



# ТРАНЗАКЦИИ И ПАРАЛЛЕЛИЗМ



# ТРАНЗАКЦИЯ

или логическая единица работы, — это в общем случае последовательность ряда таких операций, которые преобразуют некоторое непротиворечивое состояние базы данных в другое непротиворечивое состояние, не гарантируя сохранения непротиворечивости во все промежуточные моменты времени.



## ПРЕДЛОЖЕНИЕ COMMIT

явным образом закрывает открытую транзакцию и делает изменения в базе данных постоянными.

Открываться транзакции могут неявно, например, при выполнении предложений INSERT, DELETE или UPDATE, или явно, с помощью предложения START (в Oracle все транзакции запускаются неявно).

В любом случае открытая транзакция закрывается явно предложением COMMIT.

Синтаксис:

***COMMIT [WORK]***

Здесь необязательное слово WORK является пустым и не оказывает никакого влияния.



# ПРЕДЛОЖЕНИЕ ROLLBACK

возвращает транзакцию в ее исходное состояние или к определенной, заранее заданной точке сохранения (SAVEPOINT). Кроме того, это предложение закрывает все открытые курсоры.

Синтаксис предложения имеет вид:

*ROLLBACK [WORK] [TO SAVEPOINT имя\_точки\_сохранения]*

необязательное слово WORK является пустым и не оказывает никакого влияния;

[TO SAVEPOINT имя\_точки\_сохранения] позволяет не отменять всю транзакцию, а откатить ее к указанной точке сохранения (т. е. выполнить частичный откат).

Параметр имя\_точки\_сохранения может представлять собой постоянное выражение или переменную.

Если TO SAVEPOINT опущено, то закрываются все курсоры, в противном случае закрываются только те курсоры, которые были открыты соответствующим предложением SAVEPOINT.



# ПРЕДЛОЖЕНИЕ SAVEPOINT

разделяет транзакцию на логические точки сохранения.

В одной транзакции может быть несколько точек сохранения.

Синтаксис этого предложения имеет вид:  
*SAVEPOINT имя\_точки\_сохранения*

В текущей транзакции устанавливается точка сохранения с именем *имя\_точки\_сохранения*.



# МНОГОПОЛЬЗОВАТЕЛЬСКИЙ РЕЖИМ РАБОТЫ

## *ПАРАЛЛЕЛИЗМ ТРАНЗАКЦИЙ*

Поддержание механизма транзакций — показатель уровня развитости СУБД и основа обеспечения целостности базы данных.

Транзакции составляют основу изолированности в многопользовательских системах, где с одной базой данных параллельно могут работать несколько пользователей и (или) прикладных программ.

Одна из основных задач СУБД — обеспечение изолированности, т. е. создание такого режима функционирования, при котором каждому пользователю казалось бы, что база данных доступна только ему. Такую задачу СУБД принято называть **параллелизмом транзакций**.



При выполнении транзакции система управления базами данных должна придерживаться определенных правил обработки набора команд, входящих в транзакцию.

Разработано четыре правила, известные как требования ACID (Atomicity, Consistency, Isolation, Durability – неделимость, согласованность, изолированность, устойчивость), гарантирующих правильность и надежность работы системы.



1. Транзакция **неделима** в том смысле, что представляет собой единое целое. Все ее компоненты либо имеют место, либо нет. Не бывает частичной транзакции. Если может быть выполнена лишь часть транзакции, она отклоняется.
2. Транзакция является **согласованной**, потому что не нарушает бизнес-логику и отношения между элементами данных. Это свойство очень важно при разработке клиент-серверных систем, поскольку в хранилище данных поступает большое количество транзакций от разных систем и объектов. Если хотя бы одна из них нарушит целостность данных, то все остальные могут выдать неверные результаты.
3. Транзакция всегда **изолирована**, поскольку ее результаты самодостаточны. Они не зависят от предыдущих или последующих транзакций — это свойство называется сериализуемостью и означает, что транзакции в последовательности независимы.
4. Транзакция **устойчива**. После своего завершения она сохраняется в системе, которую ничто не может вернуть в исходное (до начала транзакции)



# БЛОКИРОВКА

- временное ограничение на выполнение некоторых операций обработки данных.

Блокировка может быть наложена как на отдельную строку таблицы, так и на всю базу данных.

Управлением блокировками на сервере занимается менеджер блокировок, контролирующей их применение и разрешение конфликтов.

Транзакции и блокировки тесно связаны друг с другом.

Транзакции накладывают блокировки на данные, чтобы обеспечить выполнение требований ACID.

Без использования блокировок несколько транзакций могли бы изменять одни и те же данные.

Повышение эффективности работы при использовании небольших транзакций связано с тем, что при выполнении транзакции сервер накладывается на данные блокировки.



Блокировка представляет собой метод управления параллельными процессами, при котором объект базы данных не может быть модифицирован без ведома транзакции.

Производится блокирование доступа к объекту со стороны других транзакций, чем исключается непредсказуемое изменение объекта.



## Виды блокировок

- блокировка записи — транзакция блокирует строки в таблицах таким образом, что запрос другой транзакции к этим строкам будет отменен;
- блокировка чтения — транзакция блокирует строки так, что запрос со стороны другой транзакции на блокировку записи этих строк будет отвергнут, а на блокировку чтения — принят.



В СУБД используют протокол доступа к данным, позволяющий избежать проблемы параллелизма. Его суть заключается в следующем:

- транзакция, результатом действия которой на строку данных в таблице является ее извлечение, обязана наложить блокировку чтения на эту строку;
- транзакция, предназначенная для модификации строки данных, накладывает на нее блокировку записи;
- если запрашиваемая блокировка на строку отвергается из-за уже имеющейся блокировки, то транзакция переводится в режим ожидания до тех пор, пока блокировка не будет снята;
- блокировка записи сохраняется вплоть до конца выполнения транзакции.



Решение проблемы параллельной обработки базы данных заключается в том, что строки таблиц блокируются, а последующие транзакции, модифицирующие эти строки, отвергаются и переводятся в режим ожидания.

В связи со свойством сохранения целостности базы данных транзакции являются подходящими единицами изолированности пользователей.



# УРОВНИ БЛОКИРОВАНИЯ

- уровень 0 — запрещение "загрязнения" данных. Этот уровень требует, чтобы изменять данные могла только одна транзакция; если другой транзакции необходимо изменить те же данные, она должна ожидать завершения первой транзакции;
- уровень 1 — запрещение "грязного" чтения. Если транзакция начала изменение данных, то никакая другая транзакция не сможет прочитать их до завершения первой;
- уровень 2 — запрещение неповторяемого чтения. Если транзакция считывает данные, то никакая другая транзакция не сможет их изменить. Таким образом, при повторном чтении они будут находиться в первоначальном состоянии;
- уровень 3 — запрещение фантомов. Если транзакция обращается к данным, то никакая другая транзакция не сможет добавить новые или удалить имеющиеся строки, которые могут быть считаны при выполнении транзакции. Реализация этого уровня блокирования выполняется путем использования блокировок диапазона ключей. Подобная блокировка накладывается не на конкретные строки таблицы,