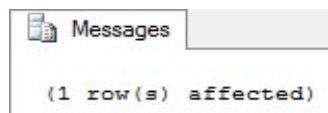


Рисунок 2.16

2.2.2 Використання триггеру INSTEAD OF

У базі даних, що розглядається, дані про постачальників зберігаються у трьох окремих таблицях. Тому, наприклад, при введенні даних про нового постачальника, який є фізичною особою, відповідні операції треба буде виконувати стосовно двох таблиць. Це може бути не дуже зручним, тому може виникнути задача розробки таких засобів роботи з даними, застосування яких може зменшити обсяг роботи кінцевого користувача.

Для вирішення цієї задачі можна створити представлення, що оновлюється, для таблиць «Suppliers» та «IndividualEntrepreneurs». При додаванні в представлення нового запису про постачальника, що є фізичною особою (ФОП – фізична особа-підприємець або ПП – приватний підприємець), додані дані повинні бути записані і до таблиці «Suppliers», і до таблиці «IndividualEntrepreneurs». Таке представлення наведене на рисунку 2.17.

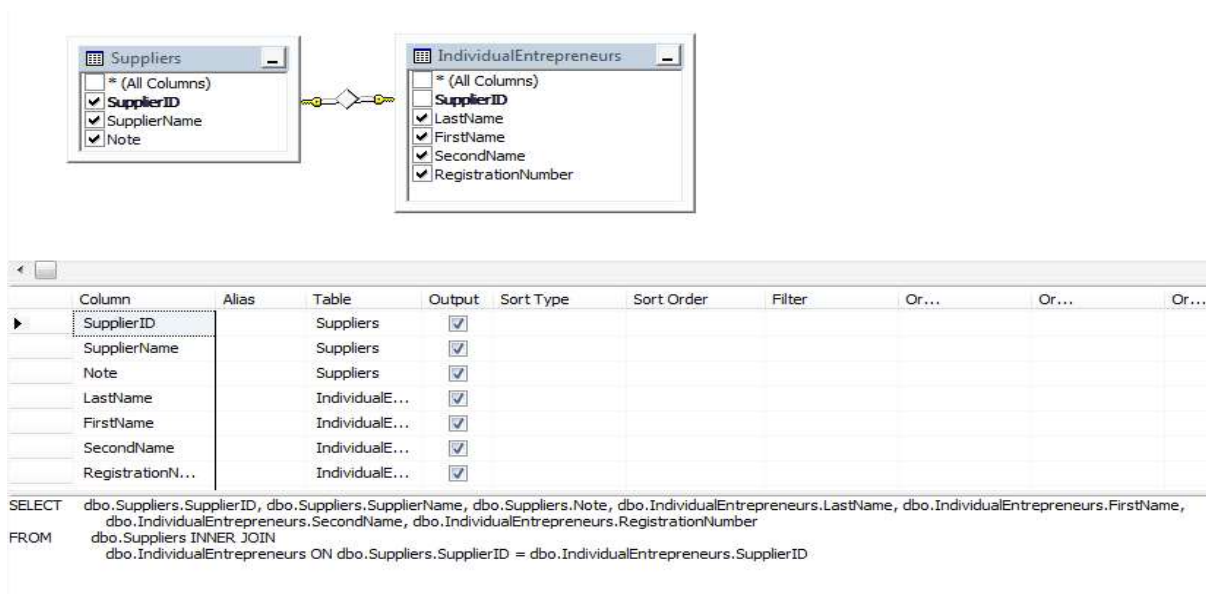


Рисунок 2.17

Стан даних у таблицях «Suppliers» та «IndividualEntrepreneurs» та дані, що сформовані за допомогою представлення, наведені на рисунку 2.18.

	SupplierID	SupplierName	Note
	1	ПП Іваненко І.І	м. Харків, вул. Пушкінська, 77 (тел.33-33-44, 12-34-56, факс 22-12-33)
	2	ТОВ «Інтерфрут»	м. Київ, пр. Перемоги, 154, к. 3
	3	ПП Петренко П.П	м. Харків, пр. Науки, 55, к. 108, тел.32-18-44
	4	ЗАТ «Транссервіс»	м. Одеса, вул. Дерибасівська, 75
	5	ПП Сидорчук М.С.	м. Полтава, вул. Свободи, 15, кв. 43
►*	NULL	NULL	NULL

	SupplierID	LastName	FirstName	SecondName	RegistrationN...
►	1	Іваненко	Ілля	Іванович	00143987
	3	Петренко	Павло	Петрович	12345678
	5	Сидорчук	Микита	Степанович ...	09876541
*	NULL	NULL	NULL	NULL	NULL

	SupplierID	SupplierName	Note	LastName	SecondName	FirstName	RegistrationN...
►	1	ПП Іваненко І.І	м. Харків, вул. Пушкі...	Іваненко	Іванович	Ілля	00143987
	3	ПП Петренко ...	м. Харків, пр. Науки, ...	Петренко	Петрович	Павло	12345678
	5	ПП Сидорчук ...	м. Полтава, вул. Сво...	Сидорчук	Степанович ...	Микита	09876541
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 2.18

Для цього представлення створимо тригер INSTEAD OF. Текст відповідного запиту наведений на рисунку 2.19.

```

CREATE TRIGGER [dbo].[Insert_IE] ON [dbo].[View_IE]
    INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @new_SupplierID int, @new_SupplierName char(20), @new_SuppliersNote char(120),
            @new_LastName char(20), @new_SecondName char(20), @new_FirstName char(20),
            @new_RegistrationNumber char(20)
    SELECT @new_SupplierID=SupplierID, @new_SupplierName=SupplierName, @new_SuppliersNote=Note,
           @new_LastName=LastName, @new_SecondName=SecondName, @new_FirstName=FirstName,
           @new_RegistrationNumber=RegistrationNumber FROM inserted
    IF NOT @new_SupplierID IS NULL AND NOT @new_LastName IS NULL
    IF (NOT EXISTS (SELECT * FROM Suppliers WHERE SupplierID=@new_SupplierID))
        INSERT INTO Suppliers VALUES (@new_SupplierID, @new_SupplierName, @new_SuppliersNote)
        INSERT INTO IndividualEntrepreneurs VALUES (@new_SupplierID, @new_LastName, @new_SecondName,
                                                    @new_FirstName, @new_RegistrationNumber)
END
GO

```

Рисунок 2.19

Для перевірки роботи представлення та тригера, введемо тестові дані про нового постачальника, про якого відразу відомо, що він є фізичною особою. Приклад введення таких даних у представлення та їх поява у таблицях, які є джерелом даних для представлення, наведений на рисунку 2.20.

	SupplierID	SupplierName	Note	LastName	SecondName	FirstName	RegistrationN...
	1	ПП Іваненко І.І	м. Харків, вул. ...	Іваненко	Іванович	Ілля	00143987
	3	ПП Петренко ...	м. Харків, пр. ...	Петренко	Петрович	Павло	12345678
▶	5	ПП Сидорчук ...	м. Полтава, ву...	Сидорчук	Степанович ...	Микита	09876541
!	7	777	777	777	777	777	777
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	SupplierID	SupplierName	Note
▶	1	ПП Іваненко І.І	м. Харків, вул. ...
	2	ТОВ «Інтерфрут»	м. Київ, пр. Пе...
	3	ПП Петренко ...	м. Харків, пр. ...
	4	ЗАТ «Транссер...	м. Одеса, вул. ...
	5	ПП Сидорчук ...	м. Полтава, ву...
	7	777	777 ...
*	NULL	NULL	NULL

	SupplierID	LastName	FirstName	SecondName	RegistrationN...
▶	1	Іваненко	Ілля	Іванович	00143987
	3	Петренко	Павло	Петрович	12345678
	5	Сидорчук	Микита	Степанович ...	09876541
	7	777	777	777	777
*	NULL	NULL	NULL	NULL	NULL

Рисунок 2.27

Наведені вище приклади представлень є досить простими. Більш детальну інформацію щодо побудови та використання представлень засобами мови Transact-SQL, можна, наприклад, отримати за посиланням:

<https://learn.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver16>

2.3 Збереження результатів роботи

Для збереження результатів виконання лабораторної роботи треба відключити базу даних, з якою працювали, та зберегти файли бази даних.

3 ВИМОГИ ДО ЗВІТУ

Відповіднім чином оформлений та роздрукований звіт з лабораторної роботи є документом, що підтверджує виконання студентом лабораторної роботи.

У звіті з лабораторної роботи:

- 1) стисло описати основні етапи виконання завдання;
- 2) для кожного з реалізованих представлень навести призначення, текст запиту для створення та результат використання представлення;
- 3) зробити висновки за результатами виконання лабораторної роботи.

Звіт з лабораторної роботи роздруковується на аркушах формату А4, він повинен мати відповідний титульний аркуш. Роздрукований звіт здається студентом викладачу у файлі.

Звіт має бути оформлений за такими вимогами:

- параметри сторінки: лівий відступ – 3 см; правий – 1,5 см; верхній та нижній відступи по 2 см;
- шрифт Times New Roman, 14;
- налаштування абзацу: вирівнювання – за шириною, відступи зліва та справа – 0 см, відступ першого рядка - 1,25 см, інтервал перед та після абзацу – 0 пт, міжрядковий інтервал – одинарний; на вкладці «Положення на сторінці» відключити функцію «Заборона висячих рядків».

Усі скріншоти розміщені у звіті, є рисунками, отже повинні мати підписи та відповідну нумерацію.

Ознакою того, що студент не тільки виконав лабораторну роботу, але й здав її (тобто підтвердив наявність відповідних знань та навичок), є наявність на титульному аркуші підпису викладача та дати здачі.

4 ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Представлення в SQL. Загальна характеристика, призначення та використання.
2. Команди мови SQL для роботи з представленнями.
3. Оновлення даних в представленнях. Яким умовам повинно відповідати представлення, за допомогою якого можна оновлювати дані?
4. Параметр WITH CHECK OPTION. Призначення і використання.
5. Тригери INSTEAD OF. Загальна характеристика, призначення та використання у представленнях.
6. Переваги використання представлень.
7. Недоліки використання представлень.

СПИСОК ЛІТЕРАТУРИ

1. Програмування баз даних. Практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 152 «Метрологія та інформаційно-вимірjuвальна техніка» / М. В. Добролюбова, М. В. Філіппова, О. М. Маркіна. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 164 с.
2. Anthony DeBarros. Practical SQL. 2nd Edition. – No Starch Press, Inc., 2022. – 440 с.
3. Jeffrey A. Hoffer, V. Ramesh, Heikki Topi. Modern Database Management. Thirteenth Edition. – Pearson Education Limited, 2020. – 591 p.
4. Dušan Petković. Microsoft SQL Server 2019. A Beginner's Guide. Seventh Edition. – McGraw-Hill Education, 2020. – 865 p.
5. Mukesh Negi. Fundamentals of Database Management System: Learn essential concepts of database systems. – BPB Publications, 2019. – 175 p.
6. Edward Sciore. Database Design and Implementation: Second Edition. – Springer Nature, 2020. – 468 p.
7. Gavin Powell. Database Modeling Step by Step. – CRC Press, 2020. – 268 p.
8. Sanjiv Purba. Handbook of Data Management: 1999 Edition. – CRC Press, 2019. – 1101 p.
9. C. J. Date. Database Design and Relational Theory: Normal Forms and All That Jazz. – Apress, 2019. – 451 p.
10. Jonathan Eckstein, Bonnie R. Schultz. Introductory Relational Database Design for Business, with Microsoft Access. – John Wiley & Sons, 2018. – 328 p.
11. Alan Beaulieu. Learning SQL: Generate, Manipulate, and Retrieve Data. – O'Reilly Media, Inc., 2020. – 384 p.
12. M. Tamer Özsu, Patrick Valduriez. Principles of Distributed Database Systems. – Springer Nature, 2019. – 674 p.

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи за темою «Створення та використання
представлень (view) засобами СУБД Microsoft SQL Server»

для студентів спеціальностей

121 «Інженерія програмного забезпечення»

122 «Комп'ютерні науки»

126 «Інформаційні системи та технології»

Укладачі:

ОРЛОВСЬКИЙ Дмитро Леонідович

КОПП Андрій Михайлович

Відповідальний за випуск Годлевський М.Д.

Роботу до видання рекомендував Гамаюн І.П.

План 2023 р., поз. 495

Підп. до друку __.__.__. Гарнітура Times New Roman.

Ум. друк. арк. 1,1.

Видавничий центр НТУ «ХП»,

вул. Кирпичова, 2, м. Харків, 61002

Свідоцтво про державну реєстрацію ДК № 3478 від 21.08.2017 р.

Самостійне електронне видання