

# Базы данных (часть 4)

Киселев Денис Викторович

# Содержание

1. *Транзакции*
2. *Блокировки*

# Транзакции

Транзакция – набор операций (изменений), который должен быть выполнен полностью или не выполнен совсем (единый логический блок).

Транзакции параллельны, если их выполнение пересекается во времени.

Механизм транзакций –  
основа обеспечения целостности БД.

Свойства транзакции (**ACID**):

- Atomicity – атомарность,
- Consistency – согласованность,
- Isolation – изолированность,
- Durability – устойчивость.

# Транзакция (пример)

Счета	
№ счета	Сумма
...	
111	1 000
112	2 500
...	

+200

-200



Счета	
№ счета	Сумма
...	
111	1 200
112	2 300
...	

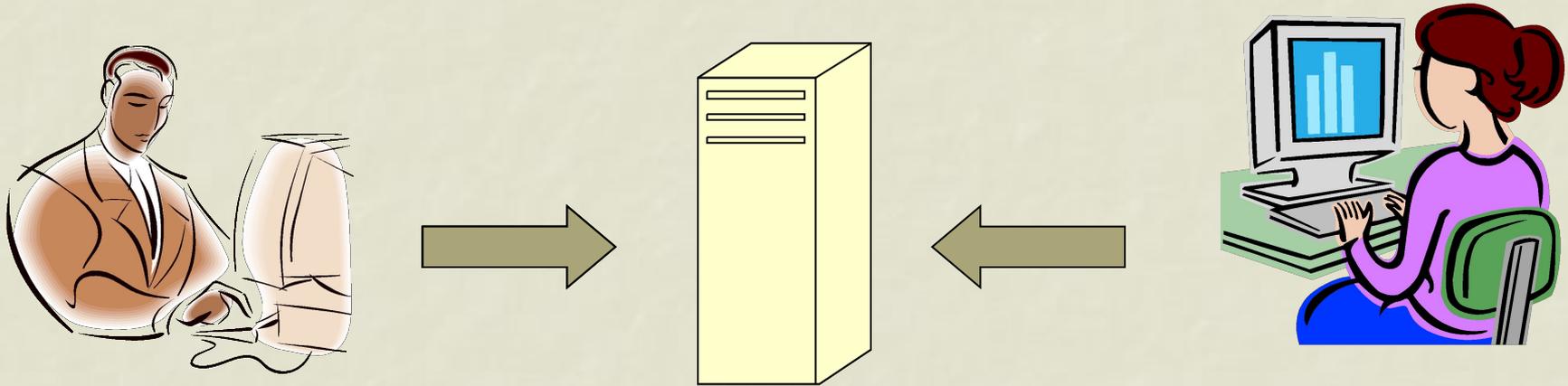
```
UPDATE Счета SET Сумма = Сумма + 200 WHERE № счета = 111;
```

```
UPDATE Счета SET Сумма = Сумма - 200 WHERE № счета = 112;
```

# Проблемы при выполнении транзакций

- потерянное обновление (*lost update*) — при одновременном изменении одного блока данных разными транзакциями, одно из изменений теряется;
- «грязное» чтение (*dirty read*) — чтение данных, добавленных или изменённых транзакцией, которая впоследствии не подтвердится (откатится);
- неповторяющееся чтение (*non-repeatable read*) — при повторном чтении в рамках одной транзакции, ранее прочитанные данные оказываются изменёнными или удалёнными;
- фантомное чтение (*phantom reads*) — при повторном чтении в рамках одной транзакции прочитаны данные (новые "фантомные" строки), которых при предыдущих чтениях не было, хотя они удовлетворяли условиям отбора предыдущего чтения.

# Проблемы параллельной обработки (Lost Update)



```
SELECT Money FROM Bank  
  WHERE Account = 565488;  
1 000
```

```
INSERT INTO Pays VALUES (... ,800);  
UPDATE Bank SET Money = 200  
  WHERE Account = 565488;
```

```
SELECT Money FROM Bank  
  WHERE Account = 565488;  
1 000
```

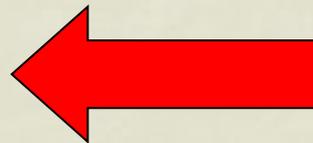
```
INSERT INTO Pays VALUES (... ,500);  
UPDATE Bank SET Money = 500  
  WHERE Account = 565488;
```

# Проблемы параллельной обработки (Non-repeatable Read)

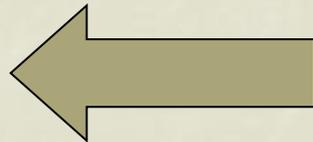
Список группы  
1...  
2...  
...  
...  
20...  
21...  
  
Итого: 20



```
SELECT * FROM Students  
WHERE Группа = 'xxx'
```



```
DELETE FROM Students  
WHERE Номер = '111111'
```



```
SELECT Count(*) FROM Students  
WHERE Группа = 'xxx'
```

# Транзакции (операторы)

Транзакции:

- Явные;
- Неявные.

```
BEGIN TRANSACTION
INSERT...
UPDATE...
...
COMMIT / ROLLBACK
```

```
BEGIN TRANSACTION
INSERT...
SAVE TRANSACTION T1
UPDATE...
...
ROLLBACK T1
```

```
BEGIN TRANSACTION
```

```
UPDATE Счета SET Сумма = Сумма + 200 WHERE № счета = 111;
```

```
UPDATE Счета SET Сумма = Сумма - 200 WHERE № счета = 112;
```

```
COMMIT
```

# Уровни изоляции транзакций

- **Serializable** – нельзя обращаться к данным, обрабатываемым другой транзакцией.
- **Repeatable Read** – нельзя обращаться к обновленным или удаленным данным, но можно к добавленным.
- **Read Committed** – можно обращаться к зафиксированным данным.
- **Read Uncommitted** – можно обращаться к любым обновленным и не зафиксированным данным.
- **Snapshot** – каждая транзакция работает со своей версией данных.

**SET TRANSACTION ISOLATION LEVEL**

**{ READ UNCOMMITTED | READ COMMITTED |  
REPEATABLE READ | SNAPSHOT | SERIALIZABLE }**

# Уровни изоляции транзакций и ошибки целостности

	<b>Lost update</b>	<b>Dirty read</b>	<b>Non-repeatable read</b>	<b>Phantom read</b>
Serializable	Green	Green	Green	Green
Repeatable Read	Green	Green	Green	Red
Read Committed	Green	Green	Red	Red
Read Uncommitted	Green	Red	Red	Red
Snapshot	Green	Green	Green	Green

# Блокировки

**Блокировка** – отметка о захвате объекта транзакцией в ограниченный или исключительный доступ с целью предотвращения коллизий и поддержания целостности данных.

## **По области действия:**

- Строчная – блокировка одной строки в таблице без ограничений действий над другими строками.
- Гранулярная – блокировка всей таблицы, всей страницы или всех строк.
- Предикатная – блокировка группы строк, ограниченных предикатами.

# Виды блокировок

## По реализации:

- Пессимистическая блокировка накладывается перед модификацией данных на все строки, участвующие в модификации. Конфликты распознаются и разрешаются немедленно.
- Оптимистическая блокировка сохраняет результаты модификации в рабочей памяти транзакции. Конфликты возникают на стадии фиксации транзакций.

## По строгости:

- **Shared (совмещаемая)**
- **Exclusive (монопольная)**
- Update (обновления)
- Intent (намерения)
- Schema (схемы)
- Bulk Update (массового обновления)
- Key-Range (диапазона ключей)

# Блокировки (MS SQL Server)

<b>Shared (S)</b>	Операции чтения (SELECT). Не совместима с X-блокировками. При Repeatable Read и выше удерживается все время выполнения транзакции.
<b>Exclusive (X)</b>	Модификация данных. Не совместима с другими блокировками.
<b>Update (U)</b>	Ожидается повышение от S-блокировки до X-блокировки. Может получить только одна транзакция.
Intent (IS, IX, SIX)	Повышение производительности и эффективности. Блокировка на объект высокого уровня (таблицу) перед установкой X/S-блокировки на объект низкого уровня (страницу).
Schema (Sch-M, Sch-S)	Блокирует все операции при модификации схемы (Sch-M) и компиляции или выполнении запросов (Sch-S)
Bulk Update (BU)	Массовая вставка. Запрещает доступ к таблицам другим процессам.
Key-range	Защита диапазона строк от фантомных вставок и удалений. Только на уровне SERIALIZABLE.

# Совместимость основных видов блокировок

	Ранее предоставленный режим		
Запрошенный режим	S	U	X
Коллективная (S)	Да	Да	Нет
Блокировка обновления (U)	Да	Нет	Нет
Монопольная (X)	Нет	Нет	Нет

# Взаимоблокировка

Транзакция 1	Транзакция 2
X-блокировка Строки 1	X-блокировка Строки 2
S-блокировка Строки 2	S-блокировка Строки 1

Транзакция 1	Транзакция 2
Обновление таблицы 1	Обновление таблицы 2
Обновление таблицы 2	Обновление таблицы 1

«Жертва» (blocking victim) –  
процесс с минимальным суммарным временем работы