

9. Транзакции

к.т.н., доцент кафедры ИиСП

Лучинин
Захар Сергеевич

Транзакции

- Транзакция – последовательность операций с данными, выполняющаяся как единое целое.
- Если СУБД может выполнить всю группу запросов, она делает это, но если любой из запросов не может быть выполнен в результате сбоя или по какой-то другой причине, не будет выполнен ни один запрос группы.

Банковская система

- Счет клиента A = \$200
- Счет клиента B = \$0

- **Задача:**

Переместить \$200 с одного счета на другой

Операция перевода денег

- SELECT
 - Убедиться, что остаток на счете A больше \$200.
- UPDATE
 - Вычесть \$200 из остатка счета A.
- UPDATE
 - Добавить \$200 к счету B.

ACID

- Atomicity - Атомарность
- Consistency - Согласованность
- Isolation - Изоляция
- Durability - Долговечность

Атомарность

- Транзакция должна функционировать как единая неделимая единица работы таким образом, чтобы вся транзакция была либо выполнена, либо отменена.
- Когда транзакции являются атомарными, не существует такого понятия, как частично выполненная транзакция: все или ничего.

Согласованность (консистентность)

- База данных должна всегда переходить из одного непротиворечивого состояния в последующее.
- Из примера: непротиворечивость гарантирует, что сбой не приведет к исчезновению \$200 со счета. Поскольку транзакция не будет зафиксирована, ни одно из изменений в этой транзакции не будет отражено в базе данных.

Изолированность

- Результаты транзакции невидимы другим транзакциям, пока она не закончена.
- В примере: гарантирует, что если программа суммирования остатков на банковских счетах будет запущена, она по-прежнему увидит \$200 на текущем счете.

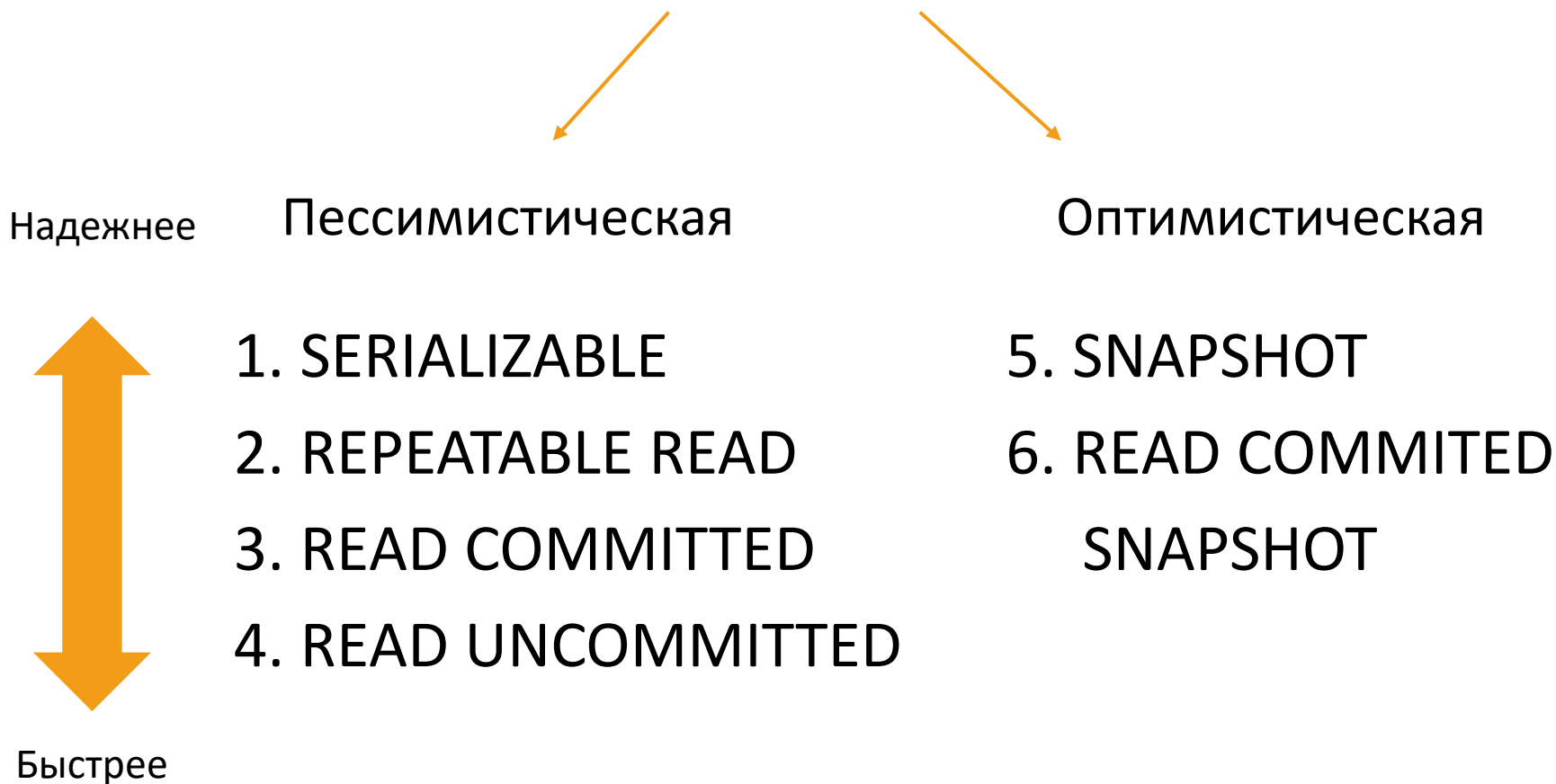
Долговечность

- Будучи зафиксированы, внесенные в ходе транзакции изменения становятся постоянными.
- Изменения должны быть записаны так, чтобы данные не могли быть потеряны в случае сбоя системы.

Уровни изоляции

- Набор правил, устанавливающих, какие изменения видны внутри и вне транзакции, а какие нет.
- Стандартизировано, но разные СУБД всё равно могут работать по разному или поддерживать не все уровни.
- **SET TRANSACTION ISOLATION LEVEL <название>;**

Стратегии изоляции



READ UNCOMMITTED

- Пользователи могут считывать строки, которые были изменены другими транзакциями, но еще не были зафиксированы.
- Данный уровень изоляции называется
“Грязное чтение” - Dirty read

```
SET TRANSACTION ISOLATION LEVEL  
  READ UNCOMMITTED
```

READ COMMITTED

- Транзакция увидит только те изменения, которые были уже зафиксированы другими транзакциями к моменту ее начала
- Изменения останутся невидимыми для других транзакций, пока текущая транзакция не будет зафиксирована
- Уровень изоляции по умолчанию в большинстве СУБД
- Возможно *невоспроизводимое чтение*

Невоспроизводимое чтение

Транзакция 1	Транзакция 2
	SELECT f2 FROM tbl1 WHERE f1=1;
UPDATE tbl1 SET f2=f2+1 WHERE f1=1;	
COMMIT;	
	SELECT f2 FROM tbl1 WHERE f1=1;

Выполненная два раза одна и та же команда возвращает разный результат.

REPEATABLE READ

- Любые строки, которые считываются в контексте транзакции будут «выглядеть такими же» при последовательных операциях чтения в пределах одной и той же транзакции

Phantom reads

- В транзакции 1 выбрали диапазон строк
- Транзакция 2 вставляет новую строку в этот диапазон
- Транзакция 1 снова выбирает этот диапазон строк

Транзакция 1	Транзакция 2
SELECT SUM(f2) FROM tbl1;	
	INSERT INTO tbl1 (f1,f2) VALUES (15,20);
	COMMIT;
SELECT SUM(f2) FROM tbl1;	

SERIALIZABLE

- Полная блокировка строк/таблиц, которые использует транзакция
- Одновременно с одними и теми же данными может работать только одна транзакция

Журнал транзакций

- Упреждающая запись в журнал:
 - Вместо обновления таблиц на диске каждый раз, когда происходит какое-либо изменение, подсистема хранения данных может изменить находящуюся в памяти копию данных.
 - Затем подсистема хранения запишет сведения об изменениях в журнал транзакции, который хранится на диске и потому долговечен (энерго-независим).
 - Впоследствии в какой-то более поздний момент времени процесс обновит таблицу на диске

Журнал транзакций

В простейшем случае журнализация изменений заключается в последовательной записи всех изменений, выполняемых в базе данных.

Записывается следующая информация:

- порядковый номер, тип и время изменения;
- идентификатор транзакции;
- объект, подвергшийся изменению (номер хранимого файла и номер блока данных в нём, номер строки внутри блока);
- предыдущее состояние объекта и новое состояние объекта.

Журнал содержит отметки начала и завершения транзакции, и отметки принятия контрольной точки.

Журнал транзакций

Общий алгоритм:

- изменения пишутся в журнал транзакций (состояние до и после) и изменяются страницы БД в памяти;
- в журнал транзакций сбрасывается маркер успешного завершения транзакции;
- журнал транзакций сбрасывается на диск;
- данные из журнала транзакций применяются к таблицам БД.

! Минимальное время транзакции не меньше времени сброса данных на диск.

Взаимоблокировки (deadlock)

Две или более транзакции запрашивают блокировку одних и тех же ресурсов, в результате чего образуется циклическая зависимость

-- Транзакция #1

BEGIN TRANSACTION

UPDATE account SET balance = 45 WHERE id = 1;

UPDATE account SET balance = 19 WHERE id = 2;

COMMIT;

-- Транзакция #2

BEGIN TRANSACTION

UPDATE account SET balance = 20 WHERE id = 2;

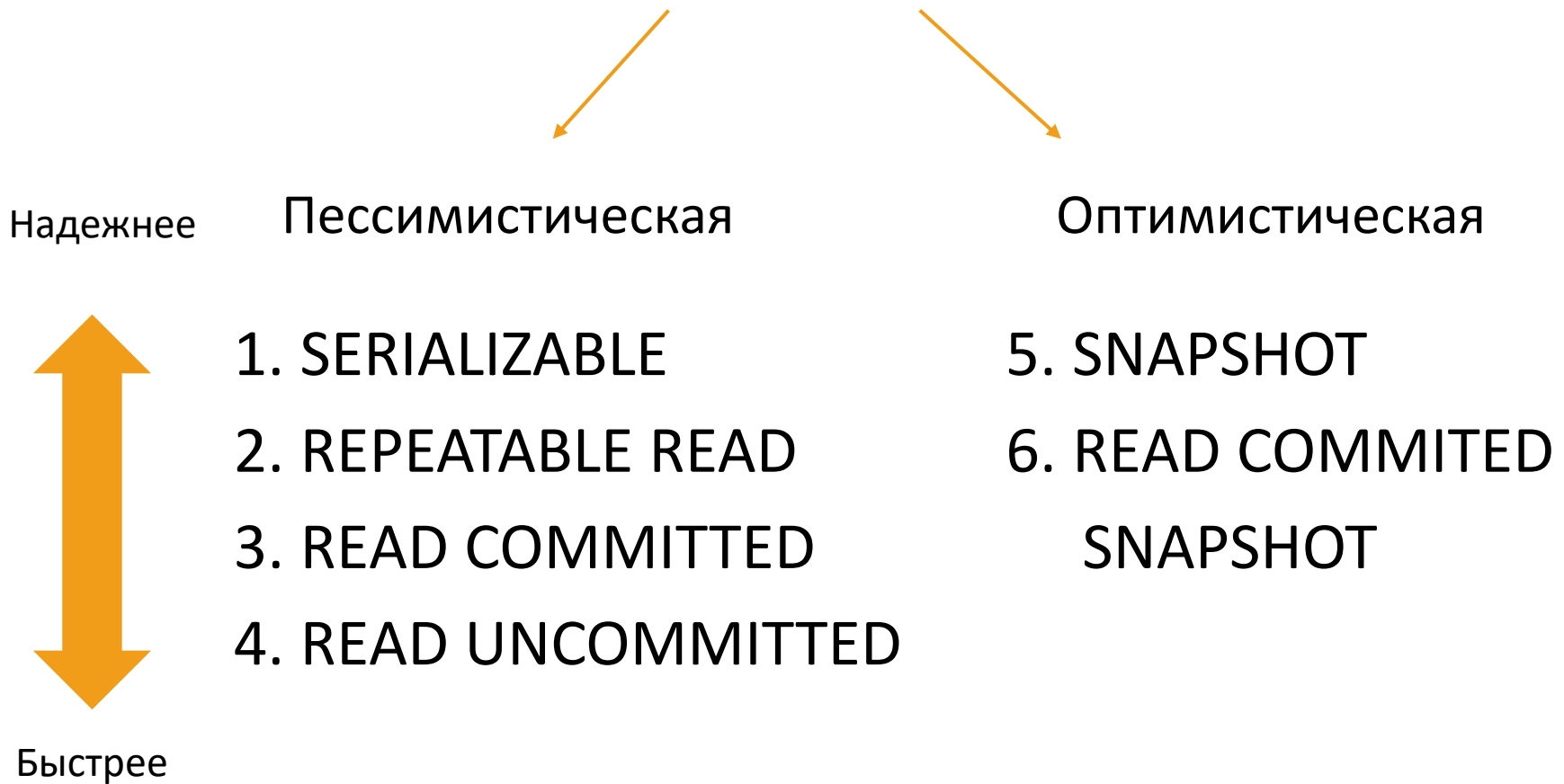
UPDATE account SET balance = 47 WHERE id = 1;

COMMIT;

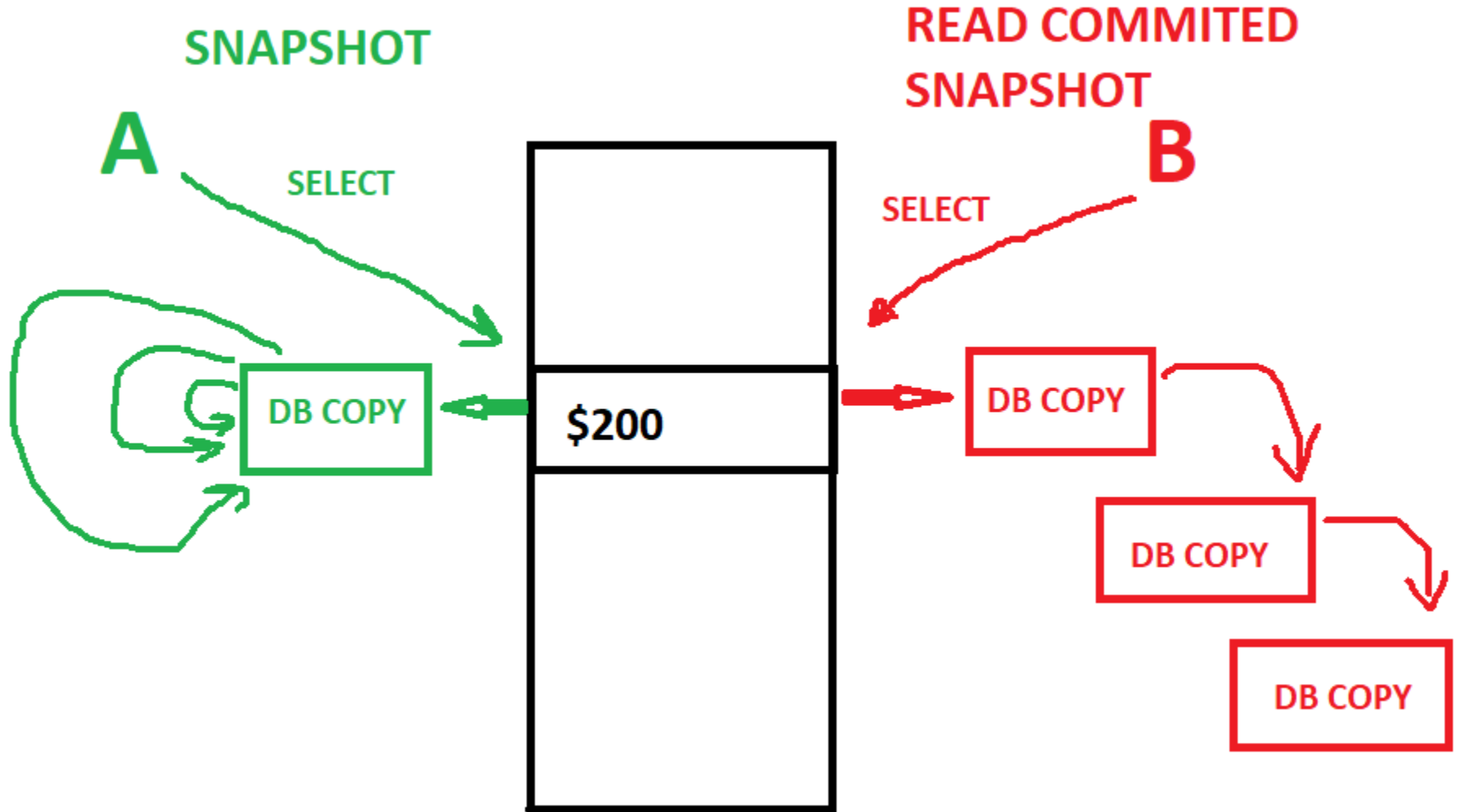
Борьба с блокировками

- 1. Укорачивать транзакции
- 2. Понижать режим изоляции
- 3. Запускать изоляции в одном направлении
- 4. SET DEADLOCK PRIORITY

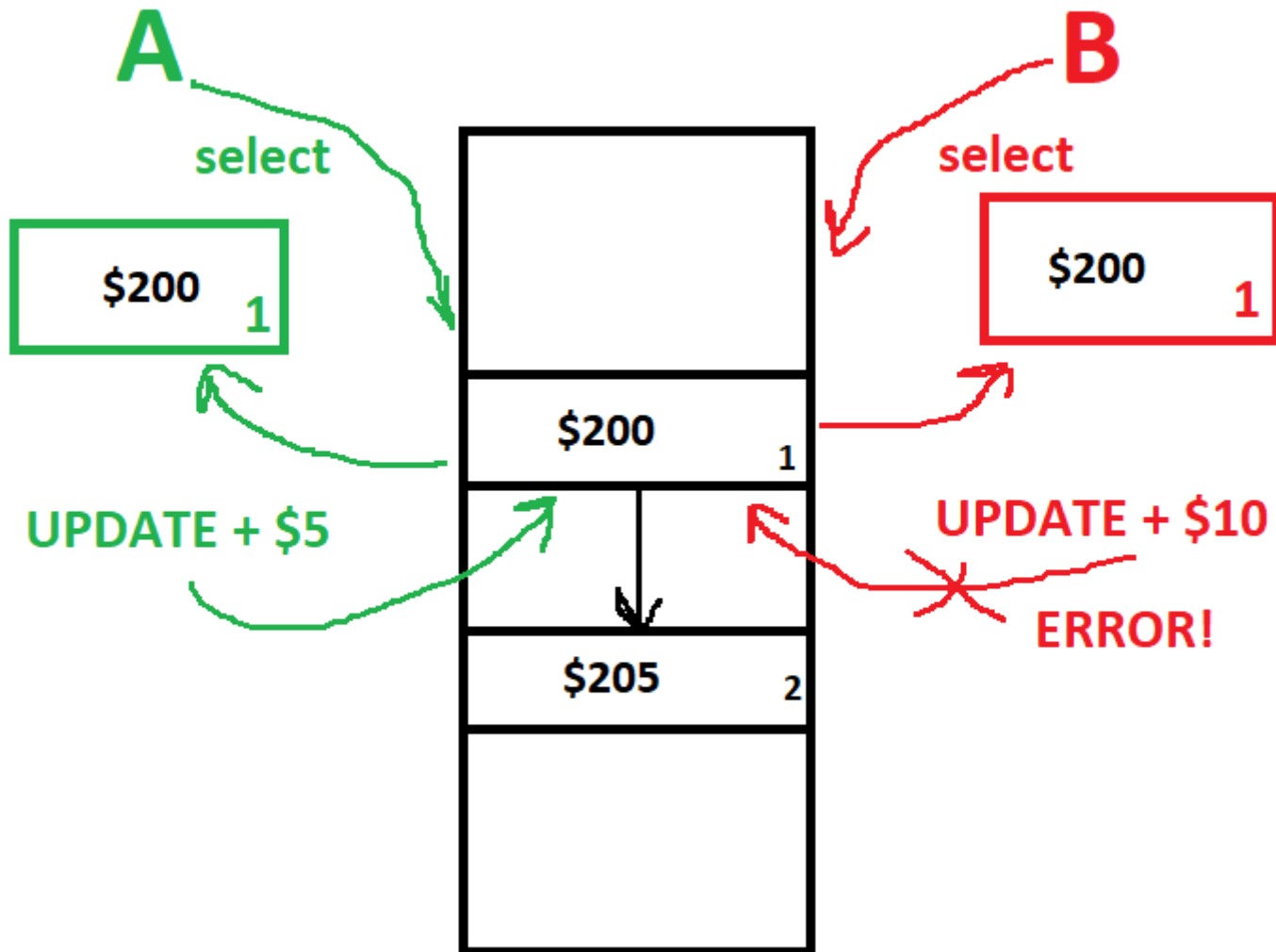
Стратегии изоляции



SNAHSHOT & READ COMMITTED S.



SNAPSHOT UPDATE CONFLICT



Multiversion Concurrency Control (MVCC)

Управление параллельным доступом с помощью многоверсионности — механизм обеспечения параллельного доступа к БД.

Заключается в предоставлении каждому пользователю так называемого «снимка» БД, обладающего тем свойством, что вносимые пользователем изменения в БД невидимы другим пользователям до момента фиксации транзакции.

MVCC

- С каждой строкой хранится два значения:
 - Время создания
 - Время, когда истек срок ее хранения (или она была удалена)
- Вместо записи реальных значений момента времени, когда произошли указанные события, строка хранит системный номер версии для этого момента.
- Данное число увеличивается на единицу в начале каждой транзакции
- Все запросы сравнивают версию строки с версией транзакции