



ТРАНЗАКЦИИ

*Бежать быстро – это значит
делать медленные шаги
без перерывов между ними.*

Б. Джонсон

Содержание

2

- Понятие транзакции
- ACID-свойства транзакции
- Точки сохранения транзакции

Транзакция

3

Accounts

AcctNo	Balance	...
000374	25000	
...		
123456	10000	
...		

Перевод средств

Accounts

AcctNo	Balance	...
000374	20000	
...		
123456	15000	
...		

- Банковская транзакция состоит из двух операций:
 1. update Accounts
set Balance=Balance-5000
where AccNo='000374';
 2. update Accounts
set Balance=Balance+5000
where AccNo='123456';
- Что произойдет в случае сбоя после выполнения первого шага транзакции?

Транзакция

4

- *Транзакция* – набор операций над базой данных, которые рассматриваются как *одна неделимая* операция и выполняют перевод базы данных из одного согласованного состояния в другое.
- В ходе выполнения транзакции согласованность базы данных может временно нарушаться.
- Результатом транзакции может быть *фиксация* (*COMMIT*) или *откат* (*ROLLBACK*) **всех** входящих в нее операций.

Транзакция

5

BEGIN TRANSACTION

```
update Accounts
set Balance=Balance-5000
where AccNo='000374';
if SQLState<>'00000' goto UNDO;
update Accounts
set Balance=Balance+5000
where AccNo='123456';
if SQLState<>'00000' goto UNDO;
COMMIT;
goto Finish;
Undo: ROLLBACK;
Finish: ;
```

END TRANSACTION

Диспетчер транзакций

6

- В состав СУБД входит *диспетчер транзакций*.
- Команды диспетчера транзакций
 - ▣ *COMMIT (зафиксировать)*
 - сигнал диспетчеру транзакций об *успешном окончании* транзакции: логическая единица работы успешно завершена, база данных вновь находится в непротиворечивом состоянии
 - *все обновления транзакции необходимо внести в базу данных.*
 - ▣ *ROLLBACK (откатить)*
 - сигнал диспетчеру транзакций о *неудачном окончании* транзакции: в процессе выполнения логической единицы работы произошла ошибка, вследствие которой база данных находится в противоречивом состоянии
 - *все обновления транзакции необходимо отменить.*

Журнал транзакций

7

- Диспетчер транзакций осуществляет ведение *журнала транзакций*, в котором отмечаются операции, производимые транзакциями и результаты их завершения.
- Журнал транзакций заполняется по *принципу опережающей записи WAL (Write Ahead Log)* – данные об операции сначала записываются в журнал, а затем производится операция.
- Журнал транзакций используется для восстановления базы данных после сбоев.

АСИД-свойства транзакции

8

- *Атомарность (Atomicity)* – выполняются все операторы транзакции или ни один.
- *Согласованность (Consistency)* – перевод базы данных из одного согласованного состояния в другое.
- *Изолированность (Isolation)* – параллельные транзакции не могут повлиять друг на друга.
- *Долговечность (Durability)* – изменения, произведенные зафиксированной транзакцией, не могут быть потеряны ни при каких обстоятельствах.

Транзакции в SQL

9

- Транзакция начинается
 - ▣ *явно* оператором `START TRANSACTION`
 - ▣ *неявно* первым выполняемым оператором SQL*.
- Транзакция завершается
 - ▣ *явно*
 - *фиксируется* оператором `COMMIT`
 - *откатывается* оператором `ROLLBACK`
 - ▣ *неявно*
 - *фиксируется* оператором модификации объекта схемы (`CREATE`, `DROP`, `ALTER`) или в случае завершения сессии пользователем
 - *откатывается* в случае аварийного завершения сессии пользователя

* Исполнение триггеров, активируемых оператором транзакции, является частью той же транзакции.

Выполнение оператора vs фиксация транзакции

10

- Успешное выполнение оператора транзакции не гарантирует сохранение результатов его выполнения в случае отката всей транзакции.

SQL>

```
create table emp (id number primary key, name char(20), age number);
```

```
insert into emp values (1, 'Иванов', 40);
```

```
insert into emp values (2, 'Петрова', 30);
```

```
insert into emp values (3, 'Сидоров', 50);
```

```
rollback;
```

```
select * from emp;
```

```
ID      NAME      AGE
```

```
нет строк
```

Неуспех оператора vs откат транзакции

10

- ❑ Неуспешное выполнение одного оператора транзакции не означает откат всей транзакции.

SQL>

```
create table emp (id number primary key, name char(20), age number);
```

```
insert into emp values (1, 'Иванов', 40);
```

```
insert into emp values (2, 'Петрова', 30);
```

```
insert into emp values (2, 'Сидоров', 50);
```

Ошибка! Дубликат первичного ключа.

```
commit;
```

```
select * from emp;
```

ID	NAME	AGE
1	Иванов	40
2	Петрова	30

Точки сохранения транзакции

12

- *Точка сохранения (savepoint)* – текстовая метка внутри транзакции.
- Точки сохранения используются для разбиения длинной транзакции на небольшие части.
- *Откат до точки сохранения (rollback to savepoint)* позволяет откатить не всю транзакцию, а только изменения после указанной точки до текущей точки транзакции.
 - Откатываются только операторы транзакции, выполненные *после* точки сохранения.
 - Указанная точка сохранения остается, но все точки сохранения, установленные после указанной, теряются.

Точки сохранения транзакции

13

```
savepoint update_cur_data;  
  update emp set ...;  
  update dept set ...;  
  update job set ...;
```

```
savepoint delete_old_data;
```

```
delete from emp ...;  
delete from dept ...;  
delete from job ...;
```

```
savepoint insert_new_data;
```

```
insert into emp values (...);  
insert into dept values (...);  
insert into job values (...);
```

```
rollback to savepoint delete_old_data;
```

```
delete from job ...;
```

```
savepoint insert_new_data;
```

```
insert into emp values (...);  
insert into dept values (...);  
insert into job values (...);
```

```
commit;
```

```
savepoint update_cur_data;
```

```
  update emp set ...;  
  update dept set ...;  
  update job set ...;
```

```
savepoint delete_old_data;
```

```
delete from job ...;
```

```
savepoint insert_new_data;
```

```
insert into emp values (...);  
insert into dept values (...);  
insert into job values (...);
```

```
commit;
```

Режимы доступа транзакции к данным

14

- Транзакция может быть запущена в одном из двух режимов: *READ-WRITE* (по умолчанию) и *READ-ONLY*.
- В *режиме READ-WRITE* транзакция
 - может модифицировать объекты базы данных
 - видит изменения, вносимые в базу данных другими транзакциями, – после фиксации этих транзакций.
- В *режиме READ-ONLY* транзакция
 - не может модифицировать объекты базы данных
 - не видит изменений, вносимых в базу данных другими транзакциями.

Заключение

- Транзакция – набор операций над базой данных, которые рассматриваются как одна неделимая операция и выполняют перевод базы данных из одного согласованного состояния в другое. Результат транзакции – фиксация (COMMIT) или откат (ROLLBACK) всех входящих в нее операций.
- АСИД-свойства транзакций: атомарность, согласованность, изолированность, долговечность.
- Проблемы параллельного выполнения транзакций: потерянное обновление, зависимость от незафиксированных результатов, анализа несовместимости.
- Блокировки помогают в решении проблем параллельного выполнения транзакций. X-блокировки, S-блокировки. Протокол доступа к данным на основе блокировок.
- Сериализация транзакций. Теорема двухфазной блокировки.