



SQL

Базовый уровень



Транзакции и блокировки

Транзакции и блокировки

- Транзакции и блокировки - механизм регулирования доступа к БД из приложений.
- Транзакция в SQL есть логическая последовательность операций DML по внесению изменений в БД, принимаемая или же отвергаемая СУБД в конечном итоге ("по завершению транзакции") в целом.

Транзакции и блокировки

Транзакции позволяют:

- восстанавливать данные при аварийном прекращении сеанса связи программы с СУБД ("единица восстановления");
- гарантировать целостное представление данных программе при одновременной работе с данными нескольких программ;
- гарантировать целостное хранение данных в БД при одновременной работе с ними нескольких программ.

Транзакции и блокировки

| Account | Value | T1 |
|--------------|------------|------------|
| A | 100 | -100 |
| B | 200 | |
| C | 300 | |
| D | 400 | +100 |
| Total | 900 | 900 |

Total ?

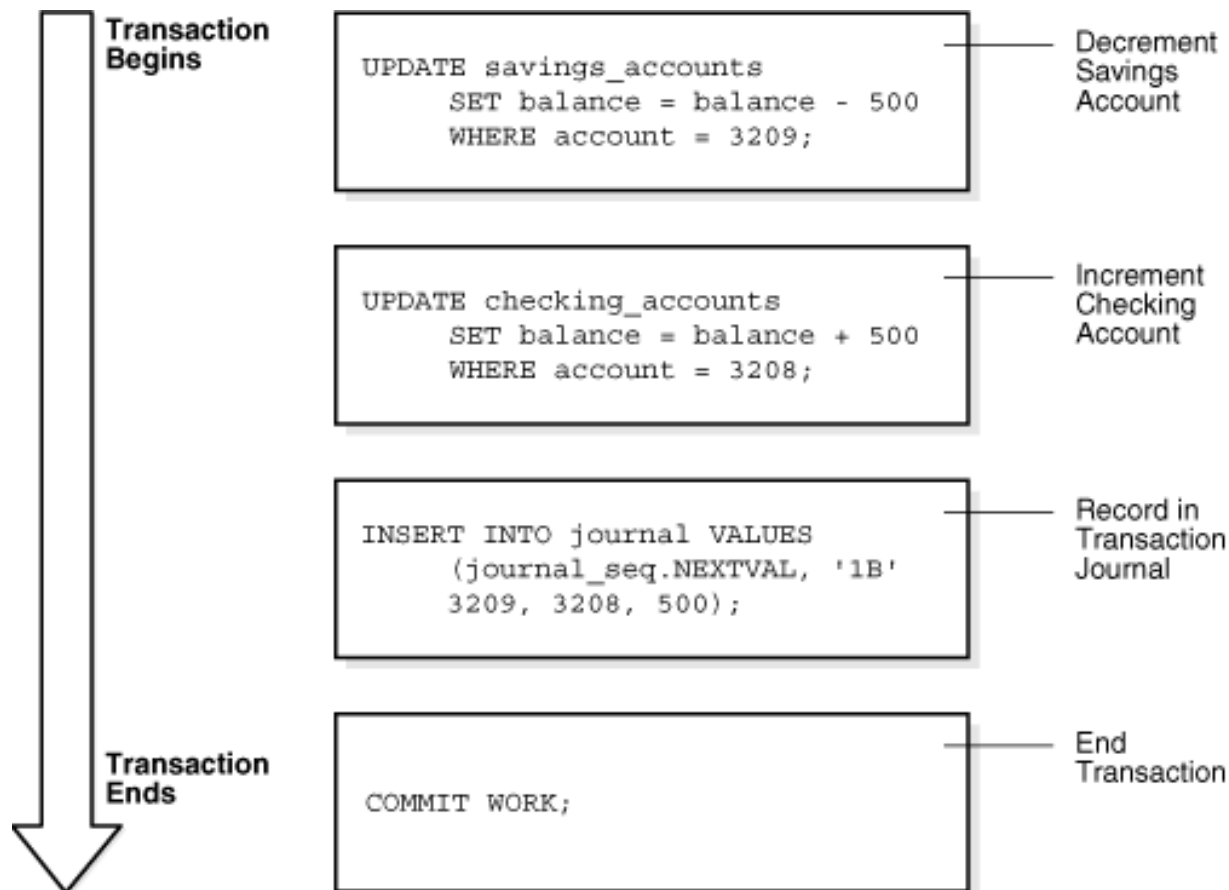


Свойства ACID

Атомарность (Atomicity).

- Это свойство означает, что результаты всех операций, успешно выполненных в пределах транзакции, должны быть отражены в состоянии БД, либо в состоянии базы данных не должно быть отражено действие ни одной операции.

Свойства ACID



Свойства ACID

Согласованность (Consistency).

- Транзакция может быть успешно завершена с фиксацией результатов своих операций только в том случае, когда действия операций не нарушают целостность базы данных, т. е. удовлетворяют набору ограничений целостности, определённых для этой базы данных.

Свойства ACID

- **Согласованность.** Ограничения целостности:
 - NOT NULL – не может быть пустым
 - UNIQUE – уникальное значение
 - CHECK – определяет множество допустимых значений
 - PRIMARY KEY - ограничения первичного ключа (не NULL, уникальный)
 - FOREIGN KEY

Свойства ACID

Изоляция (Isolation).

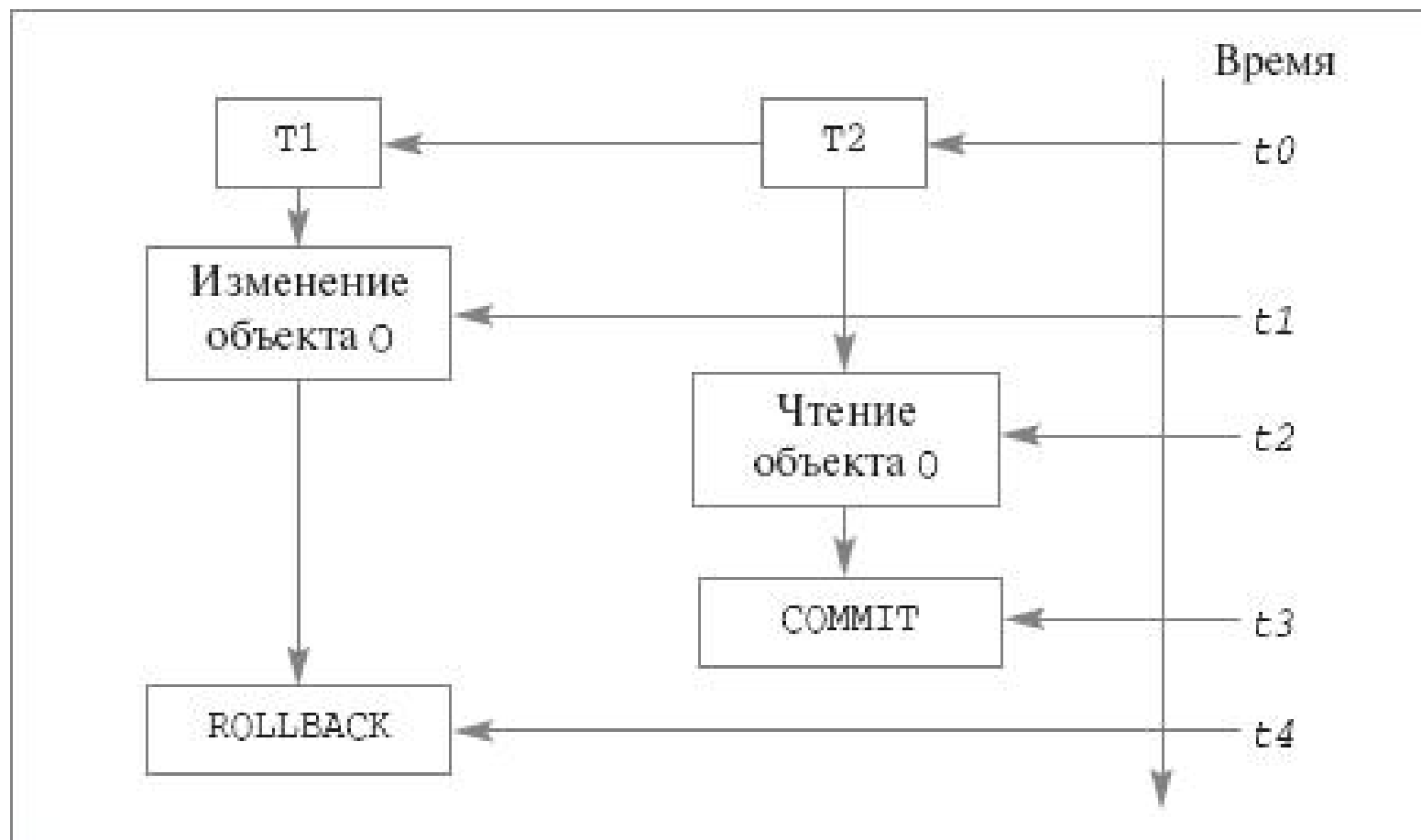
- Требуется, чтобы две одновременно выполняемые транзакции никоим образом не действовали одна на другую. Другими словами, результаты выполнения операций транзакции T1 не должны быть видны никакой другой транзакции T2 до тех пор, пока транзакция T1 не завершится успешным образом.

Свойства ACID

Долговечность (Durability).

- После успешного завершения транзакции все изменения, которые были внесены в состояние базы данных операциями этой транзакции, должны гарантированно сохраняться, даже в случае сбоев аппаратуры или программного обеспечения.

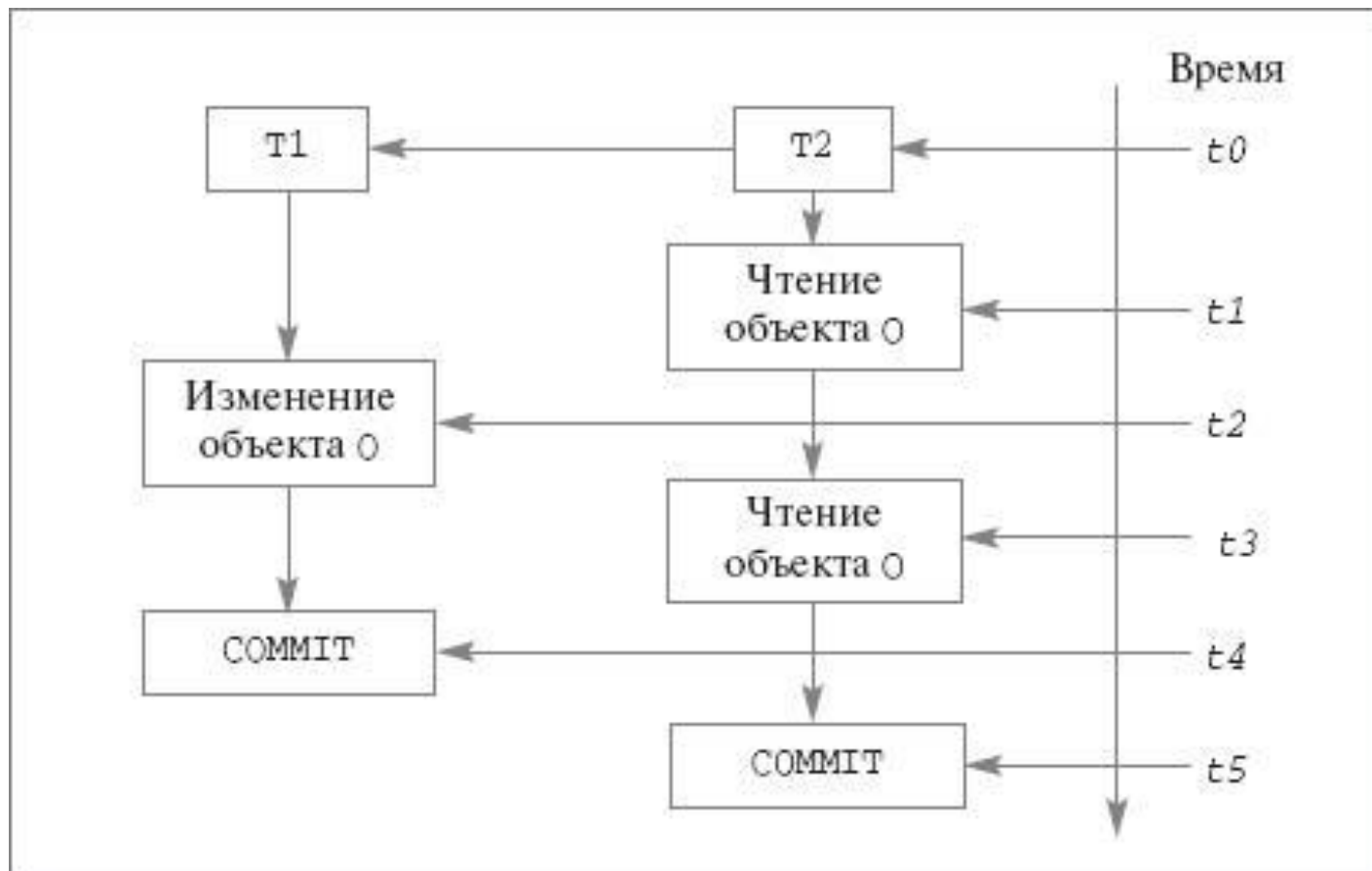
Грязное чтение



Грязное чтение

- В SQL феномен "грязного" чтения может наблюдаться у транзакций, выполняемых на уровне изоляции READ UNCOMMITTED.
- Рекомендуется использовать этот уровень изоляции только в тех транзакциях, для выполнения функций которых точные данные не обязательны.

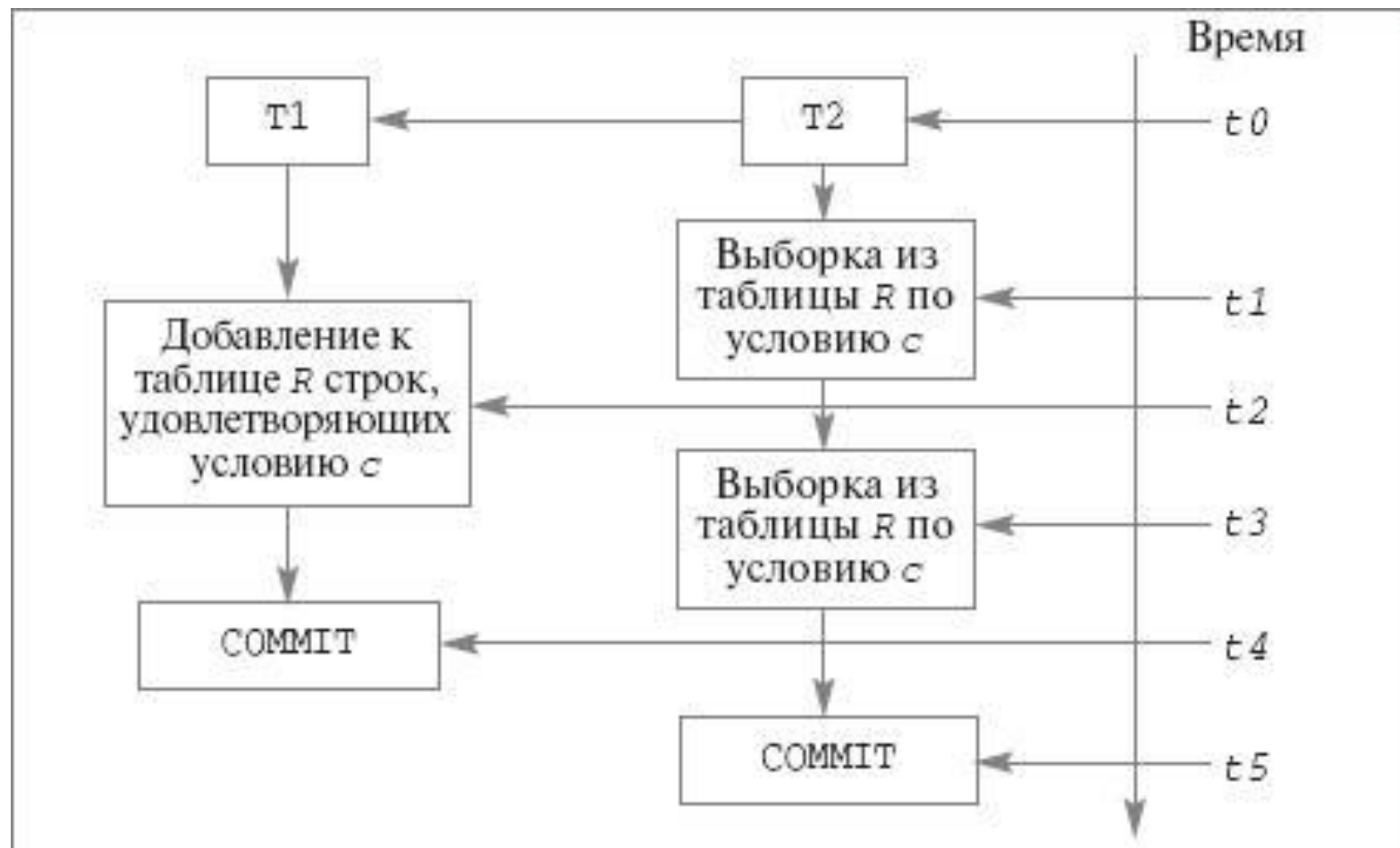
Неповторяемое чтение



Неповторяемое чтение

- В SQL феномен неповторяемого чтения может наблюдаться у транзакций, выполняемых на уровне изоляции READ COMMITTED
- Этот уровень изоляции, гарантирует отсутствие феномена "грязного" чтения .

ФАНТОМЫ



ФАНТОМЫ

- В SQL феномен фантомов может наблюдаться у транзакций, выполняемых на уровне изоляции REPEATABLE READ
- Этот уровень изоляции гарантирует отсутствие феномена неповторяемого чтения .

SERIALIZABLE

- Для транзакций, выполняемых на уровне изоляции `SERIALIZABLE`, невозможно и проявление феномена фантомов.
- Термин `serializable` (сериализуемый) используется по той причине, что при работе на данном уровне изоляции суммарный эффект выполнения набора транзакций $\{T_1, T_2, \dots, T_n\}$ идентичен эффекту некоторого последовательного выполнения этих транзакций. Это означает предельную изолированность транзакций.

УРОВНИ ИЗОЛЯЦИИ ANSI

| Isolation level | Dirty reads | Non-repeatable reads | Phantoms |
|------------------|-------------|----------------------|-----------|
| Read Uncommitted | may occur | may occur | may occur |
| Read Committed | - | may occur | may occur |
| Repeatable Read | - | - | may occur |
| Serializable | - | - | - |

Уровни изоляции MS SQL

- Установка уровня изоляции:

```
SET TRANSACTION ISOLATION LEVEL {  
    READ COMMITTED  
    READ UNCOMMITTED  
    REPEATABLE READ  
    SERIALIZABLE },
```

Например:

```
set transaction isolation level serializable;
```

Виды транзакций MS SQL

Виды транзакций

- SQL Server поддерживает четыре вида определения транзакций:
 - Автоматические (automatic);
 - Неявные (implicit);
 - Пользовательские (user-defined);
 - Распределённые (distributed) – в случае использования нескольких серверов;

Автоматические (automatic)

- По умолчанию SQL Server работает в режиме автоматического начала транзакций, когда каждая команда рассматривается как отдельная транзакция.
- Если команда выполнена успешно, то её изменения фиксируются.
- Если при выполнении команды произошла ошибка, то сделанные изменения отменяются и система возвращается в первоначальное состояние.
- Устанавливаются командой
SET IMPLICIT_TRANSACTIONS **OFF**

Неявные (implicit)

- Инициируются автоматически, при запуске любой команды DDL или DML.
- Продолжаются до тех пор, пока явно не завершаются пользователем.
- Устанавливаются при помощи команды

SET IMPLICIT_TRANSACTIONS **ON**

Пользовательские (user-defined)

- Требуют, чтобы пользователь указал начало и конец транзакции, используя следующие команды:

BEGIN TRAN[SACTION]

- Явное начало транзакции

COMMIT [TRAN[SACTION]

- Успешное завершение

ROLLBACK [TRAN[SACTION]

- Откат изменений

SAVE TRAN[SACTION]

- Сохранение промежуточного состояния

Пример

--ПОЛЬЗОВАТЕЛЬСКАЯ транзакция

begin tran

--удаление записей

delete

from HumanResources.MyEmployee;

go

--данных нет!

select * from HumanResources.MyEmployee;

--откат изменений

rollback;

--данные есть

select *

from HumanResources.MyEmployee

Явное начало

Явное окончание

Точки сохранения

-- Использование точек сохранения

```
begin tran --начало транзакции
select * from HumanResources.MyEmployee; -- начальное состояние
SAVE TRANSACTION point1 -- точка сохранения
```

--удаление записи

```
delete from HumanResources.MyEmployee where FirstName like '%Ken%';
select * from HumanResources.MyEmployee;
SAVE TRANSACTION point2 --в этой точке сохраняется таблица без сотрудника Ken
```

```
delete from HumanResources.MyEmployee where FirstName like 'Ter%';
select * from HumanResources.MyEmployee;
SAVE TRANSACTION point3 --в этой точке сохраняется таблица без сотрудников Ken и Terri
```

```
delete from HumanResources.MyEmployee where FirstName not like 'Rob%';
select * from HumanResources.MyEmployee;
ROLLBACK TRANSACTION point3; -- возвращаемся к состоянию point3
select * from HumanResources.MyEmployee;
ROLLBACK TRANSACTION point1; -- Первоначальное состояние таблицы.
select * from HumanResources.MyEmployee;
COMMIT; -- явное окончание транзакции
```

Вложенные транзакции

begin tran

.....

begin tran

.....

commit (rollback)

.....

commit (rollback)

- Подтверждение вложенной транзакции ни на что не влияет.
- Откат вложенной транзакции откатывает самую внешнюю транзакцию.

Взаимоблокировки

- Возможна ситуация, когда две транзакции блокируют друг друга так, что продолжение работы не возможно без принудительного завершения одной из транзакций.