ОНЛАЙН-ОБРАЗОВАНИЕ

OTUS

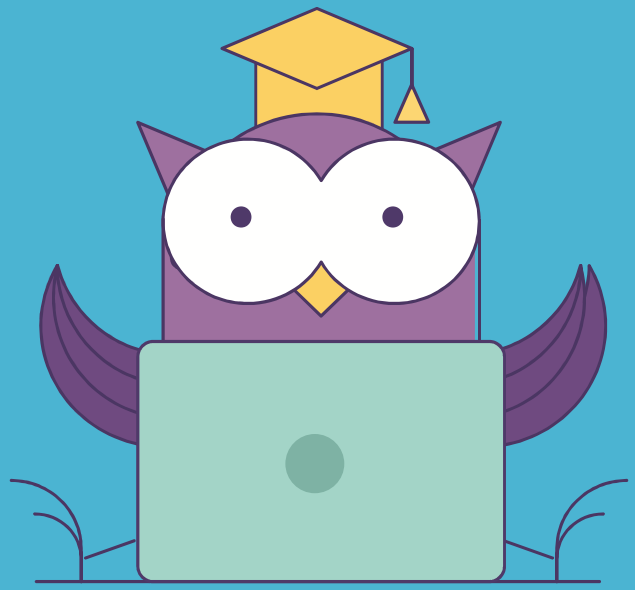
Уровни изоляции транзакций

Курс «Разработчик MS SQL Server»

Занятие № 13

Кристина Кучерова

Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо
Ставьте - если есть проблемы

Цель вебинара

По окончании вебинара вы сможете:

- Объяснить зачем нужны разные уровни изоляции и что от них зависит
- Делать правильную обработку ошибок
- Читать лог SQL Server
- Анализировать дедлоки

Маршрут вебинара

Уровни изоляции

Обработка ошибок

Deadlock

Лог

Выводы

Что было на прошлом занятии?

OTUS

- Как храняться xml, json?
- Чем отличается Exec и sp_executesql
- Что вернется в результате выполнения скрипта?

BEGIN TRANSACTION

✓ COMMIT

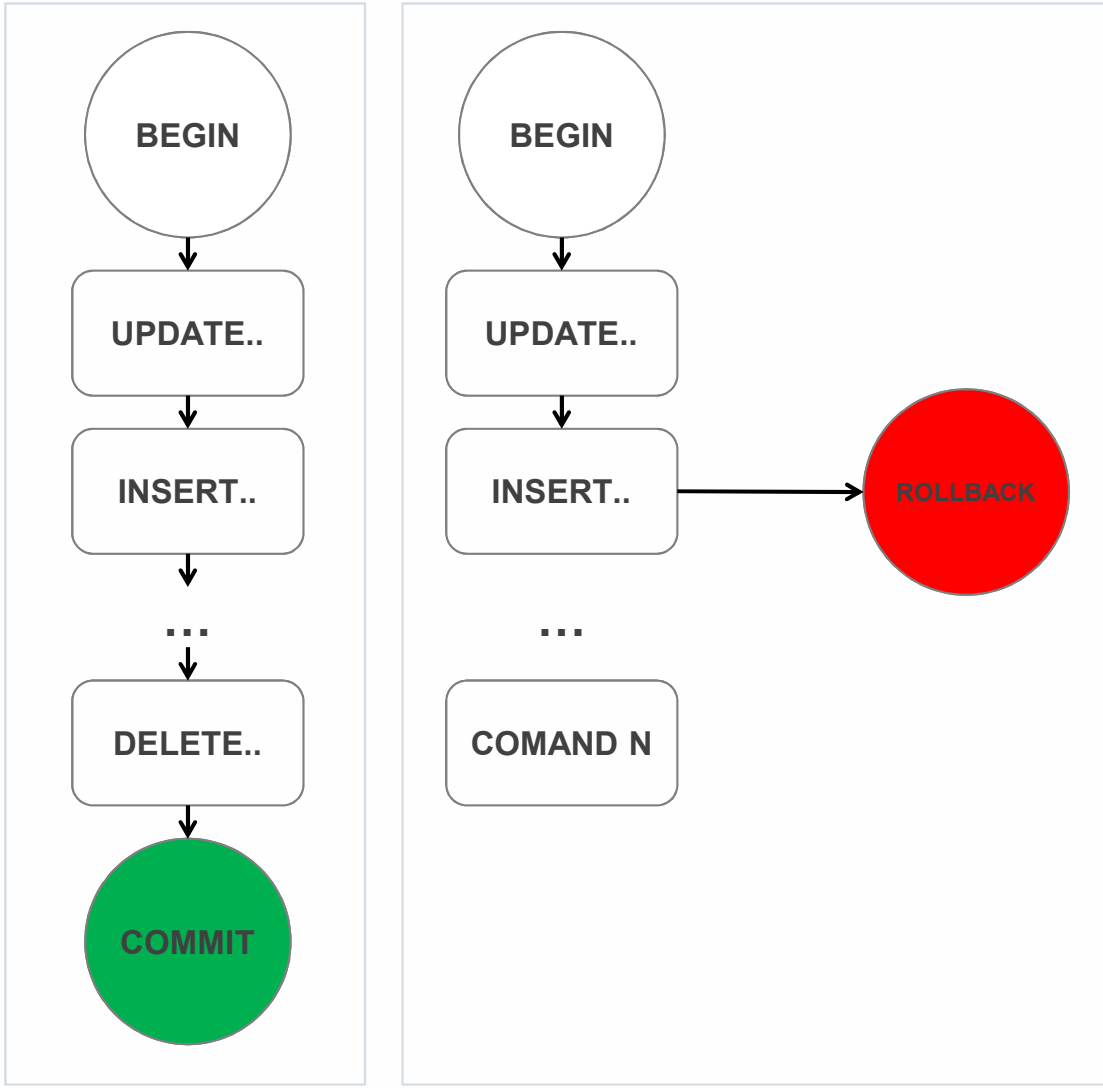
ROLLBACK

```
begin transaction

delete from city
where city = 'Bergamo';

insert into city
(city, country_id)
values ('Pskov',80);

rollback
```



Зачем нужны транзакции?

OTUS

Есть ваша карта и дополнительная карта у вашей супруги

Изоляция позволяет выдать вам правильную сумму денег если вы воспользуетесь банкоматами одновременно.

Все транзакции обладают свойствами:

- **A** atomicy - атомарность
- **C** consistency – согласованность
- **I** isolation – изолированность
- **D** durability - долговечность

«Все или ничего»

Либо все изменения транзакции фиксируются (commit),

либо все откатываются (rollback)

Вы начали покупку билета, билет заблокировался, но деньги с вашего счета не списались, администратор «убил» транзакцию.

Билет останется заблокированным?

Соблюдение целостности данных – никакая транзакция не может примениться, если целостность нарушена

ID фильма	Название	Год выпуска	ID жанра
1	Звездные войны	1977	1
2	Один дома	1990	2
3	Пила-99	2017	3
4	Странная история про странного мальчика	2009	4

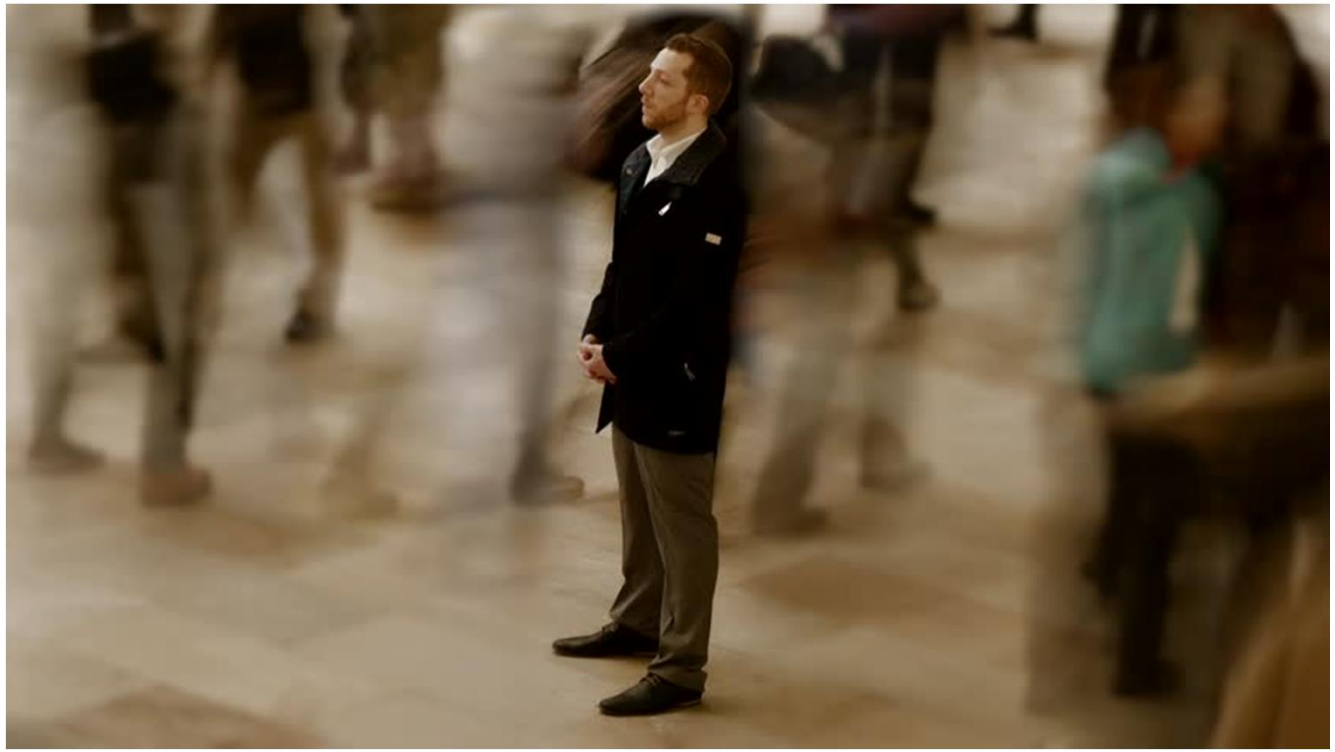
ID жанра	Жанр
1	Приключения
2	Комедия
3	Ужасы
4	Артхаус

Если у вас есть Foreign Key, Constraints, которые выдали ошибку – значит есть нарушение целостности

I isolation Изоляция

Мы были в толпе, но казалось, что мы одни

Все СУБД многопользовательские, и изоляция – это создание иллюзии того, что пользователь работает в одиночку не видит незафиксированных транзакций



D durability Долговечность

Написанного пером не вырубишь топором.

Все что зафиксировано, должно остаться в БД

Как этого добиваются? Как сделать, чтобы все транзакции в случае сбоя остались в БД?

В лог транзакций. На диск. – поэтому БД часто медленные – они ждут пока изменения запишутся на диск.

Write-ahead log или упреждающая журнализация.

Какие уровни изоляции есть в SQL Server?

OTUS

1. Read Uncommitted - Чтение незаконченных транзакций
2. Read Committed – Чтение подтвержденных данных
3. Repeatable read
4. Snapshot - При разрешенном режиме версионности
5. Serializable

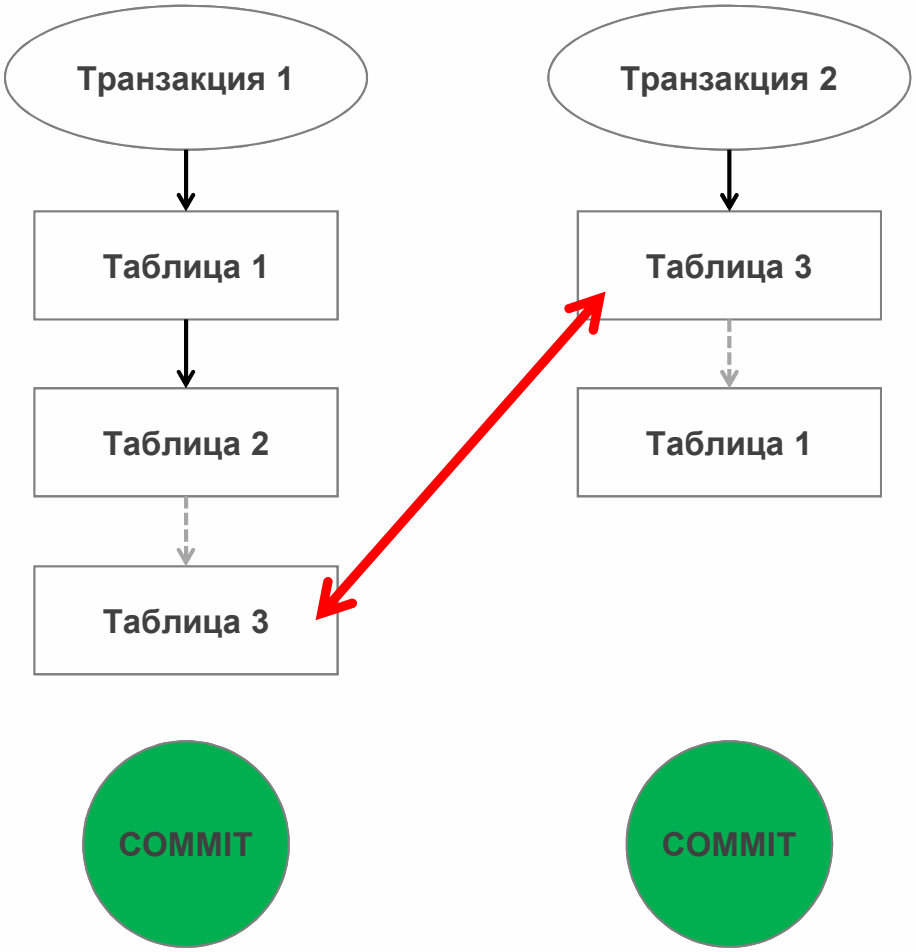
Команды транзакции

OTUS

Команда	Описание
BEGIN TRANSACTION BEGIN TRAN	Начало транзакции
COMMIT TRANSACTION COMMIT WORK COMMIT	Подтверждение транзакции
ROLLBACK TRANSACTION ROLLBACK WORK ROLLBACK	Откат транзакции
SAVE TRANSACTION SAVE TRAN SAVE	Создание точки сохранение транзакции
@@TRANCOUNT	Количество транзакций
XACT_STATE()	Статус текущей транзакции

Deadlock

Взаимная блокировка ресурса двумя транзакциями



Как понять, что вызывает deadlock?

OTUS

Как вы узнаете, что у вас есть дедлок?

```
Msg 1205, Level 13, State 51, Line 13
Transaction (Process ID 56) was deadlocked on lock resources with
another process and has been chosen as the deadlock victim. Rerun
the transaction.
```

Флаги трассировки

1222

1204

SET DEADLOCK_PRIORITY LOW | NORMAL | HIGH

Аномалии изменения данных в БД

OTUS

- **Dirty read** (грязное чтение) - ситуациā когда одна транзакция может прочитать данные измененные другой транзакцией, но не записанные (uncommited).

  Например, у вас есть друг писатель – и вы просите почитать его неизданную книгу, но потом оказывается, что в изданной книге все немного не так,

- **Non repeatable read** (неповторяющееся чтение) - когда в течении транзакции при повторном чтении одних и тех же строк - читаются разные данные. Был проведен update другой транзакцией.

- **Phantom Read** (фантомное чтение) - когда в течении транзакции при повторном выполнении запроса с одними и теми же условиями результат возвращает разное кол-во строк. Был проведен insert/delete другой транзакцией.

Аномалии и уровни изоляции транзакций

OTUS

Уровень изоляции	Dirty read	Non repeatable read	Phantom read
Read Uncommitted	Да	Да	Да
Read Committed	Нет	Да	Да
Repeatable read	Нет	Нет	Да
Snapshot	Нет	Нет	Нет
Serializable	Нет	Нет	Нет

Что такое блокировка?

OTUS

Ограничение доступа к ресурсу

Это моя корова и я ее дою!

Какие блокировки бывают?

OTUS

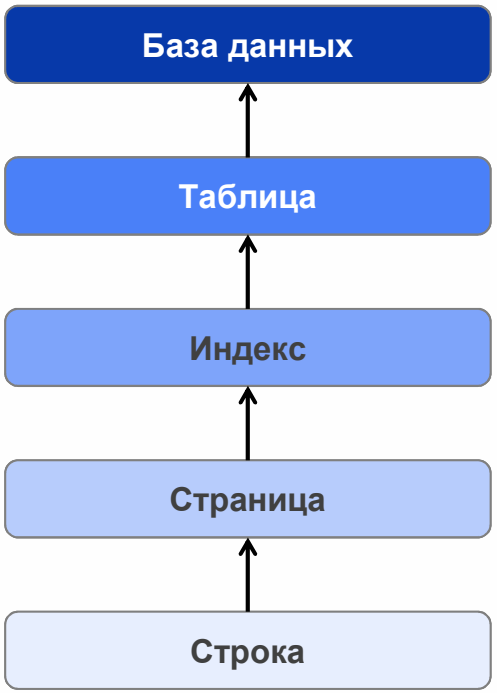
1. Shared - разделяемая – корова в колхозе
2. Update - на изменение записей – корова на семью
3. Exclusive – эксклюзивная – корова только для вас

Гранулярность и эскалация блокировок

Блокировка строки

Если сервер понял, что ему нужно заблокировать много строк – он думает а не заблокировать ли страницы.

А если много страниц – может дешевле заблокировать всю таблицу?



RAISERROR ({ msg_id | msg_str | @local_variable }
{ ,severity ,state }
[,argument [,...n]])
[WITH option [,...n]]

THROW [{ error_number | @local_variable },
{ message | @local_variable },
{ state | @local_variable }
] [;]

@@ERROR

XACT_ABORT
TRY/CATCH

Обработка ошибок

@@ERROR

XACT_ABORT
TRY/CATCH

ERROR_NUMBER()
ERROR_MESSAGE()
ERROR_LINE()
ERROR_SEVERITY()
ERROR_STATE()

Попробуем представить, что у вас своя пиццерия

Вы повар в пиццерии.

Вам надо знать сколько пицц готовить.

Вы приходите и спрашиваете какие пиццы заказаны и заказы подтверждены.

Какой уровень изоляции вам подойдет?

OTUS

1. Read Uncommitted
2. Read Committed
3. Repeatable read
4. Snapshot
5. Serializable

Вы повар в пиццерии.

Вам надо знать сколько пицц готовить.

Вы приходите и спрашиваете какие пиццы заказаны и заказы подтверждены.

Ответ:

Read Committed

Накладывает Shared lock

Вы хозяин пиццерии. Операторы принимают заказы.

Сколько всего заказов есть в системе?

Точность не так важна, можно посчитать и те, что еще в процессе заказа.

Какой уровень изоляции вам подойдет?

OTUS

1. Read Uncommitted
2. Read Committed
3. Repeatable read
4. Snapshot
5. Serializable

Владелец пиццерии

Вы хозяин пиццерии. Операторы принимают заказы.

Сколько всего заказов есть в системе?

Точность не так важна, можно посчитать и те, что еще в процессе заказа.

Это Read Uncommitted

(не накладывает блокировку на данные)

Или? Snapshot

Вы закупщик в пиццерии.

Вам надо знать сколько Маргарит готовить. У вас заканчивается сыр.

Вы просите операторов не изменять заказы, которые уже приняты, на Маргариту.

Какой уровень изоляции вам подойдет?

OTUS

1. Read Uncommitted
2. Read Committed
3. Repeatable read
4. Snapshot
5. Serializable

Вы закупщик в пиццерии.

Вам надо знать сколько Маргарит готовить. У вас заканчивается сыр.

Вы просите операторов не изменять заказы, которые уже приняты, на Маргариту.

Это Repeatable Read

Блокирует Update и Delete

Вы санэпидемстанция.

Вы приходите в пиццерию и говорите

«Никому не двигаться и ничего не менять пока я все не проверю»

Какой уровень изоляции вам подойдет?

OTUS

1. Read Uncommitted
2. Read Committed
3. Repeatable read
4. Snapshot
5. Serializable

Вы санэпидемстанция.

Вы приходите в пиццерию и говорите

«Никому не двигаться и ничего не менять пока я все не проверю»

Serializable

Merge join

OTUS

Используется, когда оба набора имеют индекс

Может использовать tempdb если в первом наборе много дубликатов

Лучший вариант для больших наборов данных

Рефлексия

О чем мы говорили сегодня?

- Какие уровни изоляции бывают?
- Как избежать блокировки?
- Как обрабатывать ошибки?
- Что такое deadlock?

Рефлексия

Напишите, пожалуйста, свое впечатление о вебинаре.

- Отметьте 3 пункта, которые вам запомнились с вебинара.
- Что вы будете применять в работе из сегодняшнего вебинара?

Заполните, пожалуйста, опрос в ЛК о занятии

Спасибо
за внимание!
До встречи в Slack и на вебинаре