

Транзакции и блокировки

Лекция №8

Бутенко И.В. 2017 год

Определение

- Транзакция представляет собой набор из одной или более команд, выполняемых как одно целое.
- Транзакция является единственной единицей работы. Если транзакция выполнена успешно, все модификации данных, сделанные в течение транзакции, принимаются и становятся постоянной частью базы данных. Если в результате выполнения транзакции происходят ошибки и должна быть произведена отмена или выполнен откат, все модификации данных будут отменены.

Требования к выполнению транзакций (ACID)

- Атомарность (atomicity) – Все изменения данных в одной транзакции, рассматриваются как единый минимальный блок.
- Согласованность (consistency) – После того, как транзакция будет успешно завершена, данные должны удовлетворять всем ограничениям целостности, определенным в БД.
- Изолированность (isolation) – Изменения данных, выполняемые различными транзакциями, должны быть независимы друг от друга. То есть изменения данных выполняемые одной транзакцией не должны зависеть от изменений, выполняемых другой транзакцией.
- Устойчивость или долговечность (durability) – После того, как транзакция выполнит все необходимые изменения, и ее работа будет завершена, система выполняет фиксирование транзакции (commit). После этого система не может быть возвращена в то состояние, в котором была до начала транзакции.
- SQL Server автоматически гарантирует выполнение требований ACID.

Режимы транзакций в SQL Server

- Автоматическое определение транзакции (по умолчанию)
- Явное определение транзакции
- Неявное определение транзакции

Автоматическое определение транзакции

- При работе в этом режиме пользователь не должен указывать в явном виде начало и конец транзакции.

```
SET IMPLICIT_TRANSACTION OFF
```

Явное определение транзакции

Необходимо явно указывать начало и конец транзакции.

- Команды для управления транзакциями:
- `Begin tran` – предназначена для обозначения начала транзакции.
- Синтаксис
- `Save tran` – предназначена для создания точки сохранения (save point)
- `Rollback Tran` – откат транзакции. Данные, которые изменились в ходе транзакции восстанавливаются в начальное состояние.
- `Commit tran` – фиксация транзакции

Пример

```
create table tmp_aa ( name varchar(20) )
```

```
begin tran  
insert into tmp_aa  
values ('1')
```

```
save tran aaa
```

```
insert into tmp_aa  
values ('2')
```

```
rollback tran aaa
```

```
select * from tmp_aa
```

```
rollback tran
```

```
select * from tmp_aa
```

```
begin tran  
insert into tmp_aa  
values ('1')
```

```
begin tran
```

```
insert into tmp_aa  
values ('2')
```

```
commit tran
```

```
select * from tmp_aa
```

```
rollback tran
```

```
select * from tmp_aa
```

Неявное определение транзакции

Система автоматически начинает новую транзакцию после завершения предыдущей транзакции.

Автоматическое открытие транзакции при операциях:

- Alter table
 - Create
 - Delete
 - Drop
 - Fetch
 - Grant
 - Select
 - Insert
 - Update
 - Revoke
 - Truncate table
-
- `set implicit_transactions on`

Вложенные транзакции

- Переменная @@trancount показывает текущую степень вложенности транзакции.
- Каждая инструкция BEGIN TRANSACTION увеличивает значение @@TRANSCOUNT на один.
- Каждая инструкция COMMIT TRANSACTION уменьшает значение @@TRANSCOUNT на один.
- Инструкции ROLLBACK TRANSACTION без имен транзакций откатывают все вложенные транзакции и уменьшают значение @@TRANSCOUNT до 0.

Основы блокировок

- Чтобы обеспечить выполнение требования изолированности, SQL Server использует механизм блокировок (locks).
- Блокировкой называется временное ограничение, накладываемое системой на использование тех или иных ресурсов.

Типы блокировок

- На отдельную строку таблицы (RID)
- Диапазон индекса (key)
- На страницы (page)
- На группы страниц (extent)
- На всю таблицу целиком (table)

Блокировками управляет диспетчер блокировок Lock Manager.

Мертвые блокировки

- Транзакция А создает общую блокировку строки 1.
- Транзакция Б создает общую блокировку строки 2.
- Транзакция А теперь запрашивает монопольную блокировку строки 2 и блокируется до того, как транзакция Б закончится и освободит общую блокировку строки 2.
- Транзакция Б теперь запрашивает монопольную блокировку строки 1 и блокируется до того, как транзакция А закончится и освободит общую блокировку строки 1.

Механизмы блокирования

- Level 0. No trashing of data (запрещение загрождения данных). Одни и те же данные в каждый момент времени может изменять только одна транзакция. Если другая транзакция пытается изменить эти же данные, она должна дождаться завершения работы первой транзакции.
- Level 1. No dirty read (запрещение грязного чтения). Когда транзакция начинает менять данные, СУБД должна блокировать ресурсы, чтобы ни одна транзакция не смогла прочитать заблокированные данные.
- Level 2. No nonrepeatable read (запрещение неповторяемого чтения). Когда транзакция начинает читать данные, СУБД должна блокировать ресурсы, чтобы ни одна транзакция не смогла изменить заблокированные данные.
- Level 3. No phantom (запрещение фантомов). Если транзакция производит выборку по какому либо логическому условию, то никакая другая транзакция не должна вставлять в таблицу или удалять из нее строки, удовлетворяющие этому условию.

Использование блокировок

- Основное назначение блокировок – обеспечение нормальной работы множества пользователей с одними и теми же данными.

Два типа конкуренции:

- Оптимистическая. При операциях чтения из транзакции не происходит блокирования используемых ресурсов. Нарушение требования изолированности, но повышение производительности.
- Пессимистическая. Система блокирования строго следует требованиям ACID.

SQL Server работает с двумя видами конкуренции.

Уровни изолированности транзакций

Уровни изоляции транзакций контролируют следующие параметры:

- Применение и типы блокировки при чтении данных.
- Время удержания блокировок чтения.
- Использование операции чтения ссылок на строки, измененные другой транзакцией:
 - Блокировка до тех пор, пока не будет снята монопольная блокировка строки.
 - Извлечение зафиксированной версии строки, которая существовала в то время, когда началось выполнение инструкции или транзакции.
 - Считывание незафиксированного изменения данных.

Уровни изолированности транзакций

- READ UNCOMMITTED - level0
- READ COMMITTED – level1
- REPEATABLE READ – level2
- SERIALIZABLE – level3
- SNAPSHOT ISOLATION. Данные не блокируются. Поддерживаются версии данных.
- Установка: `set transaction isolation level`

Управление блокировками на уровне команд

- Можно управлять уровнем изоляции не только на уровне соединения, но и на уровне запроса. Для этого в SQL Server используются специальные ключевые слова – хинты (hints).

Примеры:

- Rowlock
- Paglock
- Tablock
- Readuncommitted
- Nolock