



Microsoft Partner
Silver Learning



TRANSACTION-SQL

Процедурный язык Transact-SQL



ITVDSN
IT VIDEO DEVELOPERS NETWORK

TRANSACT-SQL

Автор курса



Станислав Зуйко



MC ID: 12974808

TRANSACT-SQL

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)

Доступ можно получить через руководство вашего учебного центра



Проверьте как Вы усвоили данный материал на TestProvider.com

TRANSACT-SQL

Тема

Процедурный язык Transact-SQL

Процедурный язык Transact-SQL

План

1. Локальные переменные
2. Инструкции управления потоком
 - условные конструкции
 - циклы
3. Обработка ошибок

Local variables

Локальные переменные

Локальная переменная – объект, содержащий некоторое значение определенного типа.

Объявление переменной:

```
DECLARE @age smallint
```

Присвоение переменной значения:

1) при объявлении (по умолчанию присваивается значение NULL)

```
DECLARE @age smallint = 2;
```

2) с помощью инструкции SET

```
SET @name = 'Elijah';
```

3) с помощью инструкции SELECT

```
SELECT @age = 5;
```

Control-of-flow statements

Инструкции управления потоком



Без языка управления выполнением отдельные инструкции языка Transact-SQL выполнялись бы последовательно – так, как они написаны. Язык управления выполнением позволяет связывать инструкции друг с другом, а также создавать независимо выполняющиеся конструкции, как в языках программирования.

Инструкции управления потоком

Условные конструкции

В программировании существует три вида алгоритмических конструкций:

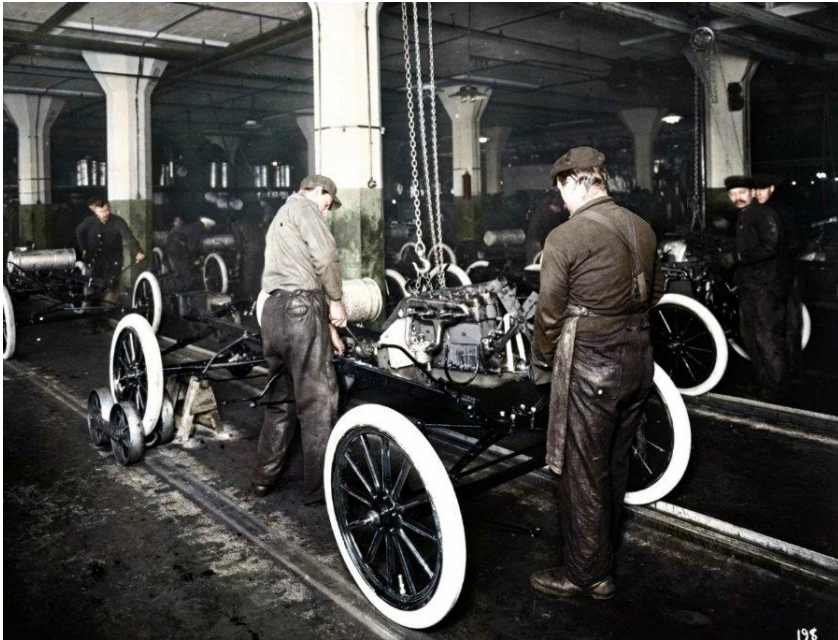
- линейный алгоритм
- разветвляющийся
- циклический



Инструкция **IF...ELSE** определяет условную конструкцию (разветвляющийся алгоритм), с помощью которой выполнится определенный набор инструкций в зависимости от того, является ли условие истинным.

Инструкции управления потоком

Циклические конструкции



Инструкция **WHILE** определяет циклическую конструкцию, с помощью которой определенный набор инструкций может выполняться многократно, пока условие является истинным.

Один проход по телу цикла (выполнение определенного набора инструкций) – называется итерацией.

С помощью ключевого слова **BREAK** мы прерываем выполнение текущей итерации и цикла в целом.

С помощью ключевого слова **CONTINUE** мы прерываем выполнение только текущей итерации.

Обработка ошибок

TRY...CATCH



Группа инструкций на языке Transact-SQL может быть заключена в блок **TRY**. Если ошибка возникает в блоке TRY, управление передается следующей группе инструкций, заключенных в блок **CATCH**.

Ошибки с уровнем серьезности, меньшим или равным 10, рассматриваются как предупреждения или информационные сообщения и не обрабатываются с помощью конструкции **TRY...CATCH**.

Ошибки с уровнем серьезности, большим или равным 20, заставляющие компонент Database Engine закрыть соединение, не могут быть обработаны конструкцией TRY...CATCH. Однако конструкция TRY...CATCH обеспечивает обработку ошибок с уровнем серьезности, большим или равным 20, если соединение не закрывается.

Обработка ошибок

RAISEERROR

Инструкция **RAISEERROR** выводит определенное пользователем сообщение об ошибке и устанавливает флаг системной ошибки.

`RAISEERROR(msg_id | msg_str, severity, state, arguments)`

- **msg_id** – номер ошибки в определяемом пользователем сообщении должен быть больше, чем 50 000, т.к. все номера ошибок меньшие или равные 50 000 определены системой и зарезервированы компонентом Database Engine. Значения номеров ошибок сохраняются в глобальной переменной @@error;
- **msg_str** – сообщение об ошибке;
- **severity** – степень серьезности ошибки (значения от 0 до 25). Уровень серьезности от 0 до 18 может указать любой пользователь. Уровни серьезности от 19 до 25 могут быть указаны только членами предопределенной роли сервера sysadmin и пользователями с разрешениями ALTER TRACE. Если задать серьезность меньше или равную 10, инструкция RAISERROR возвращает сообщения, минуя блок CATCH;
- **state** – состояние (значения от 0 до 255). Если одна и та же пользовательская ошибка возникает в нескольких местах, то при помощи уникального номера состояния для каждого местоположения можно определить, в каком месте кода появилась ошибка;
- **arguments** – параметры, использующиеся при подстановке в строке msg_str или msg_id.

Обработка ошибок

THROW

В SQL Server 2012 добавлена еще одна инструкция **THROW** – это другой механизм возврата, который работает подобно рассмотренной ранее инструкции RAISEERROR.

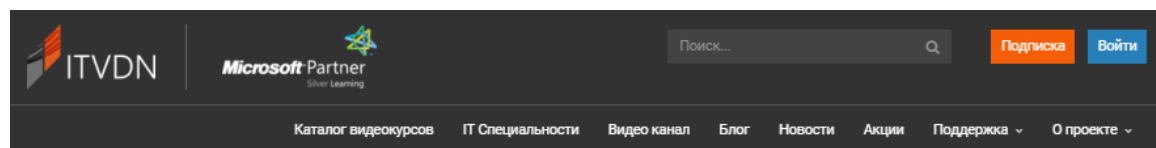
`THROW error_number, message, state`

- **error_number** – номер ошибки;
- **message** – сообщение об ошибке;
- **state** – состояние.

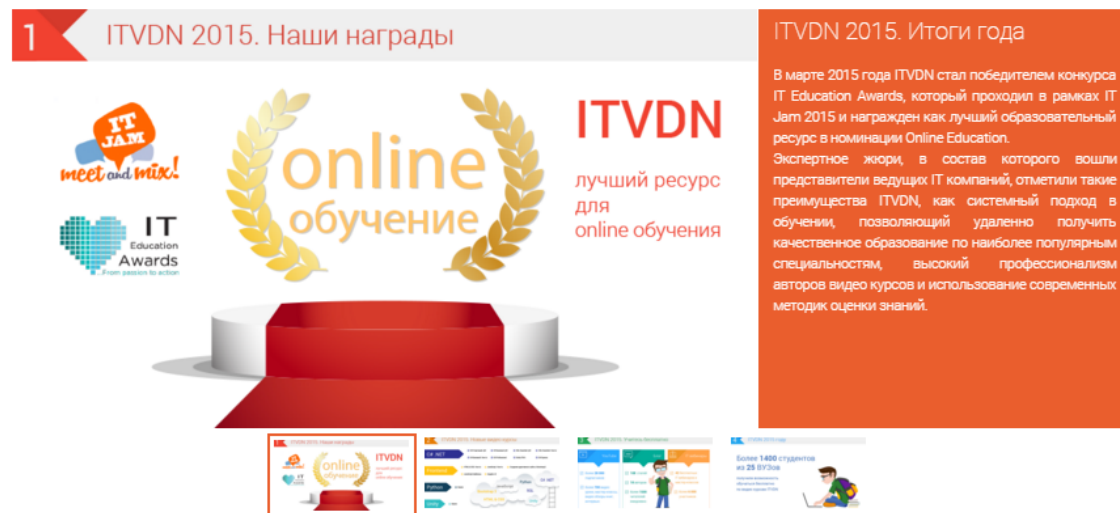
Значение степени серьезности ошибки не указывается и всегда равно 16-ти.

Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.



Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics

Новые видео

Исключения	0
Итераторы и генераторы	0

Популярные видео курсы

Видео курс C# Стартовый (для начинающих)	9 уроков (16 ч. 3 мин.)
Видео курс по шаблонам проектирования	29 уроков (16 ч. 7 мин.)

Теги

.NET Developer
Frontend Developer



Проверка знаний

TestProvider.com

TestProvider | Мы помогаем людям оценить себя

Регистрация Войти

Главная Каталог Сертификация Microsoft Поддержка О нас

Тестирование

Языки программирования и информационные технологии

Microsoft

C# ASP.NET MVC JavaScript Patterns Of Design SQL Architecture Guide WCF HTML&CSS XML SEO WPF HTML5&CSS3 JQuery XNA SharePoint GUI for Android Windows Azure Platform Microsoft Patterns&Practices TFS SCRUM ReSharper TDD WWF LINQ Entity Framework Windows Forms Refactoring Microsoft Expression Blend 4 Windows Phone 8 Windows 8 AppStore Visual Studio Tips&Tricks MSF MEF SilverLight AJAX MEF Service Oriented Architecture

Пройти тест

Наши партнеры

Microsoft Partner CyberBionic ITVDN PROMETRIC TEST CENTER PEARSON VUE Authorized Test Center Windows Azure Cloud Partner EBA

Дополнительные ресурсы:

Очное обучение On-line обучение Видео обучение

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



TRANSACTION-SQL

Q&A

Информационный видеосервис для разработчиков программного обеспечения

