

# Лекция №7 Транзакции

каф. КИБЭВС  
И.В. Горбунов

# Параллельное выполнение транзакций

Основные проблемы при параллельном выполнении транзакций:

- . Пропаавшие изменения.
- . Проблемы промежуточных данных.
- . Проблемы несогласованных данных.
- . Проблемы строк-призраков (строк-фантомов).

Правила выполнения процедуры согласованного выполнения параллельных транзакций:

- В ходе выполнения транзакции пользователь видит только согласованные данные. Пользователь не должен видеть несогласованных промежуточных данных.
- Когда в БД две транзакции выполняются параллельно, то СУБД гарантированно поддерживает принцип независимого выполнения транзакций, который гласит, что результаты выполнения транзакций будут такими же, как если бы вначале выполнялась транзакция 1, а потом транзакция 2, или наоборот, сначала транзакция 2, а потом транзакция 1.

Такая процедура называется **сериализацией транзакций**. Фактически она гарантирует, что каждый пользователь (программа), обращающаяся к базе данных, работает с ней так, как будто не существует других пользователей (программ), одновременно с ним обращающихся к тем же данным.

Для поддержки параллельной работы транзакций строится специальный план.

План (способ) выполнения набора транзакций называется сериальным, если результат совместного выполнения транзакций эквивалентен результату некоторого последовательного выполнения этих же транзакций.

Наиболее распространенным механизмом, который используется коммерческими СУБД для реализации на практике сериализации транзакций является механизм блокировок.



Самый простой вариант — это блокировка объекта на все время действия транзакции. В момент начала работы с любым объектом этот объект блокируется транзакцией, которая с ним начала работу, и он становится недоступным всем другим транзакциям до окончания транзакции, заблокировавшей («захватившей») данный объект. После окончания транзакции все заблокированные ею объекты разблокируются и становятся доступными другим транзакциям. Если транзакция обращается к заблокированному объекту, то она остается в состоянии ожидания до момента разблокировки этого объекта, после чего она может продолжать обработку данного объекта



Типы конфликтов между двумя параллельными транзакциями. Можно выделить следующие типы:

- W-W — транзакция 2 пытается изменить объект, измененный незакончившейся транзакцией 1;
- R-W — транзакция 2 пытается изменить объект, прочитанный незакончившейся транзакцией 1;
- W-R — транзакция 2 пытается читать объект, измененный незакончившейся транзакцией 1.

Режимы блокировок:

- совместный режим блокировки — нежесткая, или разделяемая, блокировка, обозначаемая как S (Shared). Этот режим обозначает разделяемый захват объекта и требуется для выполнения операции чтения объекта. Объекты, заблокированные таким образом, не изменяются в ходе выполнения транзакции и доступны другим транзакциям также, но только в режиме чтения;



Режимы блокировок:

- монопольный режим блокировки — жесткая, или эксклюзивная, блокировка, обозначаемая как X (eXclusive). Данный режим блокировки предполагает монопольный захват объекта и требуется для выполнения операций занесения, удаления и модификации. Объекты, заблокированные данным типом блокировки, фактически остаются в монопольном режиме обработки и недоступны для других транзакций до момента окончания работы данной транзакции.

## Правила совместимости захватов одного объекта разными транзакциями:

		Транзакция В		
		Разблокир ована	Нежесткая блокир овка	Жесткая блокир овка
Транзакция А	Разблокир ована	Да	Да	Да
	Нежесткая блокир овка	Да	Да	Нет
	Жесткая блокир овка	Да	Нет	Нет

Применения разных типов блокировок приводит к проблеме тупиков. Проблема тупиков возникла при рассмотрении выполнения параллельных процессов в операционных средах и также была связана с управлением разделяемыми (совместно используемыми) ресурсами.

Пример



Пусть транзакция А сначала жестко блокирует таблицу 1, а потом жестко блокирует таблицу 2. Транзакция В, наоборот, сначала жестко блокирует таблицу 2, а потом жестко блокирует таблицу 1. Если обе эти транзакции начали работу одновременно, то после выполнения операций модификации первыми объектами каждой транзакции они обе окажутся в бесконечном ожидании: транзакция А будет ждать завершения работы транзакции В и разблокировки таблицы 2, а транзакция В также безрезультатно будет ждать окончания работы транзакции А и разблокировки таблицы 1



Основой обнаружения тупиковых ситуаций является построение (или постоянное поддержание) графа ожидания транзакций.

Граф ожидания транзакций может строиться двумя способами.

Граф ожидания — это направленный граф, в вершинах которого расположены имена транзакций.

Если транзакция  $A$  ждет окончания транзакции  $B$ , то из вершины  $A$  в вершину  $B$  идет стрелка.

Дополнительно стрелки могут быть помечены именами заблокированных объектов и типом блокировки.

В лекциях профессора С. Д. Кузнецова приводится несколько иной принцип построения графа ожидания. В этом случае граф ожидания транзакций строится в виде ориентированного двудольного графа, в котором существует два типа вершин — вершины, соответствующие транзакциям, и вершины, соответствующие объектам захвата. В этом графе существует дуга, ведущая из вершины-транзакции к вершине-объекту, если для этой транзакции существует удовлетворенный захват объекта. В графе существует дуга из вершины-объекта к вершине-транзакции, если транзакция ожидает удовлетворения захвата объекта.

Для распознавания тупика здесь, так же как и в первом методе, производится построение графа ожидания транзакций и в этом графе ищутся циклы, затем они удаляются.

Для обеспечения сериализации транзакций синхронизационные захваты объектов, произведенные по инициативе транзакции, можно снимать только при ее завершении. Это требование порождает двухфазный протокол синхронизационных захватов — 2PL (two phase lock) или 2PC (two phase commit). В соответствии с этим протоколом выполнение транзакции разбивается на две фазы:

- первая фаза транзакции — накопление захватов;
- вторая фаза (фиксация или откат) — освобождение захватов.

В языке SQL введен оператор явной блокировки таблицы, который позволяет точно задать тип блокировки для всей таблицы. Синтаксис операции блокировки имеет вид:

```
LOCK TABLE имя_таблицы IN {SHARED | EXCLUSIVE} MODE
```

Имеет смысл блокировать таблицу полностью, когда выполняется операция множественной модификации одной таблицы, то есть когда в ней изменяется большое количество строк. Эта операция иногда называется пакетным обновлением.

**Спасибо за внимание!!!**