**Транзакция** − это механизм базы данных, позволяющий таким образом объединять несколько операторов, изменяющих базу данных, чтобы все выполнились или все не выполнились.

Основные свойства транзакции: *атомарность* (операторы изменения БД, включенные в транзакцию, либо выполнятся все, либо не выполнится ни один); *согласованность* (транзакция должна фиксировать новое согласованное состояние БД); *изолированность* (отсутствие взаимного влияния параллельных транзакций на результаты их выполнения); *долговечность* (изменения в БД, выполненные и зафиксированные транзакцией, могут быть отменены только с помощью новой транзакции).

**Неявные транзакции:**

Неявные транзакции в SQL Server могут быть включены для текущего соединения с сервером с использованием инструкции:

SET IMPLICIT\_TRANSACTIONS ON

Обратное переключение осуществляется с использованием ключевого слова OFF вместо ON. Неявная транзакция начинается при выполнении операторов CREATE, DROP, ALTER TABLE, INSERT, DELETE, UPDATE, SELECT, TRUNCATE TABLE, OPEN, FETCH, GRANT, REVOKE. Транзакция продолжается до выполнения оператора COMMIT или ROLLBACK.

Пример:

SET IMPLICIT\_TRANSACTIONS ON

IF EXISTS (SELECT \* FROM SYS.OBJECTS WHERE OBJECT\_ID = OBJECT\_ID(N'DBO.X'))

DROP TABLE X;

DECLARE @c INT, @flag CHAR = 'c';

CREATE TABLE X (K INT);

INSERT X VALUES (1), (2), (3);

SET @c = (SELECT COUNT(\*) FROM X);

PRINT 'Количество строк в таблице X: ' + CAST(@c AS VARCHAR(2));

IF @flag = 'c'

COMMIT; -- Завершение транзакции: фиксация

ELSE

ROLLBACK; -- Завершение транзакции: откат

SET IMPLICIT\_TRANSACTIONS OFF

IF EXISTS (SELECT \* FROM SYS.OBJECTS WHERE OBJECT\_ID = OBJECT\_ID(N'DBO.X'))

PRINT 'Таблица X есть';

ELSE

PRINT 'Таблицы X нет';В данном примере создается таблица X, и в зависимости от значения переменной @flag, транзакция завершается фиксацией или откатом.

**Транзакции с использованием TRY...CATCH:**

Транзакции могут быть использованы с операторами TRY и CATCH для обработки ошибок и выполнения фиксации или отката в зависимости от результата. Пример:

BEGIN TRY

BEGIN TRANSACTION

-- Ваши операторы изменения данных

COMMIT;

END TRY

BEGIN CATCH

PRINT 'Ошибка: ' + ERROR\_MESSAGE();

ROLLBACK;

END CATCHВ этом примере, если происходит ошибка при выполнении операторов изменения данных внутри транзакции, контроль передается блоку CATCH, где выводится сообщение об ошибке, и транзакция откатывается.

**Изоляция транзакций:**

Пример установки уровней изоляции для транзакций:

-- Уровень READ UNCOMMITTED

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

-- Уровень READ COMMITTED

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

-- Уровень REPEATABLE READ

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

-- Уровень SERIALIZABLE

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

**Пример параллельных транзакций:**

-- Транзакция A

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

BEGIN TRANSACTION

SELECT @@SPID, 'insert Товары' 'результат', \* FROM Товары WHERE Наименование = 'Блокнот';

SELECT @@SPID, 'update Заказы' 'результат', Наименование\_товара, Цена\_продажи FROM Заказы WHERE Наименование\_товара = 'Блокнот';

COMMIT;

-- Транзакция B

BEGIN TRANSACTION

INSERT Товары VALUES ('Блокнот', 2, 80);

UPDATE Заказы SET Наименование\_товара = 'Блокнот' WHERE Наименование\_товара = 'Стол';

ROLLBACK;В данном примере две параллельные транзакции A и B выполняют операции чтения и записи в базе данных.

**Пример изоляции транзакций:**

-- Транзакция A

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

BEGIN TRANSACTION

SELECT COUNT(\*) FROM Заказы WHERE Наименование\_товара = 'Стул';

COMMIT;

-- Транзакция B

BEGIN TRANSACTION

UPDATE Заказы SET Наименование\_товара = 'Стул' WHERE Наименование\_товара = 'Стол';

COMMIT;В данном примере транзакция A читает данные, а транзакция B изменяет те же данные между двумя операциями чтения в транзакции A.

**Пример вложенных транзакций:**

-- Транзакция A

BEGIN TRANSACTION

DELETE Заказы WHERE Заказчик = 'Луч';

INSERT Заказы VALUES (14, 'Стул', 78, 10, '01.12.2014', 'Луч');

UPDATE Заказы SET Заказчик = 'Луч' WHERE Наименование\_товара = 'Стул';

SELECT Заказчик FROM Заказы WHERE Наименование\_товара = 'Стул';

COMMIT;

-- Транзакция B

BEGIN TRANSACTION

DELETE Заказы WHERE Заказчик = 'Луч';

INSERT Заказы VALUES (14, 'Стул', 78, 10, '01.12.2014', 'Луч');

UPDATE Заказы SET Заказчик = 'Луч' WHERE Наименование\_товара = 'Стул';

COMMIT;

В данном примере транзакция A включает в себя вложенные транзакции. Вложенная транзакция в транзакции B приводит к откату всей последовательности изменений.