**Лабораторна робота 15.**

**Тригери DML та їх використання.**

**Мета роботи:** навчитись створювати тригери таблиць на операції додавання, зміни та видалення даних, та розібрати основні можливості щодо оцінки кількості та якості змінених даних і можливості управління транзакціями в тілі тригера.

**Задачі:**

* Розібрати поняття тригерів, тригерів DML та області їх використання
* Познайомитись із типами тригерів та навчитись створювати на модифікувати тригери таблиць
* Розібрати використання команд управління транзакціями у тілі тригера
* Використати @@ROWCOUNT, COLUMNS\_UPDATED() та UPDATE() в тілі тригера
* Розглянути роботу із таблицями INSERTED та DELETED
* Виконання індивідуального завдання згідно варіанту

1. Тригер може бути визначено, як різновид збереженої процедури, яка виконується автоматично при виникненні події на сервері бази даних. Розрізняють тригери мови DML, DDL та тригери входу. Тригери мови DML виконуються за подіями, спричинені спробою користувача змінити дані за допомогою мови обробки даних. Подіями DML є команди INSERT, UPDATE або DELETE, що застосовуються до таблиці або представлення. Тригери DDL спрацьовують у відповідь на ряд подій мови визначення даних (DDL). Ці події відповідають інструкціям T-SQL CREATE, ALTER, DROP. Тригери входу спрацьовують на подію LOGON, що виникає при встановленні сеансу користувача.

Із наведених типів тригерів, тригери DML є найбільш часто використовуваним типом тригера. Тригер DML в SQL Server – це набір інструкцій T-SQL, що запускаються сервером автоматично при виконанні є команди INSERT, UPDATE або DELETE, що застосовуються до таблиці або представлення. Кожен тригер прив'язується до конкретної таблиці, але також існує можливість створювати тригери не тільки для таблиць, але і для представлень. Також дуже важливо, що команда, з якою пов’язаний тригер та всі інструкції T-SQL, з яких складається тіло тригера, виконуються сервером, як одна транзакція.

Тригери дозволяють створити складне значення за замовчуванням; забезпечити нестандартної цілісності посилань; бути використані для виконання каскадних змін в декількох зв'язаних таблицях. Область застосування тригерів не обмежена, однак не слід застосовувати тригери для виконання дій, які можуть бути виконані за допомогою правил, обмежень цілісності, процедури або звичайного пакета команд, оскільки тригери утримують блокування до свого завершення, забороняючи звернення до ресурсу інших користувачів.

Тригери розрізняють за типом команди (INSERT, UPDATE або DELETE) та за типом поведінки (AFTER та INSTEAD OF – що виконуються після або замість команди, з якою пов’язаний тригер).

1. Далі наведемо загальний синтаксис команди створення тригера (рис.1)

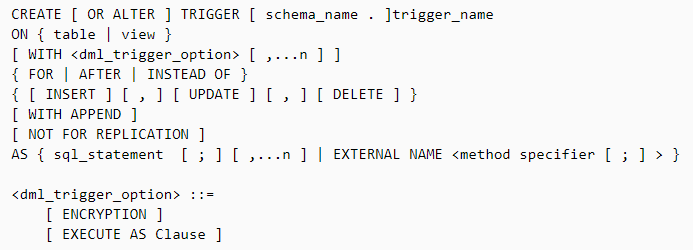


Рис.1. Синтаксис команди створення DML тригера

При створенні необхідно зазначити унікальне у межах схеми даних ім’я тригера, таблицю, з якою він пов’язаний, тип тригера, та список команд, на які він реагуватиме. Після ключового слова AS наводиться список інструкцій T-SQL, що буде виконаний при його спрацюванні. Можуть також бути вказані додаткові опції, такі як наприклад шифрування, виконання від імені іншого користувача та ін.

У тілі тригера не допускається використання команд створення або видалення БД, або модифікації структури таблиці, з якою він пов’язаний. Також не рекомендується використовувати жодні команди, що повертають дані, оскільки результат навряд чи буде оброблено. Виключення може становити етап відлагодження роботи тригера.

1. Дуже важливою особливістю є виконання тіла тригера в одній неявній транзакції разом із командою, що спричинила його спрацювання. Завдяки цьому існує можливість використання команд фіксації та відкату транзакції (commit transaction та rollback transaction). Остання є більш корисною, оскільки фіксація виконується сервером автоматично за умови, що не було виконано відкат. Наведемо приклад, що використовує відкат транзакції та забороняє будь які зміни таблиці «студенти» (рис.2).

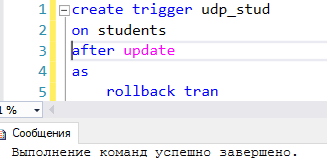


Рис.2. Заборона зміни таблиці студентів із використанням тригера

Створений тригер буде відкочувати будь яку транзакцію, що містить команду зміни таблиці студентів. Перевіримо це (рис.3)

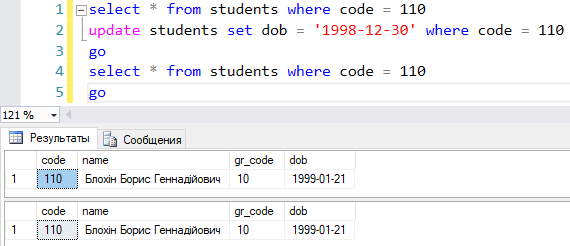


Рис.3. Перевірка роботи тригера заборони зміни даних студентів

Як бачимо, зміни не були виконані і обидва запита повертають одні і ті ж дані. Також отримаємо повідомлення про завершення транзакції у тілі тригера (рис.4). Саме через нього, команди на рис.3 і були розділені на 2 окремих пакети команд.

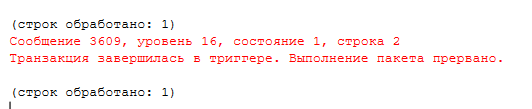


Рис.4. Повідомлення про відкат транзакції в тригері

1. Особливість тригерів SQL Server полягає у тому, що тригер буде виконано стільки разів, скільки буде виконано відповідних команд, а не скільки рядків обробляють ці команди. Покажемо це на прикладі. Для початку змінимо створений раніше тригер, щоб від виводив просте повідомлення при зміні таблиці студенти (рис.5)

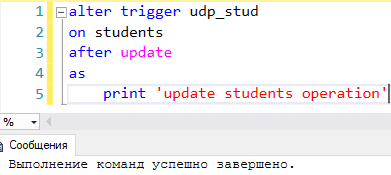


Рис.5. Тригер із простим повідомленням

Далі виконаємо 2 команди зміни таблиці студенти, кожна з яких змінює 1 рядок (рис.6)

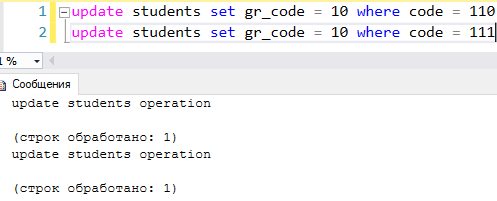


Рис.6. Багаторазова зміна таблиці

В області повідомлень бачимо, що тригер виконався 2 рази, кожного разу було оброблено 1 рядок. Тепер виконаємо вищенаведені зміни однією командою (рис.7)

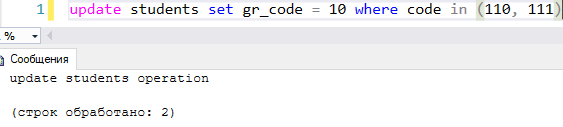


Рис.7. Зміна декількох рядків таблиці

Тепер тіло тригера відпрацьовує лише один раз, незважаючи на те, що команда змінює одразу 2 рядки.

Тепер розглянемо можливість використання системної змінної @@ROWCOUNT в тілі тригера. Дана змінна повертає кількість рядків, що була оброблена останньою успішно виконаною командою. Якщо ми звернемось до @@ROWCOUNT у першому рядку тіла тригера (до виконання будь-яких інших команд DML), то у @@ROWCOUNT буде кількість рядків, що була змінена (або додана чи видалена) у табличці, з якою пов’язаний тригер. Розглянемо приклад тригера, який забороняє видаляти, більше, ніж 5 студентів однією командою (рис.8)

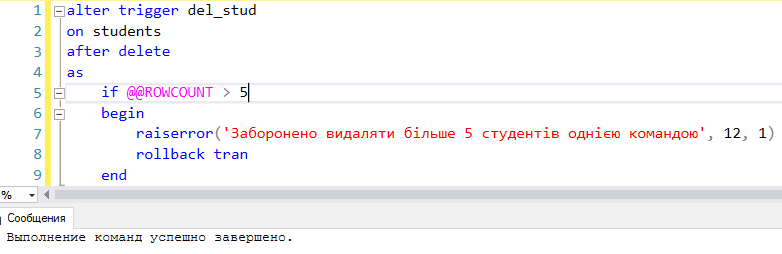


Рис.8. Використання @@ROWCOUNT в тілі тригера

Перевіримо роботу тригера, виконаємо видалення одного студента (рис.9)

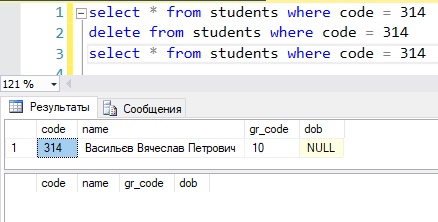


Рис.9. Видалення одного студента

Як видно, видалення пройшло успішно і друга вибірка повертається порожньою. Тепер спробуємо видалити всіх студентів однієї з студентських груп (рис.10)

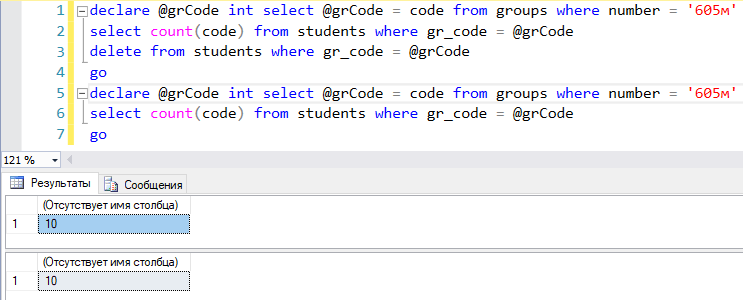


Рис.10. Видалення більше 5 рядків таблиці студенти однією командою

Команда видалення не була виконана та кількість студентів у групі 605м залишилась 10. Також ми отримали повідомлення про те, що заборонено видаляти більше 5 студентів та що транзакція була завершена у тілі тригера (рис.11)

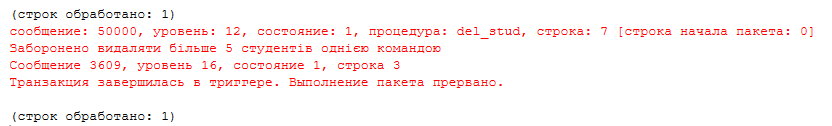


Рис.11. Повідомлення про неможливість видалення більше 5 студентів

1. Розглянемо використання у тілі тригера функцій update() та colunms\_updated(). Ці функції при використанні у тілі тригеру дозволяють визначити, чи був змінений той чи інший атрибут таблиці. Update() застосовується для одного стовбця та повертає істину, якщо від був змінений та хибність, якщо ні. Colunms\_updated() застосовується одночасно до всієї таблиці та повертає побітну маску змінених колонок. Для початку на прикладі розглянемо функцію update() та заборонимо перехід викладача з кафедри та кафедру, закривши можливість зміни поля dep\_code (рис.12)

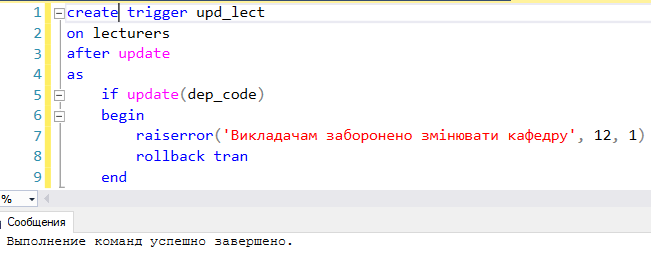


Рис.12. Тригер для заборони зміни поля dep\_code таблиці lecturers

Тепер перевіримо роботу тригера, змінивши для одного викладача поле «name», а для іншого «dep\_code» (рис.13). Як бачимо, для першого зміна пройшла успішно, а для другого отримаємо повідомлення про неможливість зміни кафедри (рис.14)

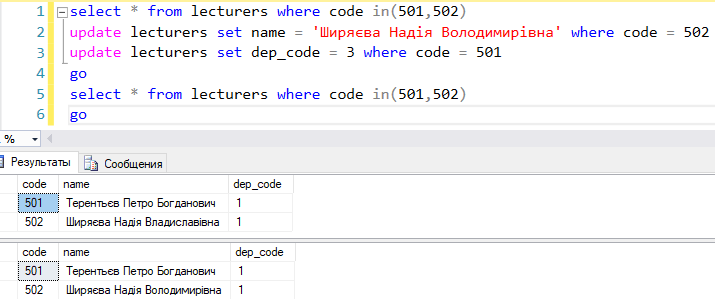


Рис.13. Зміна різних полів для 2-х викладачів

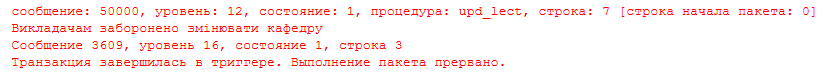


Рис.14. Повідомлення про неможливість зміни кафедри

Змінимо наш тригер для ілюстрації роботи функції columns\_updated() (рис.15)

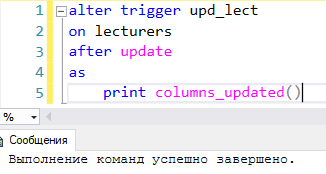


Рис.15. Використання columns\_updated()

Переглянемо, що саме повертає columns\_updated() при зміні різних комбінацій колонок (рис.16)

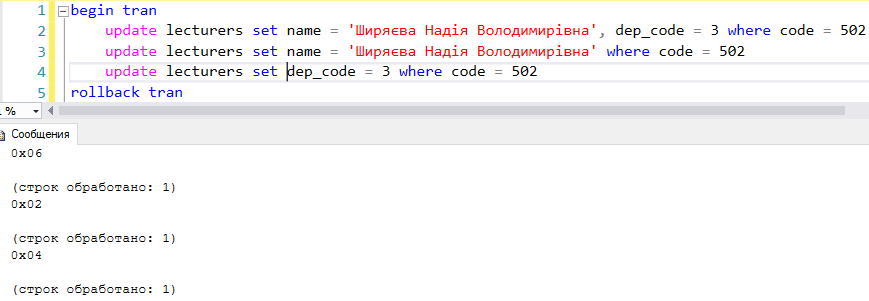


Рис.16. Зміна різних колонок у таблиці lecturers

Розберемо результат. Таблиця має поля code, name та dep\_code. При зміні name та dep\_code повертається 110 у двійковій системі, що дорівнює «6» у десятковому вираження, для name повертається 010 = «2», для dep\_code 100 = «4». Отже, якщо реалізувати приклад, що наведено на рис.12, із використанням columns\_updated(), це матиме наступний вигляд (рис.17)

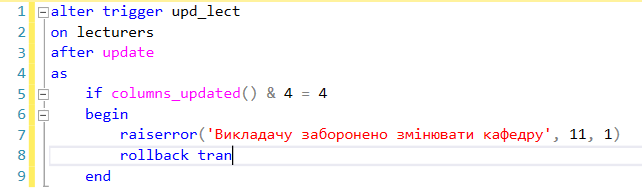


Рис.17. Використання columns\_updated() для визначення зміни поля

Отже для простих перевірок більш зручно використовувати update(), якщо ж потрібне застосування складного фільтру, columns\_updated() також може виявитись корисним.

1. Дуже потужним механізмом, що дозволяє визначити всі деталі виконаних операцій зміни даних у тілі тригеру є використання таблиць inserted та deleted. Дані таблиці за своєю структурою (набором полів) є ідентичними таблиці, до якої створений тригер. Вміст (дані) цих таблиць відрізняється в залежності від команди, яка була виконана. Для insert у табличку inserted будуть записані дані, що були додані до основної таблиці, таблиця deleted буде порожньою. Для операції delete навпаки, inserted буде порожньою, а в deleted будуть рядки, що видаляються. Для операції update в deleted будуть старі значення, а в inserted – нові.

Реалізуємо тригер, що буде вести журнал зміни таблиці students – фіксуватиме тип операції, дату час, та кількість оброблених рядків. Почнемо зі створення таблиці-журналу (рис.18)

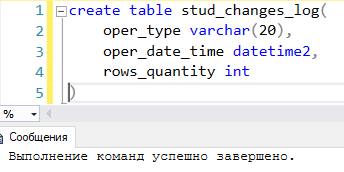


Рис.18. Створення таблиці для журналу реєстрації змін

Далі створимо сам тригер, що буде вести даний журнал (рис.19). Також видалимо всі попередньо створені тригери для таблиці students.

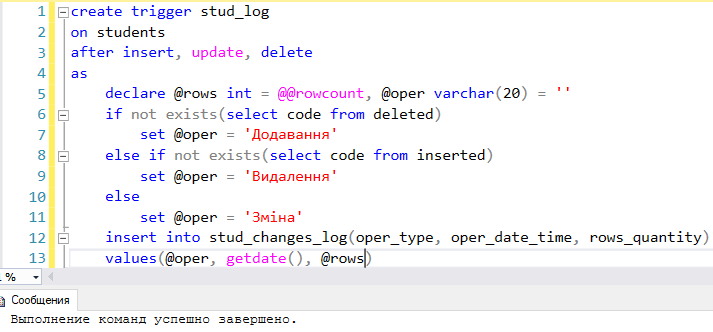


Рис.19. Тригер реєстрації змін у табличці

І нарешті перевіримо його роботу наступним скриптом (рис.20)

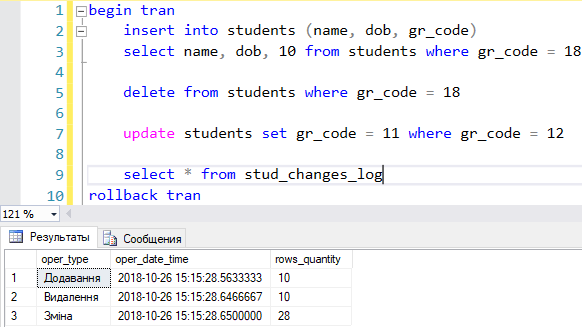


Рис.20. Перевірка роботи тригеру реєстрації змін

1. Далі розглянемо реалізацією наближеного до реальності прикладу декількома способами. На даний момент у табличці «groups» є поле кількість студентів «students\_count», яке ми заповнювали даними із використанням курсорів. Але зараз воно статичне, тобто не змінюється при додаванні нових студентів, видаленні існуючих, або при переводі студента із групи до групи.

Виправимо дану ситуацію та реалізуємо 3 тригери до таблички студенти (по одному на insert, update та delete) що будуть підтримувати поле «students\_count» таблички «groups» в актуальному стані. Задачу будемо вирішувати у 2 етапи. На першому етапі реалізуємо код тригерів по спрощеній схемі для ситуації додавання, зміни та видалення даних по одному рядку. На другому етапі переробимо тригер так, щоб вони обробляли можливість додавання, зміни та видалення декількох рядків одночасно, при чому із різних студентських груп.

На рис.21 наведемо тригер, що обробляє додавання одного студента. Якщо додається більше одного рядку, відкочуємо транзакцію.

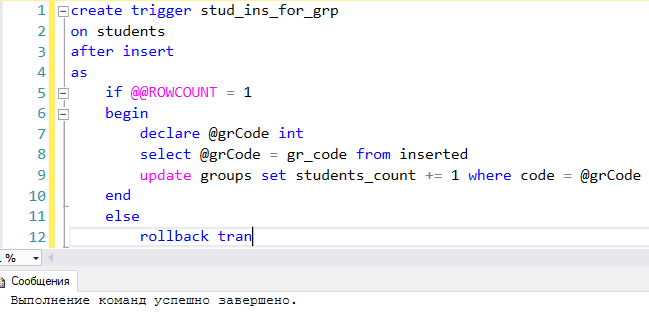


Рис.21. Обробка додавання одного рядку

На рис.22 наведемо тригер, що обробляє видалення одного студента. Тригер дуже схожий на попередній, тому реалізуємо його без використання змінних за допомогою вкладеного запиту.

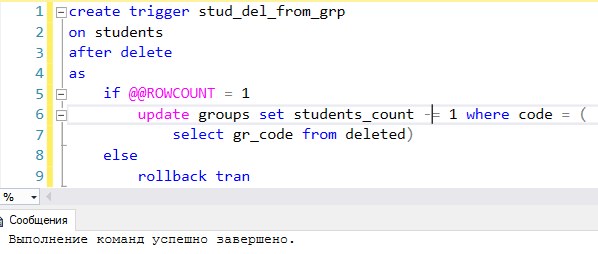


Рис.22. Обробка видалення одного рядку

На рис.23 наведемо тригер, що обробляє зміну одного студента та поєднує у собі дві вищенаведені операції. Виконуємо тіло тригеру тільки при умові зміни поля «код групи»

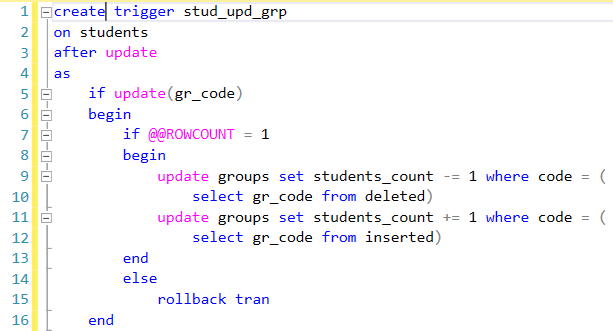


Рис.23. Обробка зміни одного рядку

У створених тригерів є суттєвий недолік – вони відкочують всі транзакції, що намагаються змінити (додати, видалити) більше одного рядку. Виправимо цю ситуацію і перепишемо їх таким чином, щоб вони працювали для будь-якої кількості рядків. Як і в попередньому випадку, реалізуємо тригери додавання та видалення даних із використанням різних підходів (рис.24-26).

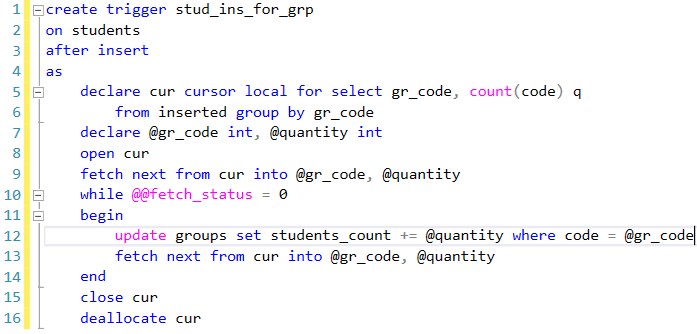


Рис.24. Обробка додавання до таблиці студентів

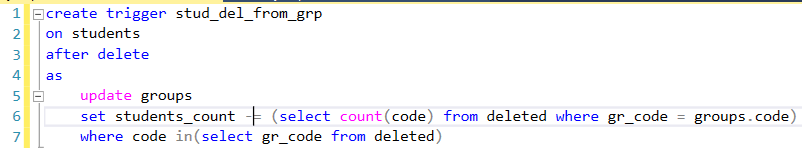


Рис.25. Обробка видалення із таблиці студентів

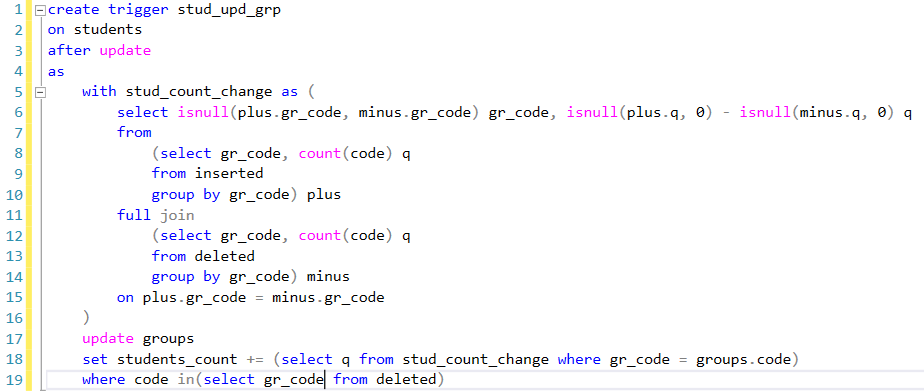


Рис.26. Обробка зміни таблиці студентів

1. Кінець роботи

**Завдання для індивідуального виконання.**

В своїй БД для числового поля, доданого згідно свого варіанту у індивідуальному завданні лабораторної роботи 10 організувати підтримку його значення в актуальному стані за допомогою реалізації тригерів до відповідної таблиці.

Для отримання максимальної кількості балів тригер має забезпечувати обробку додавання, зміни або видалення більше одного рядку в одній команді модифікації даних. Спосіб реалізації тригера не обмежується.