

Технологии обработки транзакций клиент-серверных приложений зима-весна 2022

ФИО преподавателя: Матчин Василий Тимофеевич

e-mail: matchin@mirea.ru

[Online-edu.mirea.ru](https://online-edu.mirea.ru)

online.mirea.ru

Условия обучения

- По итогам изучения дисциплины проводится экзамен
- В течение семестра необходимо выполнить все задания по календарному плану, которые опубликованы на Учебном портале
- Баллы за активность до 25 баллов

ТЕМА

Изоляция транзакций

Реализация уровней изоляции

Существует два глобально различных подхода к реализации изолированности: блокирование и версионирование.

Версионирование (snapshot)

Блокирование (lock)

Для блокировок характерно понятие оптимистичной и пессимистичной блокировки.

Уровни изоляции транзакций

Стандарт SQL определяет четыре уровня изоляции транзакций:

Read uncommitted (Чтение незафиксированных данных)

Read committed (Чтение зафиксированных данных)

Repeatable read (Повторяемое чтение)

Serializable (Сериализуемость)

Особые условия изоляции транзакций

При работе транзакций существуют особые условия

«Грязное» чтение (Dirty Reads)

Неповторяемое чтение (None-Repeatable Reads)

Фантомное чтение (Phantom Reads)

Аномалия сериализации/Потерянное обновление (Lost Update)

Уровни изоляции и особые условия

Уровень изоляции	Особые условия	«Грязное» чтение (Dirty Reads)	Неповторяемое чтение (None-Repeatable Reads)	Фантомное чтение (Phantom Reads)	Аномалия сериализации (Lost Update)
Read uncommitted (Чтение незафиксированных данных)		Да, но не в PG	Да	Да	Да
Read committed (Чтение зафиксированных данных)		Нет	Да	Да	Да
Repeatable read (Повторяемое чтение)		Нет	Нет	Да, но не в PG	Да
Serializable (Сериализуемость)		Нет	Нет	Нет	Нет

Особенности в PostgreSQL

Команды определены в стандарте SQL, за исключением режима транзакции DEFERRABLE и формы SET TRANSACTION SNAPSHOT, которые являются расширениями PostgreSQL.

Уровень изоляции Read Uncommitted

Уровень, имеющий самую плохую согласованность данных, но самую высокую скорость выполнения транзакций. Каждая транзакция видит незафиксированные изменения другой транзакции (феномен грязного чтения).

Выполняются две транзакции T1 и T2. T1 вносит изменения в данные с помощью операторов `INSERT`, `DELETE`, `UPDATE`. При этом T2 видит данные другой транзакции, которые еще не были зафиксированы.

При этом уровне изоляции, в случае отката T1 данные полученные T2 окажутся ошибочными. Таким образом, наблюдается феномен грязного чтения.

Уровень изоляции Read committed

На этом уровне параллельно исполняющиеся транзакции видят только зафиксированные изменения из других транзакций. Таким образом, данный уровень обеспечивает защиту от грязного чтения.

Выполняются две транзакции T1 и T2. T1 вносит изменения в данные с помощью операторов INSERT, DELETE, UPDATE. При этом T2 **НЕ** видит данные другой транзакции, которые еще не были зафиксированы. **НО** в T1 на данном этапе видны внесенные изменения.

После выполнения COMMIT, т.е. фиксации изменений транзакцией T1, другая транзакция T2 теперь увидит внесенные изменения.

Уровень изоляции Repeatable read

Уровень, позволяющий предотвратить феномен неповторяющегося чтения. При этом не видно в исполняющейся транзакции T2 измененные и удаленные записи другой транзакцией T1.

Выполняются две транзакции T1 и T2. В T1 выполняем запросы INSERT, UPDATE и DELETE. Затем, в T2 пытаемся обновить ту же самую строку, которую обновили в T1.

В такой ситуации T2 будет ждать, пока T1 зафиксирует изменения или откатится.

Таким образом, можно читать все изменения только своей транзакции. Данные, измененные другими транзакциями, недоступны.

Уровень изоляции Serializable

Транзакции могут выполняться только одна за другой. Медленная эффективность выполнения. В классическом представлении этот уровень избавляет от эффекта чтения фантомов.

Выполняются две транзакции T1 и T2. T2 читает таблицу table1 (делает SELECT), затем T1 пытается выполнить INSERT, UPDATE и DELETE для той же таблицы прочитанные T2 в рамках транзакции.

В такой ситуации T1 не может изменить данные, прочитанные T2. Поэтому T1 будет ждать, пока T2 завершит работу.

Значения по умолчанию для изоляции

В большинстве приложений уровень изолированности редко меняется и используется значение по умолчанию (например, в MySQL это repeatable read, в Oracle, MSSQL Server, PostgreSQL — read committed).

Однако на уровне операторов (SELECT, UPDATE и т.д.) в Oracle по умолчанию уже есть REPEATABLE READ, т.е. в рамках одного оператора всегда получается согласованное чтение, что достигается конечно же за счет сегмента отката.

Чем сильнее уровень изоляции, тем меньше производительность БД — потому что транзакциям чаще приходится ждать друг друга.

Установка требуемого уровня изоляции

Для выбора нужного уровня изоляции транзакций используется команда SET TRANSACTION.

```
SET TRANSACTION режим_транзакции [, ...]
```

Где режим_транзакции может быть следующим:

```
ISOLATION LEVEL { SERIALIZABLE | REPEATABLE  
READ | READ COMMITTED | READ  
UNCOMMITTED }  
READ WRITE | READ ONLY  
[ NOT ] DEFERRABLE
```

Пример SET TRANSACTION

Пример использования:

```
SET TRANSACTION ISOLATION LEVEL READ  
COMMITTED, READ WRITE;
```

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE  
READ, READ ONLY;
```

```
SET TRANSACTION ISOLATION LEVEL  
SERIALIZABLE;
```

Команда SET TRANSACTION SNAPSHOT

Команда SET TRANSACTION SNAPSHOT позволяет выполнить новую транзакцию со снимком данных, который имеет уже существующая транзакция.

```
SET TRANSACTION SNAPSHOT id_снимка
```

pg_export_snapshot — функция экспорта снимка возвращает id_снимка (например, 000003A1-1).

Особенности SET TRANSACTION

Если команде SET TRANSACTION не предшествует START TRANSACTION или BEGIN, она выдаёт предупреждение и больше ничего не делает.

Поэтому сначала нужно написать

START TRANSACTION

или

BEGIN

а затем

SET TRANSACTION

Транзакция со снимком данных

Начинаем транзакцию

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE  
READ;
```

```
SELECT pg_export_snapshot();
```

```
pg_export_snapshot
```

```
-----
```

```
00000003-0000001B-1
```

```
(1 row)
```

Передаем id команде SET TRANSACTION SNAPSHOT

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE  
READ;
```

```
SET TRANSACTION SNAPSHOT '00000003-0000001B-1';
```

Команда START TRANSACTION

Команда начинает новый блок транзакции

START TRANSACTION — начать блок транзакции
(команда описана в стандарте SQL)

START TRANSACTION режим_транзакции, [, ...]

Где режим_транзакции может быть следующим:

```
ISOLATION    LEVEL    {    SERIALIZABLE    |  
REPEATABLE READ | READ COMMITTED | READ  
UNCOMMITTED }  
READ WRITE | READ ONLY
```

Команда BEGIN

Команда начинает блок транзакции

BEGIN — начать блок транзакции

BEGIN режим_транзакции, [, ...]

Где режим_транзакции может быть следующим:

```
ISOLATION      LEVEL      {      SERIALIZABLE      |  
REPEATABLE READ | READ COMMITTED | READ  
UNCOMMITTED }  
READ WRITE | READ ONLY
```

Команда SAVEPOINT

Команда устанавливает новую точку сохранения в текущей транзакции.

SAVEPOINT — определяет новую точку сохранения в текущей транзакции

`SAVEPOINT имя_точки_сохранения`

Параметры:

`имя_точки_сохранения`

Имя, назначаемое новой точке сохранения.

Команда COMMIT

Команда начинает блок транзакции

COMMIT — начать блок транзакции

COMMIT [AND CHAIN]

Параметры:

AND CHAIN

начинается новая транзакция с такими же
характеристиками транзакции

Команда ROLLBACK

Команда начинает блок транзакции

ROLLBACK — прерывает текущую транзакцию

ROLLBACK [AND CHAIN]

Параметры:

AND CHAIN

начинается новая транзакция с такими же характеристиками транзакции

Команда ROLLBACK TO SAVEPOINT

Команда возвращает изменения в базе данных к точке сохранения

ROLLBACK TO SAVEPOINT — откатиться к точке сохранения

ROLLBACK TO SAVEPOINT имя_точки_сохранения

Параметры:

имя_точки_сохранения

Точка сохранения, к которой нужно откатиться.

Команда RELEASE SAVEPOINT

Команда уничтожает точку сохранения, определённую ранее в текущей транзакции.

ROLLBACK TO SAVEPOINT — освобождает ранее определённую точку сохранения

```
RELEASE [ SAVEPOINT ] имя_точки_сохранения
```

Параметры:

имя_точки_сохранения

Имя точки сохранения, подлежащей уничтожению.

Популярный уровень изоляции

На вопрос: какой уровень изоляции в вашем текущем основном проекте?

размещенном на ресурсе <https://habr.com/> в одном из постов были получены следующие ответы:

более 80%

Read committed (Чтение зафиксированных данных)

около 13%

Repeatable read (Повторяемое чтение)

около 10%

Serializable (Сериализуемость)

Вопросы



Список литературы

1. Шёниг, Г. -. PostgreSQL 11. Мастерство разработки / Г. -. Шёниг ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2020. — 352 с. — ISBN 978-5-97060-671-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131714> (дата обращения: 25.02.2022). — Режим доступа: для авториз. пользователей.

Список литературы

2. MySQL 8 для больших данных / Ш. Чаллавала, Д. Лакхатария, Ч. Мехта, К. Патель ; перевод с английского А. В. Логунова. — Москва : ДМК Пресс, 2018. — 226 с. — ISBN 978-5-97060-653-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131684> (дата обращения: 25.02.2022). — Режим доступа: для авториз. пользователей.

Список литературы

3. Джуба, С. Изучаем PostgreSQL 10 / С. Джуба, А. Волков. — Москва : ДМК Пресс, 2019. — 400 с. — ISBN 978-5-97060-643-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/116125> (дата обращения: 25.02.2022). — Режим доступа: для авториз. пользователей.

Дополнительная литература

1. Волк, В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник для вузов / В. К. Волк. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 244 с. — ISBN 978-5-8114-9368-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/193373> (дата обращения: 25.02.2022). — Режим доступа: для авториз. пользователей.

Дополнительная литература

2. Сьоре, Э. Проектирование и реализация систем управления базами данных / Э. Сьоре ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2021. — 466 с. — ISBN 978-5-97060-488-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/190718> (дата обращения: 25.02.2022). — Режим доступа: для авториз. пользователей.

Дополнительная литература

3. Брэдшоу, Ш. Mongo DB Полное руководство : руководство / Ш. Брэдшоу, Й. Брэзил, К. Ходоров ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2020. — 540 с. — ISBN 978-5-97060-792-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/179483> (дата обращения: 25.02.2022). — Режим доступа: для авториз. пользователей.

Спасибо за внимание!