Транзакции и уровни изоляции

Транзакции

Транзакция — это последовательность операций, выполняемая как единое целое.

Для поддержания целостности транзакция должна обладать четырьмя свойствами АСИД: атомарность, согласованность, изоляция и долговечность. Эти свойства называются также ACID-свойствами (от англ., atomicity, consistency, isolation, durability).

- Атомарность (Atomicity)
- Согласованность (или Непротиворечивость) (Consistency)
- Изоляция (Isolation)
- Долговечность (или Устойчивость) (Durability)

Транзакции

По признаку определения границ различают:

- автоматические транзакции
- неявные транзакции
- явные транзакции

Автоматические транзакции

Режим автоматической фиксации транзакций является режимом управления транзакциями по умолчанию. В этом режиме каждая инструкция T-SQL выполняется как отдельная транзакция. Если выполнение инструкции завершается успешно, происходит фиксация; в противном случае происходит откат.

Если возникает ошибка компиляции, то план выполнения пакета не строится и пакет не выполняется.

Автоматические транзакции

```
CREATE TABLE Tab1 (
    Col1 int NOT NULL PRIMARY KEY,
    Col2 char(3)
);


INSERT INTO Tab1 VALUES (1, 'aaa');
INSERT INTO Tab1 VALUES (2, 'bbb');
INSERT INTO Tab1 VALUES (1, 'ccc'); -- Ошибка времени исполнения.


SELECT * FROM Tab1; -- Возвращаются две строки.
```

Неявные транзакции

Если соединение работает в режиме неявных транзакций, то после фиксации или отката текущей транзакции автоматически начинает новую транзакцию. В этом режиме явно указывается только граница окончания транзакции с помощью инструкций COMMIT TRANSACTION и ROLLBACK TRANSACTION.

Неявные транзакции

```
CREATE TABLE Tab1 (
    Col1 int NOT NULL PRIMARY KEY,
    Col2 char(3) NOT NULL
)

-- Первая неявная транзакция, начатая оператором INSERT
INSERT INTO Tab1 VALUES (1, 'aaa')
INSERT INTO Tab1 VALUES (2, 'bbb')
COMMIT TRANSACTION -- Фиксация первой транзакции

-- Вторая неявная транзакция, начатая оператором INSERT
INSERT INTO Tab1 VALUES (3, 'ccc')
SELECT * FROM Tab1
COMMIT TRANSACTION -- Фиксация второй транзакции
```

Явные транзакции

Для определения явных транзакций используются следующие инструкции:

- BEGIN TRANSACTION
- COMMIT TRANSACTION или COMMIT WORK
- ROLLBACK TRANSACTION или ROLLBACK WORK
- SAVE TRANSACTION

Явные транзакции

```
BEGIN TRANSACTION royaltychange
    UPDATE titleauthor
    SET royaltyper = 65
    FROM titleauthor, titles
    WHERE royaltyper = 75 AND titleauthor.title_id = titles.title_id
                AND title = 'The Gourmet Microwave'
    UPDATE titleauthor
    SET royaltyper = 35
    FROM titleauthor, titles
    WHERE royaltyper = 25 AND titleauthor.title_id = titles.title_id
          AND title = 'The Gourmet Microwave'
SAVE TRANSACTION percentchanged

/* После того, как обновлено royaltyper для двух авторов, вставляется точка
сохранения percentchanged, а затем определяется, насколько изменится
заработок авторов после увеличения на 10 процентов цены книги */
UPDATE titles
SET price = price * 1.1
WHERE title = 'The Gourmet Microwave'

SELECT (price * royalty * ytd_sales) * royaltyper
FROM titles, titleauthor
WHERE title = 'The Gourmet Microwave' AND titles.title_id = titleauthor.title_id

/* Откат транзакции до точки сохранения и фиксация транзакции в целом */
ROLLBACK TRANSACTION percentchanged
COMMIT TRANSACTION
```

Управлением параллельным выполнением транзакций

Когда множество пользователей одновременно пытаются модифицировать данные в базе данных, необходимо создать систему управления, которая защитила бы модификации, выполненные одним пользователем, от негативного воздействия модификаций, сделанных другими. Выделяют два типа управления параллельным выполнением:

- **Пессимистическое** управление параллельным выполнением.
- **Оптимистическое** управление параллельным выполнением.

Управлением параллельным выполнением транзакций

Если в случае пессимистического управления в СУБД не реализованы механизмы блокирования, то при одновременном чтении и изменении одних и тех же данных несколькими пользователями могут возникнуть следующие проблемы одновременного доступа:

- **проблема последнего изменения**
- **проблема "грязного" чтения (Dirty Read)**
- **проблема неповторяемого чтения (Non-repeatable or Fuzzy Read)**
- **проблема чтения фантомов (Phantom)**

Управлением параллельным выполнением транзакций

уровень 0 – запрещение «загрязнения» данных. Этот уровень требует, чтобы изменять данные могла только одна транзакция; если другой транзакции необходимо изменить те же данные, она должна ожидать завершения первой транзакции;

уровень 1 – запрещение «грязного» чтения. Если транзакция начала изменение данных, то никакая другая транзакция не сможет прочитать их до завершения первой;

уровень 2 – запрещение неповторяемого чтения. Если транзакция считывает данные, то никакая другая транзакция не сможет их изменить. Таким образом, при повторном чтении они будут находиться в первоначальном состоянии;

уровень 3 – запрещение фантомов. Если транзакция обращается к данным, то никакая другая транзакция не сможет добавить новые или удалить имеющие строки, которые могут быть считаны при выполнении транзакции. Реализация этого уровня блокирования выполняется путем использования блокировок диапазона ключей. Подобная блокировка накладывается не на конкретные строки таблицы, а на строки, удовлетворяющие определенному логическому условию.

Управлением параллельным выполнением транзакций

Уровень изоляции	«Грязное» чтение	Неповторяющееся чтение	Фантомное чтение
read uncommitted	Да	Да	Да
read committed	Нет	Да	Да
repeatable read	Нет	Нет	Да
snapshot	Нет	Нет	Нет
serializable	Нет	Нет	Нет