

***Авторы:***

***Танаев Иван Владимирович***

студент

***Швейкин Владислав Витальевич***

студент

***Завгородний Станислав Дмитриевич***

студент

***Дмитриев Егор Андреевич***

студент

ФГАОУ ВО «Самарский национальный исследовательский  
университет им. академика С.П. Королева»

г. Самара, Самарская область

## **ОБЕСПЕЧЕНИЕ СОГЛАСОВАННОСТИ ДАННЫХ В СУБД ПРИ ПОМОЩИ ТРАНЗАКЦИЙ**

***Аннотация:*** в представленной работе исследователями рассматривается вопрос обеспечения согласованности изменения данных в СУБД без снижения производительности при помощи технологий на основе транзакций.

***Ключевые слова:*** транзакция, взаимная блокировка, СУБД, базы данных, информационная безопасность, откат, точка сохранения.

### ***Введение***

На сегодняшний день большинство СУБД работают в многопользовательском режиме, то есть пользователи и процессы обрабатывают данные одновременно. Но в реляционной таблице не определён порядок хранения данных, а сами операции производятся над таблицами в целом. Для того чтобы при выполнении операций некоторым пользователем, другие пользователи при обращении к базе данных не получили произвольный набор из изменённых и ещё не изменённых, но подлежащих изменению, данных, необходимо использовать специальные тех-

нологии. Допустим несколько пользователей, независимо друг от друга и действуя примерно в одно и то же время, решили изменить значение атрибута в некоторой таблице.

Наиболее очевидным решением, обеспечивающим соответствие состояния таблицы естественному пониманию дел, является построение очереди запросов. Но такой подход к обеспечению согласованности изменения данных в СУБД привет к сильному снижению производительности. Наиболее совершенным решением являются технологии на основе транзакций.

### *Транзакции*

Транзакция – группа последовательностей операций с базой данных, которая представляет собой логически неделимую единицу работы с данными. Система гарантирует невозможность фиксации только части действий из транзакции в базе данных. Невозможна ситуация, когда результатом запроса является часть изменённых и часть неизменённых данных. Таким образом, либо транзакция выполняется целиком и успешно, переводя базу данных из одного целостного состояния в другое, независимо от параллельно идущих других транзакций, либо не выполняется совсем, и база данных возвращается в исходное состояние, которое было до начала транзакции, то есть происходит откат(отмена) транзакции.

До тех пор, пока транзакция не зафиксирована (COMMIT WORK), её можно откатить (ROLLBACK WORK). То есть отменить все изменения, сделанные операторами из транзакции, в базе данных. Следует отличать смысл фраз «SQL-операторы транзакции успешно завершены» и «транзакция зафиксирована» (COMMITTED). Успешное выполнение SQL-операторов означает, что операторы проанализированы, интерпретированы как правильные, а затем безошибочно исполнены. «Зафиксировать транзакцию» – означает сделать изменения, выполненные данной транзакцией в базе данных, постоянными. Результат действий транзакции не виден другим пользователям, пока она не зафиксирована.

Транзакция начинается автоматически с началом сеанса работы с БД и продолжается до тех пор, пока не произойдёт одно из следующих событий:

- подана команда COMMIT WORK или COMMIT (зафиксировать транзакцию);
- подана команда ROLLBACK WORK или ROLLBACK (откатить транзакцию);
- о DDL (операторы DDL всегда фиксируют транзакцию, даже если завершаются неудачно);
- завершение сеанса (последняя транзакция автоматически фиксируется);
- аварийное завершение процесса пользователя (транзакция автоматически откатывается).

После завершения транзакции сразу неявно начинается новая транзакция.

При отсутствии в конце транзакции операторов COMMIT или ROLLBACK нормальное завершение сеанса зафиксировывает транзакцию, а аварийное завершение произведёт откат транзакции. Если сбой произошёл на стороне сервера, то автоматически будет выполнен откат транзакций с использованием сведений из журналов транзакций при восстановлении нормального режима работы.

Для управления транзакциями в языке SQL используются операторы SET TRANSACTION, COMMIT WORK, SAVEPOINT, ROLLBACK WORK. Оператор SET TRANSACTION начинает транзакцию и позволяет установить её характеристики. Оператор COMMIT WORK фиксирует транзакцию и делает любые выполненные в ней изменения постоянными. Оператор SAVEPOINT позволяет создать точку сохранения во время транзакции, чтобы при необходимости можно было вернуться к состоянию в точке сохранения. Оператор ROLLBACK выполняет откат транзакции, то есть производит отмену изменений, выполненных данной транзакцией, в базе данных. Откат транзакции, как правило, выполняется, когда произошла ошибка в приложении, произошло нарушение ограничений целостности или обнаружена взаимная блокировка системных ресурсов (deadlock).

Взаимная блокировка – ситуация в СУБД, при которой несколько процессов находятся в состоянии бесконечного ожидания ресурсов, занятых самими этими процессами.

*Иллюстрация взаимной блокировки двух процессов*

Шаг	Процесс 1	Процесс 2
1	Запросил ресурс X	Запросил ресурс Y
2	Получил в монопольное использование ресурс X	Получил в монопольное использование ресурс Y
3	Запросил в монопольное использование ресурс Y	Запросил в монопольное использование ресурс X
4	Ожидает освобождения ресурса Y	Ожидает освобождения ресурса X
5	Взаимная блокировка	

Решением этой проблемы является предоставления пользовательским приложениям возможности определять ситуацию взаимной блокировки и выполнять откат транзакции.

Для фиксации транзакции используется оператор COMMIT WORK, который обеспечивает выполнение следующих действий:

- фиксация транзакции и становление изменений, сделанных в текущей транзакции, постоянными в базе данных;
- завершение транзакции;
- уничтожение точек сохранения для данной транзакции;
- освобождение объектов, заблокированных в процессе выполнения транзакции.

При выполнении транзакции, содержащей операторы изменения данных, выполняются действия:

- создание копии данных в сегментах отката;
- выполнение формирования соответствующих записей в журнальный файл;
- вносятся изменения в буферы базы данных.

В процессе явного или не явного завершения транзакции, происходит выполнение операций следующих операций:

- отметка транзакции, как зафиксированной;

- при отсутствии записи из журнального файла в файлах базы данных, происходит запись в долговременную память;
- освобождение заблокированных системных объектов.

После выполнения оператора COMMIT WORK информация из оперативной памяти сервера записывается в долговременную память.

Так как выполнение транзакции в больших базах данных может происходить продолжительное время, во многих СУБД реализовано создание промежуточных точек сохранения, в которых фиксируются частичные результаты выполнения транзакции.

Для разделения длительной транзакции на логические точки сохранения используется оператор SAVEPOINT. Сохранение промежуточных изменений в определённых точках транзакции позволяет при возникновении ошибки или взаимной блокировки выполнить откат не к началу транзакции, а к последней успешно выполненной точке сохранения.

Для отката транзакции к точке сохранения или исходному состоянию используется оператор ROLLBACK. При использовании данного оператора, происходит выполнение следующих действий:

- завершение выполнения транзакции;
- отмена всех изменений в текущей транзакции;
- отмена всех блокировок транзакции.

### *Заключение*

В заключение следует отметить, что при написании прикладных программ рекомендуется явно завершать транзакцию и использовать оператор COMMIT WORK перед завершением сеанса. Применение такого подхода позволит производить автоматический откат неуспешной транзакции с прогнозируемым результатом в случае аварийного завершения прикладного процесса.

### *Список литературы*

1. Смирнов С.Н. Безопасность систем баз данных. – М.: Гелиос АРВ, 2007. – 352 с.

2. Дейт К.Дж. Введение в системы баз данных. – 8-е изд. – М.: Вильямс, 2005. – 1328 с.