



Базы Данных

(T-SQL: скалярные и агрегатные функции,
агрегирование объединений, группировка)

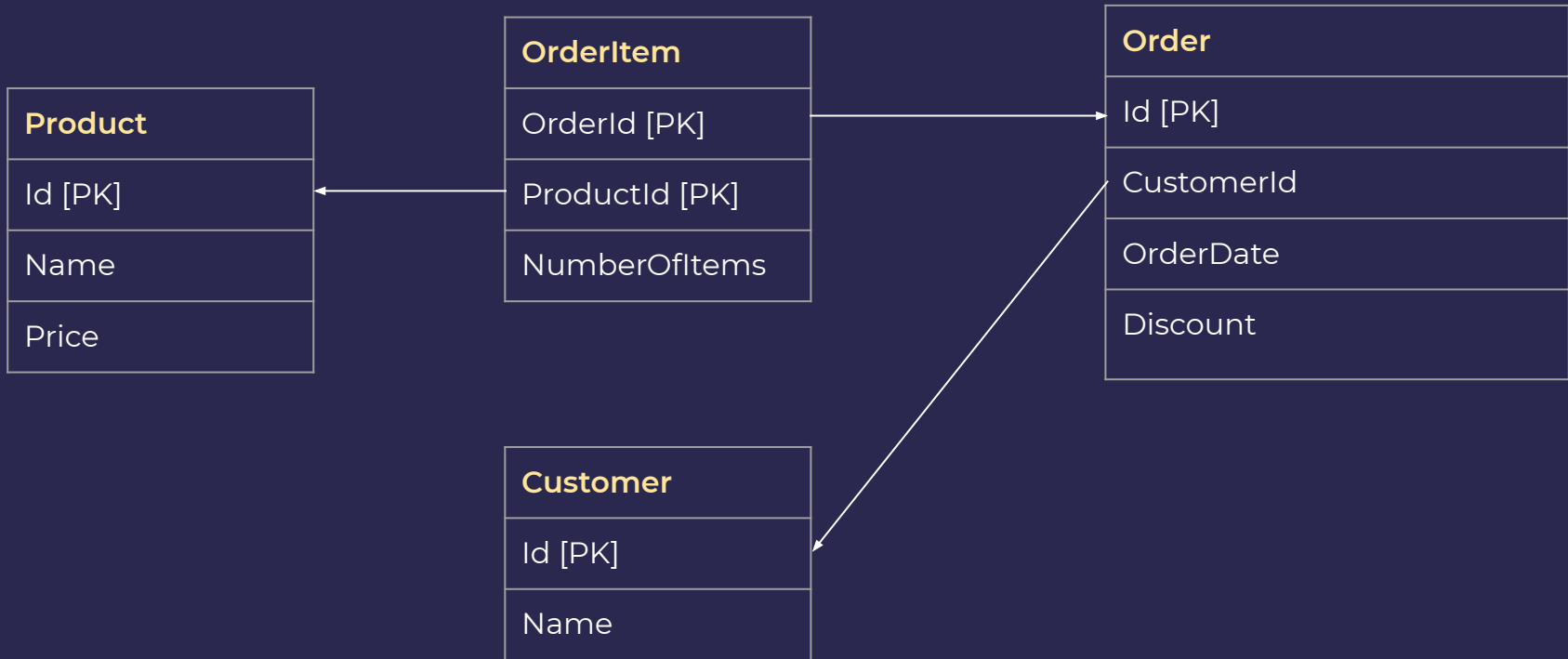
Андрей Голяков

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





Функции T-SQL

В T-SQL существует большое количество встроенных функций. 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.



Строковые функции в T-SQL (На примере LEN)

Функция **LEN** возвращает количество символов строки, исключая конечные пробелы. Может использоваться с константами, при выборке данных или в условном выражении:

```
-- LEN: вызов для константы или переменной
```

```
SELECT LEN('Hello, world!');
```

```
-- LEN: вызов в качестве вычисляемого поля
```

```
SELECT [SomeField], LEN([StringField]) FROM [SomeTable]
```

```
-- LEN: вызов в условном выражении
```

```
SELECT * FROM [SomeTable] WHERE LEN([StringField]) > 20
```



Функции даты/времени (На примере YEAR)

Функция **YEAR** возвращает год как целое число от даты, переданной аргументом. Может использоваться с константами, при выборке данных или в условном выражении:

```
-- YEAR: вызов для константы или переменной
```

```
DECLARE @year AS INT
```

```
SET @year = YEAR(GETUTCDATE())
```

```
SELECT @year;
```

```
-- YEAR: вызов в качестве вычисляемого поля
```

```
SELECT YEAR([DateTimeField]) FROM [SomeTable]
```

```
-- YEAR: вызов в условном выражении
```

```
SELECT [AnyField] FROM [SomeTable] WHERE YEAR([DateTimeField]) = @year
```



Агрегатные/статистические функции

- **COUNT, COUNT_BIG** : возвращает количество элементов, найденных в группе набора данных. Функции различаются только типами возвращаемых значений (int / bigint):

```
SELECT COUNT(*) FROM SomeTable
```

```
SELECT COUNT(DISTINCT SomeField) FROM SomeTable
```

- **MAX, MIN** : возвращает соответственно максимальное или минимальное значение элементов, найденных в группе набора данных:

```
SELECT MAX(SomeField), MIN(SomeField) FROM SomeTable
```

- **SUM**: возвращает сумму всех, либо только уникальных, значений в выражении. Функция SUM может быть использована только для числовых столбцов. Значения NULL пропускаются:

```
SELECT SUM(SomeField) FROM SomeTable
```

- **AVG**: возвращает среднее арифметическое группы значений. Значения NULL не учитываются:

```
SELECT AVG(SomeField) FROM SomeTable
```

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)



Вложенные запросы в условиях

-- Найти ID и имена клиентов, сделавших заказы в 2018 году

```
SELECT C.[Id], C.[Name]
FROM [dbo].[Customer] AS C
WHERE C.[Id] IN (
    SELECT O.CustomerId
    FROM [dbo].[Order] AS O
    WHERE YEAR(O.OrderDate) = 2018
)
```

-- Найти ID и название товара с максимальной длиной наименования
-- (надо понимать, что потенциально такой запрос может вернуть N записей,
-- в случае, если у нас N единиц товара, имеющих одинаковую длину,
-- которая окажется максимальной)

```
SELECT Id, [Name]
FROM [dbo].[Product] AS P
WHERE LEN(P.[Name]) = (
    SELECT MAX(LEN(P.[Name]))
    FROM [dbo].[Product] AS P
)
```



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



Агрегирование значений в объединениях

Алгоритм действий

1. Сначала определяем по схеме, в каких таблицах лежат необходимые для вывода и условий данные
2. Объединяем целевые таблицы, при необходимости используя связи с промежуточными таблицами
3. Определяем условия с помощью конструкции WHERE
4. Перечисляем необходимые поля и/или, при необходимости, вводим расчётные поля

Совместная работа

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



Самостоятельная работа #3

Написать запрос, возвращающий полную итоговую сумму, потраченную Марией.



Группировка значений (с помощью GROUP BY)

Команда **GROUP BY** позволяет группировать результаты при выборке из базы данных. При этом агрегирующие функции будут применяться к отдельным группам.

-- Общее количество обработанных заказов

```
SELECT COUNT(*)  
FROM [dbo].[Order] AS O
```

-- Количество обработанных заказов сгруппированных по годам

```
SELECT YEAR(O.[OrderDate]), COUNT(*)  
FROM [dbo].[Order] AS O  
GROUP BY YEAR(O.[OrderDate])
```



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.



Домашняя работа



Спасибо за внимание.

