

# Семинар 11 – Обработка ошибок и транзакции

## Обзор

На этом семинаре Вы будете использовать Transact-SQL для обработки ошибок и реализации транзакций в базе данных **AdventureWorksLT**.

## Что необходимо для выполнения

- Доступ к облачному сервису Microsoft Azure SQL Database с БД **AdventureWorksLT**.  
или
- Установленный Microsoft SQL Server с SQL Server Management Studio и БД **AdventureWorksLT**.

## Задача 1: Журналирование ошибок

Вам необходимо реализовать скрипт на Transact-SQL для удаления заказов и при этом необходимо обрабатывать любые ошибки, возникающие во время удаления.

**Подсказка:** посмотрите документацию по [THROW](#) и [TRY...CATCH](#) в справочнике по Transact-SQL.

### 1. Выбросите ошибку для несуществующих заказов

В настоящее время для удаления данных по заказу используется следующий код:

```
DECLARE @SalesOrderID int = <ID_заказа_для_удаления>

DELETE FROM SalesLT.SalesOrderDetail WHERE SalesOrderID = @SalesOrderID;

DELETE FROM SalesLT.SalesOrderHeader WHERE SalesOrderID = @SalesOrderID;
```

Этот код всегда выполняется успешно, даже если указанный заказ не существует. Измените код так, чтобы проверить существование заказа с указанным идентификатором, прежде чем пытаться его удалить. Если заказ не существует, то Ваш код должен выбросить ошибку: «Заказ #<SalesOrderID> не существует». В противном случае он должен удалить данные по заказу.

### 2. Обработка ошибок

Сейчас Ваш код выбрасывает ошибку, если указанный заказ не существует. Теперь Вы должны дописать свой код, чтобы поймать эту (или любую другую) ошибку и вывести сообщение об ошибке в пользовательский интерфейс с помощью команды PRINT.

## Задача 2: Обеспечение согласованности данных

Вы внедрили логику обработки ошибок в некоторый код на Transact-SQL, который удаляет заказанные товары и заголовки заказов. Тем не менее, Вы обеспокоены тем, что в результате сбоя (например, при отключении электропитания) Ваш скрипт приведет к несогласованности данных в виде оставшихся заголовков заказов, для которых данные по товарам были удалены.

**Подсказка:** посмотрите документацию по [транзакциям](#) в справочнике по Transact-SQL.

### 1. Внедрение транзакций

Усовершенствуйте Ваш код, созданный в предыдущем задании, чтобы два оператора DELETE рассматривались как единая транзакция. В обработчике ошибок измените код так, чтобы, если транзакция находится в процессе выполнения, она бы отменялась, и ошибка пробрасывалась в

клиентское приложение. Если транзакция не выполняется, то обработчик ошибок должен просто напечатать сообщение об ошибке (PRINT).

Чтобы проверить транзакции, добавьте инструкцию THROW между двумя операторами DELETE для имитации возникновения ошибки. При тестировании с действительным (существующим в БД) идентификатором заказа, ошибка должна быть повторно выброшена обработчиком ошибок, и никакие строки не должны быть удалены из таблиц.