Курс «Базы данных»

Тема. Транзакции. Часть 1

Барабанщиков
Игорь Витальевич

План лекции

1. Определение транзакции.
2. Модели транзакций.
3. Команды управления транзакциями.
4. Журнал транзакций.

Изменение БД

- Изменения БД обусловлены событиями во внешнем мире (перевод денег со счета на счет).
- При этом **одно внешнее событие обычно приводит к нескольким изменениям в БД**.
- Чтобы не нарушить целостность БД надо **все изменения выполнить как одно целое**.
- Поэтому изменения БД, вызванные одним событием, надо вносить по принципу «**либо все, либо ничего**».
- SQL обеспечивает такое поведение с

Пример. Банковская транзакция

Transaction Begins

```
UPDATE savings_accounts
     SET balance = balance - 500
     WHERE account = 3209;
```

Decrement Savings Account

```
UPDATE checking_accounts
     SET balance = balance + 500
     WHERE account = 3208;
```

Increment Checking Account

```
INSERT INTO journal VALUES
     (journal_seq.NEXTVAL, '1B'
     3209, 3208, 500);
```

Record in Transaction Journal

```
COMMIT WORK;
```

End Transaction

Transaction Ends

Что такое транзакция?

- **Транзакция** – это несколько последовательных инструкций SQL, которые **вместе образуют логическую единицу работы**.
- Инструкции, входящие в транзакцию, обычно **тесно связаны** между собой и **выполняют взаимосвязанные действия**.
- Каждая инструкция решает часть общей задачи, но для того, чтобы задачу можно было считать решенной, **требуется выполнить <u>все</u> эти инструкции**.

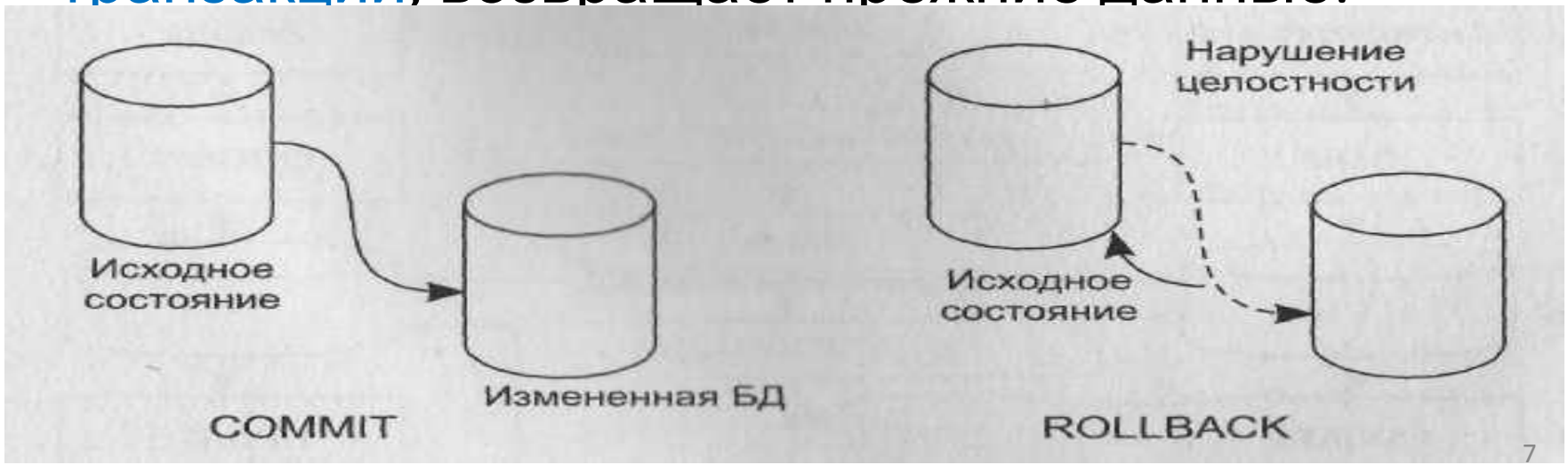
Определение транзакции

- **Транзакция** – это последовательность команд SQL, которые воспринимаются БД как одно целое.
- Либо *ВСЕ* команды транзакции <u>успешно выполнятся</u>, либо действие *ВСЕХ* команд <u>полностью отменяется</u>.
- Транзакция переводит БД из одного целостного состояния в другое

Команды управления транзакциями

Для управления транзакциями в языке SQL используются команды:

- **COMMIT** – фиксирует в БД изменения, сделанные в транзакции. Изменения становятся постоянными.
- **ROLLBACK** – отменяет изменения, сделанные в транзакции, возвращает прежние данные.



Свойства транзакции

Любая транзакция должна обладать свойствами:

- **Атомарность** (**A**tomicity) — все входящие в транзакцию операции выполняются нераздельно, т.е. будут выполнены либо все операции, либо не выполнено ни одной.
- **Согласованность** (**C**onsistency) — транзакция, фиксирующая результаты, должна сохранять согласованность данных в базе.
- **Изоляция** (**I**solation) — во время выполнения транзакции параллельные транзакции не должны оказывать влияние на её результат. Другие процессы не должны видеть данные в промежуточном состоянии.
- **Долговечность** (**D**urability) — как только транзакция зафиксирована, она остается постоянной. Никакое внешнее событие не должно привести к потере изменений.

Модель транзакций ANSI/ISO (SQL3)

Инструкция	Описание
START TRANSACTION	Устанавливает свойства новой транзакции и запускает транзакцию.
SET TRANSACTION	Устанавливает свойства очередной транзакции. Не влияет на текущую выполняемую транзакцию.
SET CONSTRAINTS	Устанавливает режим ограничений в текущей транзакции. Он определяет применяется ли ограничение немедленно или откладывается до более позднего времени.
SAVEPOINT	Создает точку сохранения в пределах транзакции. Откат текущей транзакции м.б. выполнен не к началу, а к транзакции, а к точке сохранения.
RELEASE SAVEPOINT	Освобождает точку сохранения и все ресурсы, которые она могла занимать.
COMMIT	Завершает успешную транзакцию и сохраняет в БД все внесенные изменения.
ROLLBACK	Выполняет откат изменений либо к началу транзакии, либо к заданной точке сохранения.

Управление транзакциями в Oracle

В СУБД Oracle используется **неявный режим транзакции**.

Новая транзакция начинается **первым оператором SQL**, следующим сразу после COMMIT или ROLLBACK .

Команды управления транзакциями.

- **COMMIT**
- **ROLLBACK**
- **SAVEPOINT <имя точки сохранения>**
- **ROLLBACK TO <имя точки сохранения>**
- **SET TRANSACTION**

Журнал транзакций

Журнал транзакций — системная структура, хранящая информацию об изменениях базы данных.

Цель журнализации: обеспечение возможности восстановления согласованного состояния базы данных после любого сбоя.

Восстанавливается последнее по времени согласованное состояние базы данных.

Структура журнала

Общая структура журнала — последовательный файл, в котором фиксируется <u>каждое изменение</u> БД, которое происходит в ходе выполнения транзакции.

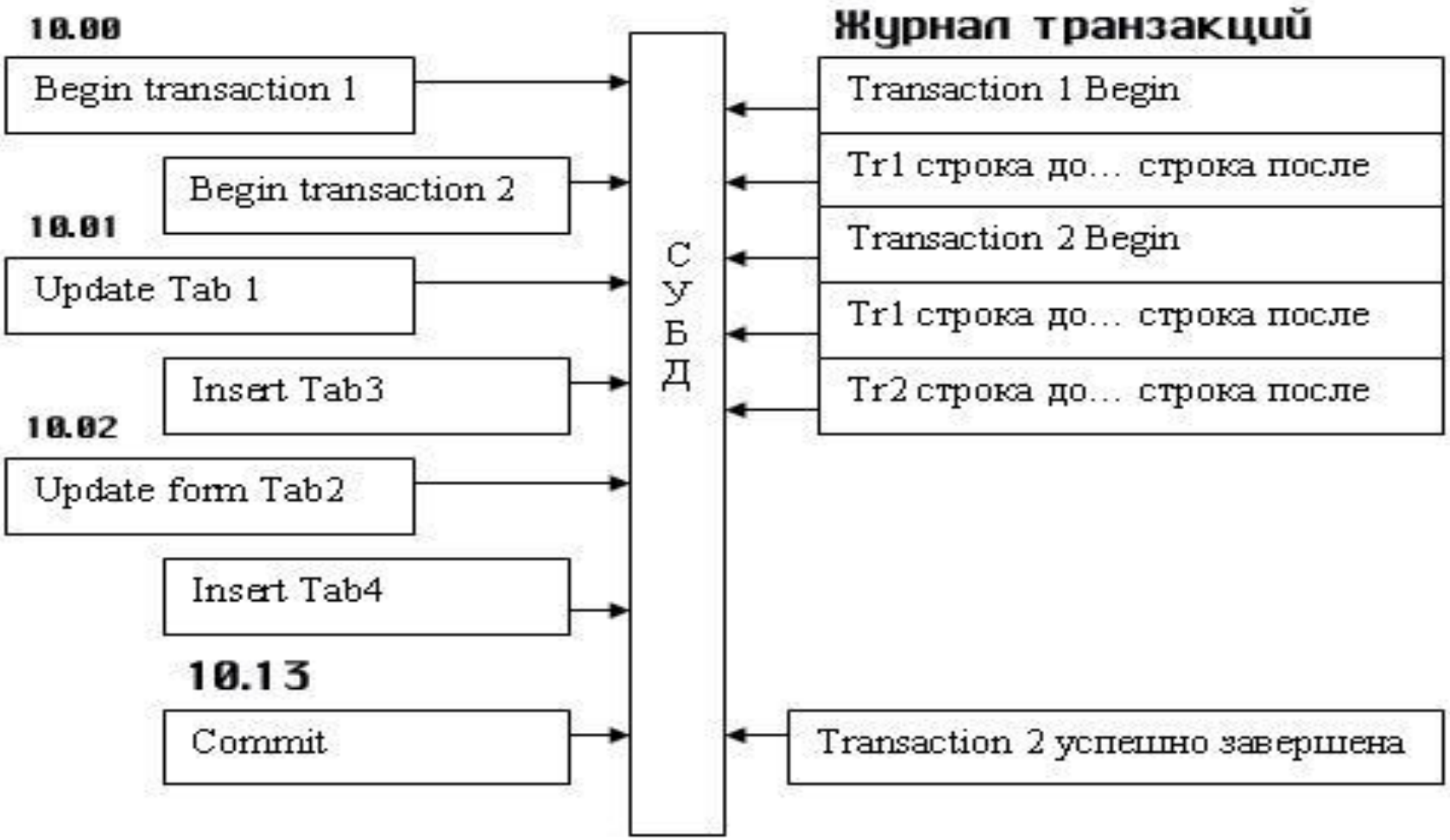
Варианты ведения журнала транзакций:

- **Протокол с отложенными обновлениями**

  <T1, ID_RECORD, атрибут, новое_значение … >

- **Протокол с немедленными обновлениями**

Журнал транзакций



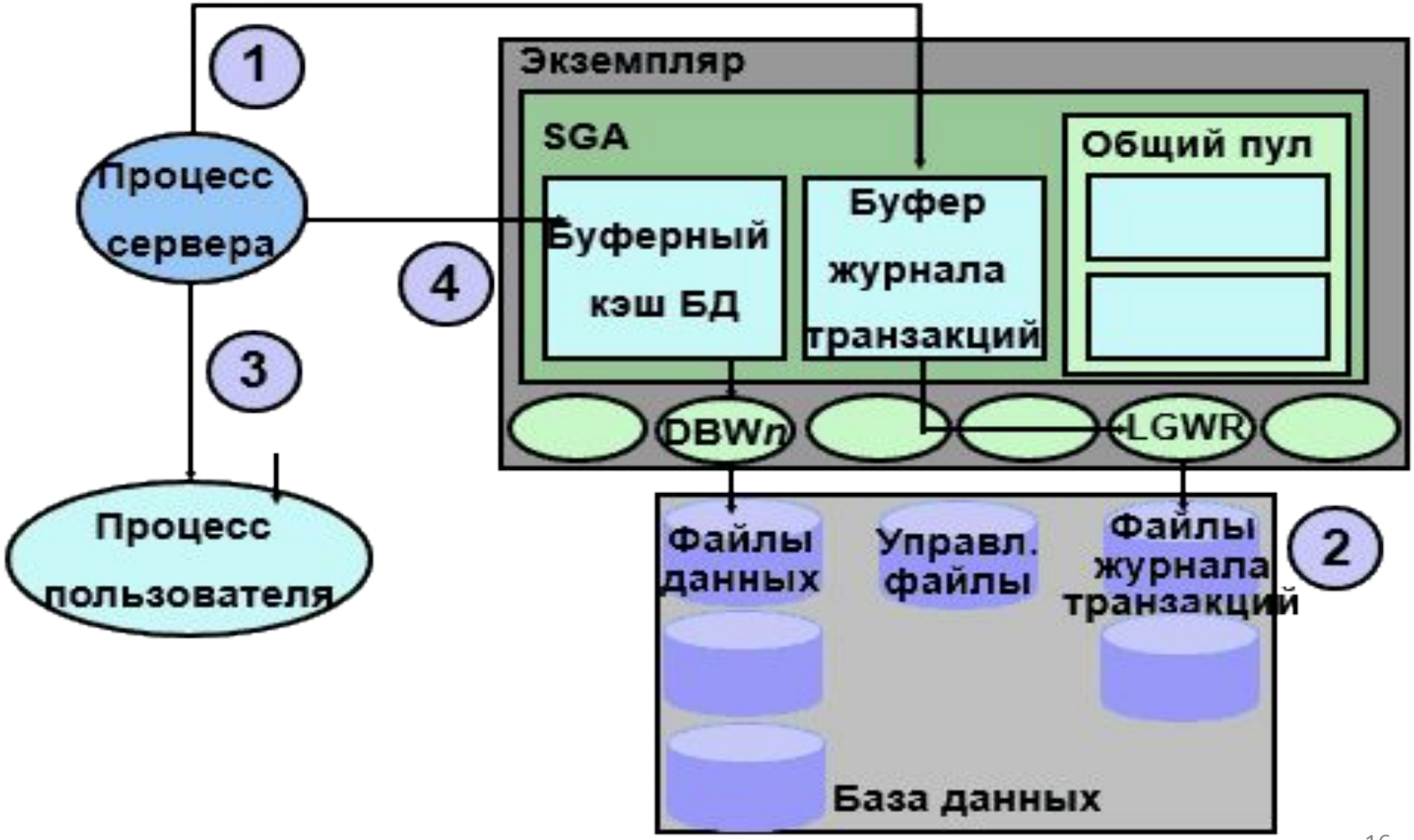
Как работает журнал

- Когда пользователь выполняет SQL-команду на изменение базы данных, **СУБД автоматически вносит в журнал транзакций одну запись для каждой строки, измененной в процессе выполнения команды**.
- Эта запись содержит две копии строки.
  - копия данных строки до изменения,
  - копия данных строки после изменения.
- **СУБД изменяет физическую строку только после того, как в журнале будет сделана соответствующая запись**.

Как работает журнал

- Если пользователь выполняет инструкцию **COMMIT**, в журнале отмечается конец транзакции.
- Если же пользователь выполняет инструкцию **ROLLBACK**, СУБД обращается к журналу и извлекает из него исходные копии строк, измененных во время транзакции.
- Используя эти копии, СУБД возвращает строки в прежнее состояние и таким образом отменяет изменения, внесенные в базу данных в ходе транзакции.

Обработка фиксации транзакции



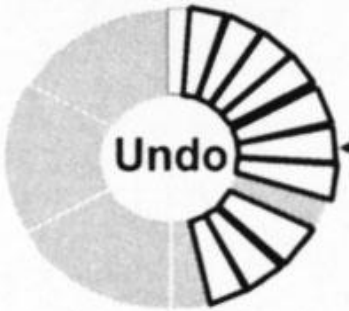
Восстановление БД

Журнал транзакций используется для восстановления БД. Возможны варианты:

- **Индивидуальный откат транзакции** *(Только для незавершившихся транзакций!)*
- **Восстановление после внезапной потери содержимого оперативной памяти**
  (**мягкий сбой**)
- **Восстановление после поломки основного внешнего носителя базы данных**

Фазы восстановления экземпляра

1. Файлы данных не синхронизированы
1. Подкат вперед (по журналу)
2. Зафиксированные и незафиксированные данные в файлах
3. Откат назад (данные undo)
4. Зафиксированные данные в файлах



Итоги

- В различных СУБД механизм транзакций реализован по-разному.
- **Для наиболее эффективного использования конкретной СУБД необходимо понимать то, как в ней реализован механизм транзакций.**
- СУБД Oracle имеет эффективный механизм транзакций, который допускает одновременную работу большого числа пользователей.