**ОБРАБОТКА ТРАНЗАКЦИЙ**

1. **Использование транзакций**

Обработка транзакций обеспечивает сохранение целостности баз данных за счет того, что пакеты SQL-запросов выполняются полностью или не выполняются вовсе.

Обработка транзакций – это механизм, применяемый для управления наборами SQL-запросов, которые должны быть выполнены только целиком, чтобы в БД не могли попасть результаты частичного выполнения.

При обработке транзакций можно быть уверенным в том, что выполнение набора запросов не было прервано посередине-они или выполнены все, или не выполнен ни один из них. Если никаких ошибок не произошло, результаты всего набора фиксируются в таблицах баз данных. Если произошла ошибка, должны быть отменены все операции, чтобы вернуть БД в прежнее состояние.

Например, нужно добавить в БД новый заказ клиента:

* проверить, есть ли клиент в БД, если нет, то добавить;
* зафиксировать информацию о клиенте;
* получить ID клиента;
* добавить строку в таблицу Заказы;
* если при предыдущей операции возникла ошибка, операция отменяется;
* получить ID заказа, присвоенного в таблице Заказы;
* добавить строку в таблицу Детали\_заказа для каждого заказанного товара;
* если при предыдущей операции возникла ошибка, операция отменяется.

Можно отменить: INSERT, UPDATE, DELETE.

Нельзя отменить: SELECT, CREATE, DROP, их можно использовать в блоке транзакций, но, если произвести откат, то они не будут отменены.

1. **Управление транзакциями**

Нужно сгруппировать запросы в логические блоки и явно указать, когда может быть выполнен откат, а когда – нет.

BEGIN TRANSACTION

…..

COMMIT TRANSACTION

ТРАНЗАКЦИЯ-единый набор SQL–запросов.

ОТКАТ-процесс отмены указанных инструкций.

ФИКСАЦИЯ-запись несохраненных инструкций SQL.

ТОЧКА СОХРАНЕНИЯ-временное состояние в ходе выполнения транзакции, в которое можно вернуться после отмены части инструкций набора (в отличие от отмены всей транзакции).

ROLLBACK- отмена SQL-запросов (откат).

Например

DELETE FROM ORDERS

ROLLBACK;

Выполняется, а затем аннулируется запрос DELETE, т.е. будучи включенными в транзакцию, операции DELETE, INSERT, UPDATE не являются окончательными.

После выполнения инструкций SQL результаты записываются в БД, т.е. происходит *неявная фиксация* (автоматическая запись).

Для принудительной фиксации изменений предназначена инструкция COMMIT.

BEGIN TRANSACTION;

DELETE Детали\_заказа WHERE Order\_num=12345;

DELETE Заказы WHERE Order\_num=12345;

COMMIT TRANSACTION;

Заказ 12345 полностью удаляется из БД, это приводит к обновлению таблиц Заказы и Детали\_заказа.

Транзакция применяется для того, чтобы не допустить частичного удаления заказа.

COMMIT фиксирует изменения только в том случае, если не произошло никаких ошибок. Если первая инструкция будет выполнена, а вторая из-за ошибки нет, удаление не будет зафиксировано.

1. **Точки сохранения**

COMMIT и ROLLBACK позволяют фиксировать или отменять транзакции в целом.

Для более сложных транзакций могут понадобиться частичные фиксации или откаты.

Для отмены части транзакций нужно иметь возможность размещения меток в стратегически важных точках блока инструкций. Тогда, если понадобиться сделать частичный откат, можно вернуть БД состояние, соответствующее одной из меток.

В SQL такие метки называются *точками сохранения*.

SAVE TRANSACTION delete1;

Для отмены всех инструкций после этой точки в SQL Server:

ROLLBACK TRANSACTION delete1;

Запишем полный пример:

BEGIN TRANSACTION;

**INSERT INTO** Поставщики (ID\_поставщика, Name\_поставщика)

VALUES (10001, ‘Производитель игр’);

SAVE TRANSACTION StartOrder;

**INSERT INTO** Заказы (№заказа, дата\_заказа, ID\_поставщика)

VALUES (20100,’2020/12/01’, 10001);

IF @@ERROR<>0

ROLLBACK TRANSACTION StartOrder;

**INSERT INTO** Детали\_заказа (№заказа, №электрзаписи, ID\_товара, Количество, Цена\_товара)

VALUES (20100, 1,’BR01’, 100, 549);

IF @@ERROR<>0

ROLLBACK TRANSACTION StartOrder;

**INSERT INTO** Детали\_заказа (№заказа, №электрзаписи, ID\_товара, Количество, Цена\_товара)

VALUES (20100, 2,’BR03’, 100, 1099);

IF @@ERROR<>0

ROLLBACK TRANSACTION StartOrder;

COMMIT TRANSACTION;

Набор из четырех INSERT–одна транзакция.

Точка сохранения определена после первой инструкции, если один из последующих запросов не выполнится, то отмена транзакции будет до этой точки.

Для контроля успешности запроса используем системную переменную @@ERROR. Если она содержит не 0(<>0), значит произошла ошибка, транзакция отменится до точки сохранения. Если транзакция успешна, то выполняется COMMIT.