

*Гончар В.М асистент
кафедри інформаційних технологій*

ЗАСТОСУВАННЯ ТРАНЗАКЦІЙ НА ПРИКЛАДІ ПОКУПОК У МЕРЕЖІ ІНТЕРНЕТ

Донецький національний університет імені Василя Стуса, м. Вінниця

Під час роботи багатокористувацької бази даних необхідно забезпечити такі умови, щоб під час виконання паралельних завдань, дії одних користувачів не могли заподіяти несподіваний вплив на виконання запитів інших користувачів. Іноді головною метою є те, щоб в умовах паралельних дій користувач гарантовано отримав такий самий результат, який він мав би отримати у випадку коли він був би єдиним користувачем бази даних. В інших випадках мається на увазі, що дії різних користувачів будуть мати вплив один на одного, але при цьому їх можна буде спрогнозувати, тобто вони будуть очікуваними.[1]

На жаль, не існує стратегії організації паралельної обробки даних, що ідеально підходить для всіх випадків, тому що всі способи припускають певний компроміс між зручністю роботи і забезпеченням несуперечності бази даних.

Важлива складова частина вирішення зазначеного завдання це використання механізму транзакцій.

Транзакція - це послідовність дій у базі даних, в якій або всі дії виконуються успішно, або не виконується жодна з них. Транзакція є атомарною, вона виконується як єдине ціле.

Транзакцію можна розглядати як перетворення одного логічно узгодженого стану бази даних в інший, причому в проміжних точках під час виконання транзакції база даних може перебувати в неузгодженому стані.

Як приклад можна навести переказ грошей на оплату покупки, при якій гроші мають списатися у покупця, а потім зарахуватися на рахунок продавця, у тому разі якщо вся інформація знаходиться в спільній базі даних. Якщо відбулося списання з карти клієнта, але при цьому вона не була зарахована на рахунок продавця(як приклад можна навести збій серверу), в такому разі виникає внутрішнє протиріччя у базі даних, і гроші просто зникають. Тому для таких послідовностей операцій необхідно використовувати транзакції. Як це зробити ми поговоримо нижче.

Для встановлення рамок транзакції прикладна програма повинна дати команди для початку транзакції, збереження транзакції, скасування транзакції. Транзакція починається з оператора BEGIN TRANSACTION (в MS SQL Server допустимо скорочення BEGIN TRAN, а деякі СУБД використовують інструкцію START TRANSACTION або SET TRANSACTION).[2]

Закінчується транзакція виконанням оператора COMMIT (в запису без скорочень для MS SQL Server - COMMIT TRANSACTION або COMMIT TRAN) при успішному її виконанні. Оператор COMMIT встановлює точку фіксації, яка позначає закінчення логічної одиниці роботи.

Або оператора ROLLBACK (ROLLBACK TRANSACTION, ROLLBACK TRAN) у разі скасування цієї транзакції, та повернення бази даних до того стану, що був до початку транзакції.

Розглянемо приклад з відправкою суми з рахунку покупця на рахунок продавця. Нехай для простоти сума буде фіксована – 100грн, Дані про рахунки знаходяться в таблиці Accounts, у продавця і покупця унікальні в рамках таблиці імена ("Покупець!" Та "Продавець") і тільки за одним рахунком. Тоді транзакція може виглядати наступним чином:

```
BEGIN TRAN
```

```
UPDATE Accounts SET AccountSum = AccountSum-100 WHERE Name = 'Покупець';
```

```
UPDATE Accounts SET AccountSum = AccountSum+100 WHERE Name = 'Продавець';
```

```
COMMIT TRAN[3]
```

Якщо в процесі виконання подібної транзакції до виконання оператора COMMIT відбудеться збій електроживлення, то після процедури відновлення база даних буде повернута в стан, що був перед початком транзакції.

Можливо, знадобиться скасувати транзакцію і в інших випадках. Для цього, зокрема, може використовуватися умовний оператор IF, який визначає умову успішного завершення транзакції:

```
BEGIN TRANSACTION
```

```
IF (<умова завершення: ")
```

```
COMMIT TRANSACTION
```

```
ELSE
```

```
ROLLBACK TRANSACTION
```

Окрім явних транзакцій, початок і кінець яких чітко зазначений у коді, існують ще неявні транзакції. Наприклад, СУБД MS SQL Server за замовчуванням обробляє як неявні транзакції такі оператори SQL, як INSERT, DELETE, UPDATE. Зокрема, якщо оператор UPDATE буде змінювати кілька рядків таблиці і хоча б в один рядок ввести зміну не вийде, то вся операція буде скасована, і таблиця повернеться в стан на початок операції.

Транзакції мають таких чотири властивості:[4]

- атомарність- транзакція є неподільною, виконуються або всі дії, або нічого;

- узгодженість- транзакція переводить один узгоджений стан бази даних в інший, без дотримання обов'язкової підтримки узгодженості в проміжних точках транзакції;

- ізоляція - навіть якщо запущено кілька конкуруючих між собою транзакцій, будь-яке оновлення, виконане однією з транзакцій, буде приховано від інших, поки зміни у транзакції не будуть зафіксовані;

• довговічність - коли транзакція виконана, її поновлення зберігаються, навіть якщо в наступний момент станеться збій системи.

Коли одночасно відбувається виконання декількох транзакцій, що використовують одні ті ж об'єкти бази даних, ці транзакції називаються паралельними або конкуруючими. Якщо не зробити додаткових дій, спрямованих на забезпечення перерахованих вище властивостей, поява конкуруючих транзакцій може призвести до таких проблем, як:

- проблема 'брудного читання' - може виникнути при читанні транзакцією запису, який змінений, але ще не збережений в базі даних;
- проблема неповторюваного читання - при повторному читанні даних, які змінила інша транзакція, теперішня транзакція виявляє модифікації або видалення які були виконані попередньою транзакцією подібна зміна може порушити логіку роботи транзакції;
- проблема читання фантомів з'являється, коли при повторному читанні даних транзакція виявляє нові рядки, вставлені або змінені іншою транзакцією, яка була завершена після попереднього читання цього набору даних.[5]

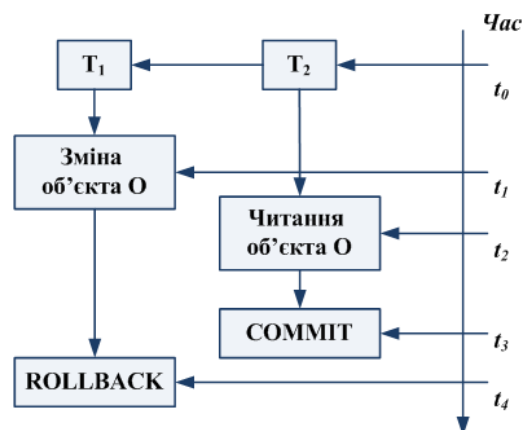


Рис 1 – Ілюстрація до феномену «брудного» читання

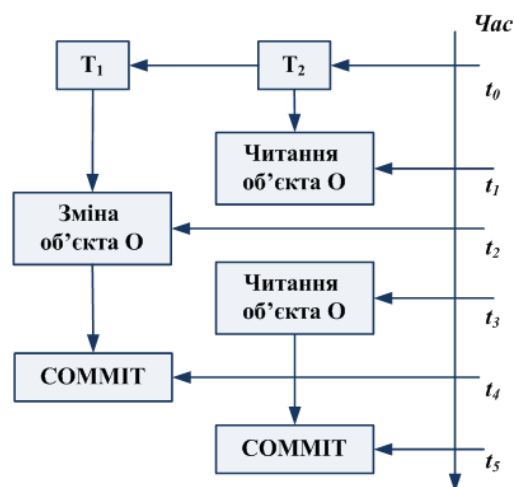


Рис 2 – Ілюстрація до феномену неповторюваного читання

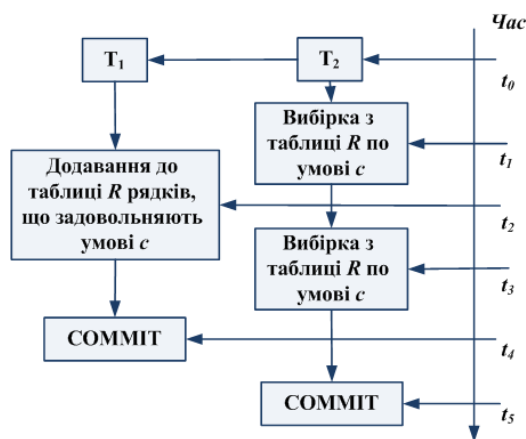


Рис 3 – Ілюстрація до феномену фантомів

Список літератури

1. Транзакції, URL: <https://stud.com.ua/93815/informatika/tranzaktsiyi>
2. SQL docs, URL: <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/begin-transaction-transact-sql?view=sql-server-ver15>
3. Transactions in sql server for beginners, URL: <https://www.sqlshack.com/transactions-in-sql-server-for-beginners/>
4. Controlling transactions, URL: <https://www.w3resource.com/sql/controlling-transactions.php>
5. Управління транзакціями SQL, URL: https://elearning.sumdu.edu.ua/free_content/lectured:a8104441b8e00905159c1ff04257b014dd456247/20151030212503/162628/index.html

УДК 004.4(043.2)

Горохова О. Г., студентка 2 курсу
спеціальності 122 «Комп'ютерні науки»
Федоров Є. Є., д.т.н., професор, професор
кафедри інформаційних технологій

ДОСЛІДЖЕННЯ ТА РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ВЕБ-СЕРВІСУ НА ОСНОВІ ЕКСПЕРТНИХ СИСТЕМ

Донецький національний університет імені Василя Стуса, м. Вінниця

Із розвитком технологій у ХХІ столітті для сучасної людини з'явилась значна кількість веб-сервісів, що полегшують прийняття певних рішень, пошук або визначення об'єктів. Так як Інтернет став головним джерелом інформації, а обсяг доступної для користувача інформації став неосяжним, до роботи з персональним комп'ютером або смартфонами долучається все більше людей без великого досвіду роботи. Це призводить до того, що користувач опиняється перед проблемою вибору та визначення об'єктів різних сфер життя. Особливо це