

Лабораторный практикум

МДК 11.01 Технология разработки
и защиты баз данных

SQL

Куропаткина О.П.

ЛАБОРАТОРНАЯ РАБОТА №11

ИНСТРУКЦИЯ TRY CATCH

Начиная с версии SQL Server 2005, **исключения можно перехватывать** для обработки с помощью инструкций TRY и CATCH.

Исключением (exception) называется проблема (обычно ошибка), которая не позволяет продолжать выполнение программы. Выполняющаяся программа не может продолжать исполнение по причине недостаточности информации, необходимой для обработки ошибки в данной части программы. Поэтому задача обработки ошибки передается другой части программы. **Роль инструкции TRY заключается в перехвате исключения.** (Поскольку для реализации этого процесса обычно требуется несколько инструкций, то обычно применяется термин "блок TRY", а не "инструкция TRY".) **Если же в блоке TRY возникает исключение,** компонент системы, называющийся обработчиком исключений, **доставляет это исключение для обработки** другой части программы. Эта часть программы обозначается ключевым словом **CATCH**.

Обработка исключений с помощью блоков TRY и CATCH предоставляет программисту множество преимуществ, включая следующие:

- исключения предоставляют аккуратный способ определения ошибок без загромождения кода дополнительными инструкциями;
- исключения предоставляют механизм прямой индикации ошибок, вместо использования каких-либо побочных признаков;
- программист может видеть исключения и проверить их в процессе компиляции.

Пример использования данной инструкции представлен ниже.

Код создания таблицы TestTable помещаем в блок TRY. Если при выполнении инструкции возникнет ошибка, свидетельствующая о том, что данная таблица уже существует (код ошибки 2714), выводим соответствующее сообщение.

```
181 -----
182 ----- Блоки TRY и CATCH -----
183 -- предназначены для обработки ошибок
184
185 BEGIN TRY -- Попытка выполнить код, если возникает ошибка то бросаем ее в блок CATCH, иначе продолжаем работу без блока CATCH.
186
187     CREATE TABLE TestTable
188     (
189         col1 int,
190         col2 varchar(10)
191     );
192
193 END TRY
194 BEGIN CATCH -- блок CATCH - блок-обработчик ошибок
195
196     DECLARE @ErrorNo int,
197             @Message nvarchar(4000);
198
199     SELECT
200         @ErrorNo = ERROR_NUMBER(), --системная функция, возвращает код текущей ошибки
201         @Message = ERROR_MESSAGE(); --системная функция, возвращает сообщение текущей ошибки
202
203     IF @ErrorNo = 2714
204         PRINT 'Данная таблица уже существует!';
205     ELSE
206         PRINT CAST(@ErrorNo as varchar)+' '+@Message;
207 END CATCH
208 GO
```

108 %

Сообщения

Данная таблица уже существует!

Еще один вариант применения блока TRY CATCH рассмотрим на примере.

Нам необходимо добавить четыре строки в таблицу Товаров. В одной из них ошибка – ссылка на ID производителя, которого нет в базе данных. Для того, чтобы в случае ошибки откатить исполнение всех четырех строк, поместим их в блок TRANSACTION и TRY. Если в транзакции возникнет ошибка, блок TRY ее перехватит и в блоке CATCH транзакция будет отменена.

```
1 USE ForEducationSQL;
2
3 BEGIN TRY
4     BEGIN TRANSACTION
5         insert into Product (Title, Cost, ManufacturerID) values ('Товар 1', 1850, 5);
6         insert into Product (Title, Cost, ManufacturerID) values ('Товар 2', 1950, 15);
7         insert into Product (Title, Cost, ManufacturerID) values ('Товар 3', 2050, 55);
8         -- ошибка ссылочной целостности (производителя с таким ID в базе данных нет)
9         insert into Product (Title, Cost, ManufacturerID) values ('Товар 4', 2150, 25);
10    COMMIT TRANSACTION
11    PRINT 'Transaction committed'
12 END TRY
13
14 BEGIN CATCH
15     ROLLBACK -- откатываем транзакцию
16     PRINT 'Transaction rolled back';
17     THROW
18 END CATCH
```

108 %

Сообщения

(затронута одна строка)

(затронута одна строка)

(затронута строк: 0)

Transaction rolled back

Сообщение 547, уровень 16, состояние 0, строка 7

Конфликт инструкции INSERT с ограничением FOREIGN KEY 'FK_Product_Manufacturer'. Конфликт произошел в базе данных "ForEducationSQL", таблица "dbo.Manufacturer", column 'ID'.

САМОСТОЯТЕЛЬНО:

- 1. Напишите инструкцию INSERT, которая будет добавлять в таблицу Client сведения о трех клиентах по аналогии с предыдущим примером. Инструкцию поместите в блок TRY CATCH. Спровоцируйте ошибку, убедитесь, что инструкция работает корректно.**
- 2. Создайте хранимую процедуру, которая будет принимать в качестве входных параметров название товара и количество. Функционал: необходимо добавить в таблицу ProductSale сведения о продаже, дату продажи фиксировать как текущую системную дату. Если товара с таким названием в базе данных нет, выведите соответствующее сообщение. В случае успешного добавления данных в таблицу, выведите сообщение: Сведения о продаже товара «Название товара» в количестве «Количество» на сумму «Цена товара * Количество» добавлены в базу данных. Для обработки возможных ошибок используйте блок TRY CATCH.**