

Понятие транзакции

Лекция 15

МГТУ им. Н.Э.Баумана. ФН12

Понятие транзакции

Перевести 1000 рублей с лицевого счета клиента с номером 1 на лицевой счет клиента с номером 2.

```
UPDATE Client_Balance SET ClientSum=ClientSum - 1000  
WHERE ClientId=1;
```

```
UPDATE Client_Balance SET ClientSum=ClientSum + 1000  
WHERE ClientId =2;
```

Если первый запрос выполнен, а вот второй - нет, то база данных будет в несогласованном состоянии, и такая ситуация грозит реальными денежными потерями.

Оба запроса должны выполняться как одно целое.

Транзакция - это последовательность операторов манипулирования данными, выполняющаяся *как единое целое* и переводящая базу данных из одного целостного состояния в другое целостное состояние.

Выполняется по принципу «все или ничего», все, предписанные транзакцией, действия или должны быть выполнены, или все эти действия должны остаться невыполненными, то есть транзакция должна быть атомарной.

Транзакции являются единицами *восстановления* данных после сбоев.

Транзакции не могут быть вложенными.

MySQL, по умолчанию, работает в режиме автоматической фиксации транзакций (режим autocommit).

Любой отдельный оператор SQL, изменяющий данные, всегда атомарен и результаты его выполнения будут сразу сохраняться.

Все изменения, вносимые в базу данных в ходе каждой транзакции, записываются в журнал.

При откате транзакции отменяется все проведенные инструкции в соответствии с журналом.

Журнал используется в случае непредвиденного сбоя для восстановления согласованного состояние базы данных.

Ведение журнала транзакций заставляет систему работать медленнее.

Транзакцию открывает оператор `START TRANSACTION`, режим автоматического завершения транзакций отдельных операторов до оператора `COMMIT` отключается.

Завершает транзакцию оператор `COMMIT`. Все результаты работы транзакции фиксируются (сохраняются) в базе данных.

Оператор `ROLLBACK` отменяет(откатывает) все изменения, сделанные текущей транзакцией и возвращает базу данных в начальное состояние.

Пример.

```
START TRANSACTION;  
UPDATE Client_Balance SET ClientSum=ClientSum - 1000  
WHERE ClientId=1;  
UPDATE Client_Balance SET ClientSum=ClientSum + 1000  
WHERE ClientId =2;  
COMMIT;
```

Оператор `START TRANSACTION` выключает режим автоматического завершения (режим `autocommit`) транзакций.

Режим `autocommit` остается выключенным до явного завершения транзакции оператором `COMMIT` или оператором отката `ROLLBACK`.

В MySQL нельзя выполнить откат с помощью ROLLBACK для операторов относящихся категории DDL.

CREATE DATABASE, CREATE TABLE, CREATE INDEX,
DROP DATABASE, DROP TABLE, DROP INDEX,
ALTER DATABASE, ALTER TABLE,
RENAME TABLE.

Если оператор, следующий за операторами определения данных, завершится аварийно, то все изменения, выполненные во время транзакции не могут быть отменены оператором ROLLBACK.

Операторы категории DDL неявно завершают транзакцию.

Можно установить именованную точку до которой произойдет откат транзакции оператором `SAVEPOINT`. Оператор `ROLLBACK TO SAVEPOINT` откатывает транзакцию к именованной точке сохранения, все модификации строк, выполненные транзакцией после этой точки, отменяются. Точки сохранения, установленные в более поздние моменты, чем именованная точка, удаляются. Если текущая транзакция уже имеет точку сохранения с именем, определенным оператором `SAVEPOINT`, старая точка удаляется и устанавливается новая. При выполнении операторов `COMMIT` или `ROLLBACK` без указания имени точки сохранения, все точки сохранения транзакций удаляются. Если названная точка сохранения не существует, то оператор `ROLLBACK TO SAVEPOINT` возвращает ошибку.

```
SAVEPOINT NamePoint  
ROLLBACK TO SAVEPOINT
```

NamePoint

`SAVEPOINT` и `ROLLBACK TO SAVEPOINT` поддерживаются в таблицах `INNODB`, начиная с версий `MySQL 4.1.1`.

Требования ACID:

Atomicity - **Атомарность**.

Consistency - **Согласованность**.

Isolation - **Изоляция**.

Durability — **Надежность** иногда в русскоязычной литературе это правило часто называется **Долговечность** или **устойчивость**.

Атомарность.

Атомарность гарантирует, что никакая транзакция не будет зафиксирована в системе частично.

Если транзакцию не удастся полностью завершить, то результаты всех действий уже выполненных, будут отменены, система вернётся в исходное состояние, произойдет «откат» (rollback).

Согласованность (Consistency).

Транзакция переводит базу данных из одного согласованного (целостного) состояния в другое согласованное (целостное) состояние.

Система должна находиться в согласованном состоянии до начала транзакции и должна остаться в согласованном состоянии после завершения транзакции. Требование согласованности не является эквивалентным требованию целостности БД.

Например, если какая-либо транзакция произведёт списание со счета, но не произведёт зачисление, то система останется в несогласованном состоянии.

Операция списание со счета и зачисление на счет две разные операциями, между их выполнением внутри транзакции БД будет в несогласованном состоянии.

Внутри самой транзакции согласованность базы данных может нарушаться.

Изоляция (Isolation)

Транзакции не воздействуют одна на другую в процессе выполнения, все транзакции являются изолированными друг от друга.

Каждую транзакцию можно представить единственным объектом, наделенным полномочиями считывать и изменять данные, для этого используется блокировка.

Все изменения, вызываемые каждой из транзакций, становятся доступными лишь после их завершения.

Надежность (*Durability*)

Надежность (Долговечность или устойчивость). После того, как транзакция выполнена, результаты ее работы должны сохраниться в базе данных, даже если в следующий момент произойдет сбой системы.

Если сбой произошел во время выполнения транзакции, то необходимо на основе резервной копии или (и) файлов регистрации привести базу данных в состояние, предшествующее сбою.

В MySQL: следующие уровни изоляции транзакций:

1. **READ UNCOMMITTED**
2. **READ COMMITTED**
3. **REPEATABLE READ**

1. **READ UNCOMMITTED.** Часто называется dirty read («грязное чтение»). Этот уровень редко используется на практике.
2. **READ COMMITTED.** Прочитать данные возможно только после вызова COMMIT. Внутри транзакции данные тоже будут не доступны.

3. REPEATABLE READ. Этот уровень используется по умолчанию в MySQL. Вновь добавленные данные доступны внутри транзакции, но не доступны извне до подтверждения.

Может возникнуть проблема «фантомного чтения». То есть, когда внутри одной транзакции происходит чтение данных, а в этот момент внутри другой транзакции происходит вставка новых данных, то в первой - доступны старые данные.

4. SERIALIZABLE

На данном уровне MySQL блокирует каждую строку над которой происходит какое либо действие, это исключает появление проблемы «фантомов».

Посмотреть текущий уровень изоляции можно командой:
`SHOW VARIABLES LIKE '%tx_isolation%';`

Установить уровень изоляции можно командой:
`SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;`