Темы урока

[**Цель урока**](#_r8dwlw946bhp) **1**

[**Схема и данные для работы в классе**](#_tqfa6obmvxfv) **2**

[**⚠ Виды функций в T-SQL**](#_vkpkw1elsxs7) **2**

[**Скалярные функции T-SQL**](#_xkkqynomgo0l) **3**

[Строковые функции на примере LEN](#_eve5xslg30cx) 3

[Функции даты и времени на примере YEAR](#_9nhwk59rbws0) 3

[**Агрегатные функции**](#_mc0sdtr7a3j9) **3**

[**Самостоятельная работа #1**](#_xliovyhyekrq) **3**

[**Вложенные запросы в условии**](#_aa72t5yvhwdg) **4**

[**Самостоятельная работа #2**](#_4eqfvqrm8mfa) **4**

[**Объединения**](#_sdc9p4xbq7fa) **4**

[**Самостоятельная работа #3**](#_47cmhzcvgdbn) **4**

[**Группировка значений**](#_dflyltvvykco) **4**

[**Самостоятельная работа #4**](#_11dkn2njqzs1) **5**

[**Домашнее задание**](#_y4t66bgbzg1n) **5**

# Цель урока

Дать больше практики для работы с SQL, особенный упор на агрегатные функции и объединения нескольких таблиц.

# 

# Схема и данные для работы в классе

Для работы в классе мы будем использовать БД, состоящую из четырёх таблиц, хранящих информацию о каталоге и продажах простого магазина:

* Customer : клиент
  + Id : идентификатор
  + Name : имя
* Product: продукт
  + Id : идентификатор
  + Name: наименование
  + Price: цена
* Order : заказ
  + Id : идентификатор
  + CustomerId : идентификатор клиента
  + OrderDate : дата заказа
  + Discount : размер скидки
* OrderItem: единичная позиция заказа
  + OrderId : идентификатор заказа
  + ProductId : идентификатор продукта
  + NumberOfItems: количество единиц

*Схему и данные предлагаю не создавать всем вместе, а просто дать как данность, потому что проектирование БД, всё-таки, уже было читано, а вот статистических (агрегатных) функций ещё нет.*

Показываем им схему по слайду, даём скрипты на создание схемы и заполнение таблиц данными и начинаем уже непосредственно с запросов после небольшого перечисления самих функций.

Внешних ключей я здесь умышленно не вешал, чтобы не усложнять то, что не в фокусе урока.

Код создания схемы: **L29\_C01\_LiveHeroTour2.Schema.sql**  
Код для вставки данных: **L29\_C02\_LiveHeroTour2.Data.sql**

# ⚠ Виды функций в T-SQL

В T-SQL существует большое количество встроенных функций, помогающих решать различные задачи. Среди прочих хочется выделить две основные группы:

* **Скалярные** функции обрабатывают одиночное значение и возвращают одиночное значение,
* **Агрегатные** функции выполняют вычисление на наборе значений и возвращают одиночное значение.

# 

# Скалярные функции T-SQL

Ещё раз: скалярные функции обрабатывают и возвращают одиночное значение.

Для ознакомления мы рассмотрим две скаляр, чтобы уяснить подход к их использованию, также мы посмотрим на документацию, где можно ознакомиться с полным перечнем функций.

Код для демонстраций: **L29\_C03\_Scalar-value\_functions\_LEN\_YAER\_demo.sql**

## Строковые функции на примере LEN

Объясняем по слайду и даём примеры.

**⚠** Нажимаем F1 выделяя имя функции в MS SQL Server Management Studio, чтобы показать справку и смежные функции.

## Функции даты и времени на примере YEAR

Объясняем по слайду и даём примеры.

**⚠** Нажимаем F1 выделяя имя функции в MS SQL Server Management Studio, чтобы показать справку и смежные функции.

# Агрегатные функции

Ещё раз: агрегатная функция выполняет вычисление на наборе значений и возвращает одиночное значение (за исключением COUNT, не учитывают значения NULL).

Сначала рассказываем по слайду назначение пяти наиболее востребованных агрегатных функций, а потом даём живую демонстрацию.

Код для демонстрации: **L29\_C04\_Aggregate\_function\_demo.sql**

# Самостоятельная работа #1

Написать запросы, возвращающие следующие данные:

1. Полное количество записей в таблице OrderItem
2. Количество уникальных заказов (по таблице OrderItem)
3. Максимальный номер заказа (по таблице Order)
4. Средний размер скидки (по таблице Order)
5. Дата первой и последней продажи (по таблице Order)
6. Дата последней продажи в 2018 году (по таблице Order)
7. Максимальная длина наименования товара (по таблице Product)

Код примеров решения: **L29\_C05\_Simple\_Scalar\_and\_Aggregate\_Functions\_SF.sql**

# Вложенные запросы в условии

Вложенный запрос — это запрос, который используется внутри инструкции SELECT, INSERT, UPDATE или DELETE или внутри другого вложенного запроса.

Код для демонстрации: **L29\_C06\_Sub-queries\_demo.sql**

# Самостоятельная работа #2

Написать запросы, возвращающие следующие данные:

1. Номер заказа с максимальной скидкой в 2016 году
2. Номер первого заказа в 2019 году
3. ID и имя клиента, получившего максимальную скидку в 2016 году
4. ID и имя клиента, сделавшего первый заказ в 2019 году

Код примеров решения: **L29\_C07\_Sub-queries\_SF.sql**

# Объединения

На базе INNER JOIN решаем на демонстрации следующую задачу:

* Найти список товаров с ценой, количеством и стоимостью для заказа c ID = 22, а также посчитать полную стоимость этого заказа

Код для демонстрации: **L29\_C08\_Joins\_demo.sql**

# Самостоятельная работа #3

Написать запросы, возвращающий следующие данные:

* Итоговая сумма, потраченная Марией

Код примеров решения: **L29\_C09\_Joins\_SF.sql**

Тут нужно отметить, что это не совсем честно рассчитанное значение потраченных Марией денег, так как мы использовали для рассчётов цену товара без учёта скидок к каждому из заказов. Но применить скидку, в данном случае, не получится, так как в разных заказах она разная (среднее арифметическое нам не подойдёт :)

Без группировок эту задачу не решить. Так можно плавно перейти к группировкам.

# Группировка значений

Рассказываем про группировку по слайду, больше там и не скажешь, лучше показать на живом примере.

Сначала простой вариант с COUNT

А затем прилюдно разобрать пример с честным расчётом денег, потраченных Марией.

Код: **L29\_C10\_Group\_By\_demo.sql**.

Рассказать 2 лайфхака:

1. Что в GROUP BY тупо пишутся все поля без агрегатных функций,
2. Использование псевдонимов для полей не допускается (нужно писать оригинальные столбцы или выражения, выводящиеся в полях без агрегатных функций).

# Самостоятельная работа #4

1. Написать запрос, возвращающий полную итоговую сумму, потраченную каждым клиентом в формате:
   * Id клиента
   * Имя клиента
   * Итоговая потраченная сумма
2. Добавить разбивку по годам и сортировку по имени, а затем по году.

# Домашнее задание

Вообще можно их особенно не мучать, для расширения кругозора я им показал всё что необходимо.

Однако, если уж очень очень им захочется, можно, например, подумать вот над такими запросами:

* Найти год с самой большой выручкой
* Вывести все года, в которых были заказы и рядом id и имена клиентов с самым большим заказом по данному году.