

Лабораторный практикум

МДК 11.01 Технология разработки
и защиты баз данных

SQL

Куропаткина О.П.

ЛАБОРАТОРНАЯ РАБОТА №5

Основы DML

Data Manipulation Language

(язык манипулирования данными)

SQL

Structured Query Language

(язык структурированных запросов)

SQL — это аббревиатура выражения Structured Query Language (язык структурированных запросов). SQL основывается на реляционной алгебре и специально разработан для взаимодействия с реляционными базами данных.

SQL является, прежде всего, информационно-логическим языком, предназначенным для описания хранимых данных, их извлечения и модификации. SQL не является языком программирования. Вместе с тем конкретные реализации языка, как правило, включают различные процедурные расширения.

SQL является стандартизированным языком. Стандартный SQL поддерживается комитетом стандартов ANSI (Американский национальный институт стандартов), и соответственно называется ANSI SQL.

Многие разработчики СУБД расширили возможности SQL, введя в язык дополнительные операторы или инструкции. Эти расширения необходимы для выполнения дополнительных функций или для упрощения выполнения определенных операций. И хотя часто они очень полезны, эти расширения привязаны к определенной СУБД и редко поддерживаются более чем одним разработчиком. Все крупные СУБД и даже те, у которых есть собственные расширения, поддерживают ANSI SQL (в большей или меньшей степени). Отдельные же реализации носят собственные имена (PL-SQL, Transact SQL и т.д.). **Transact-SQL(T-SQL) – реализация языка SQL корпорации Microsoft, используемая, в частности, и в SQL Server.**

ЗАПРОСЫ НА ВЫБОРКУ ДАННЫХ (ОПЕРАТОР SELECT)

SELECT– наиболее часто используемый SQL оператор. Он предназначен для выборки информации из таблиц. Чтобы при помощи оператора SELECT извлечь данные из таблицы, нужно указать как минимум две вещи — что вы хотите выбрать и откуда.

1	USE ForEducationSQL
2	GO
3	
4	SELECT Title FROM Service

108 %	
Результаты Сообщения	
	Title
1	Замена сальника привода
2	Ремонт компрессора кондиционера

В приведенном выше операторе используется оператор SELECT для выборки одного столбца под названием Title из таблицы Service. Искомое имя столбца указывается сразу после ключевого слова SELECT, а ключевое слово FROM указывает на имя таблицы, из которой выбираются данные.

Выборка нескольких столбцов

Для выборки из таблицы нескольких столбцов используется тот же оператор SELECT.

Отличие состоит в том, что после ключевого слова SELECT необходимо через запятую указать несколько имен столбцов.

1	USE ForEducationSQL
2	GO
3	
4	SELECT Title, Cost, DurationInSeconds FROM Service

108 %			
Результаты Сообщения			
	Title	Cost	DurationInSeconds
1	Замена сальника привода	3820.00	34200
2	Ремонт компрессора кондиционера	4760.00	9000
3	Ремонт автоэлектрики	4230.00	27000

Выборка всех столбцов

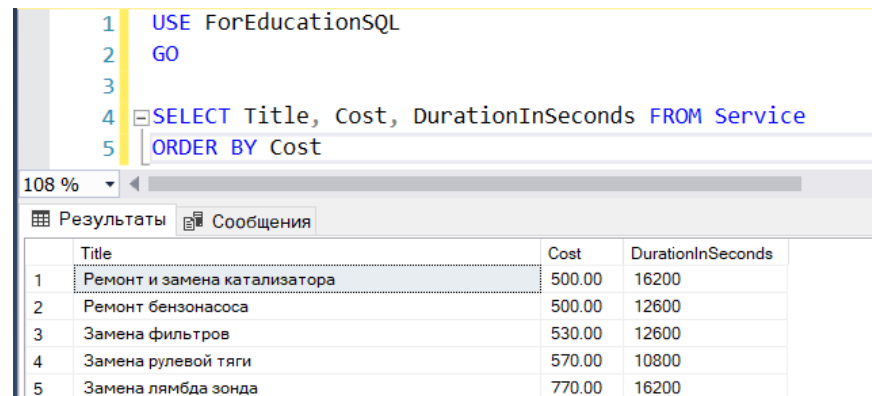
Помимо возможности осуществлять выборку определенных столбцов (одного или нескольких), при помощи оператора SELECT можно запросить все столбцы, не перечисляя каждый из них.

1	USE ForEducationSQL
2	GO
3	
4	SELECT * FROM Service

108 %				
Результаты Сообщения				
	ID	Title	Cost	DurationInSeconds
1	1	Замена сальника привода	3820.00	34200
2	2	Ремонт компрессора кондиционера	4760.00	9000
3	3	Ремонт автоэлектрики	4230.00	27000

Сортировка данных

В результате выполнения запроса на выборку данные выводятся в том порядке, в котором они находятся в таблице. Для точной сортировки выбранных при помощи оператора `SELECT` данных используется предложение ***ORDER BY***. В этом предложении указывается имя одного или нескольких столбцов, по которым необходимо отсортировать результаты.



The screenshot shows a SQL query window with the following text:

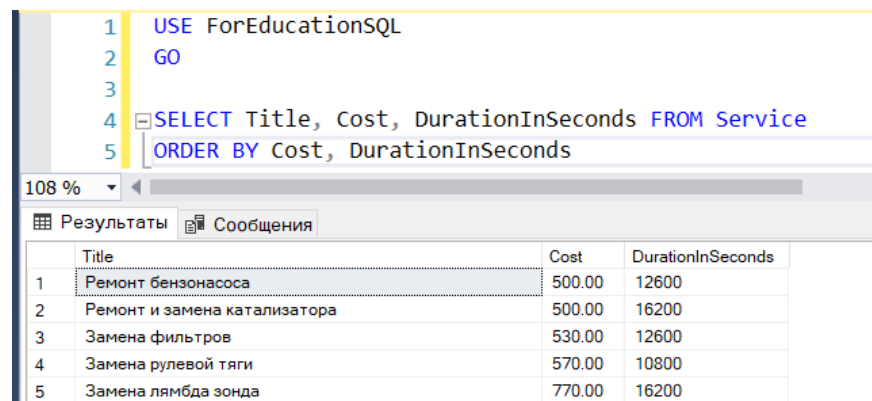
```
1 USE ForEducationSQL
2 GO
3
4 SELECT Title, Cost, DurationInSeconds FROM Service
5 ORDER BY Cost
```

Below the query window, the 'Results' tab is active, displaying a table with 3 columns: Title, Cost, and DurationInSeconds. The table contains 5 rows of data, sorted by Cost in ascending order.

	Title	Cost	DurationInSeconds
1	Ремонт и замена катализатора	500.00	16200
2	Ремонт бензонасоса	500.00	12600
3	Замена фильтров	530.00	12600
4	Замена рулевой тяги	570.00	10800
5	Замена лямбда зонда	770.00	16200

Сортировка по нескольким столбцам

Чтобы осуществить сортировку по нескольким столбцам, просто укажите их имена через запятую.



The screenshot shows a SQL query window with the following text:

```
1 USE ForEducationSQL
2 GO
3
4 SELECT Title, Cost, DurationInSeconds FROM Service
5 ORDER BY Cost, DurationInSeconds
```

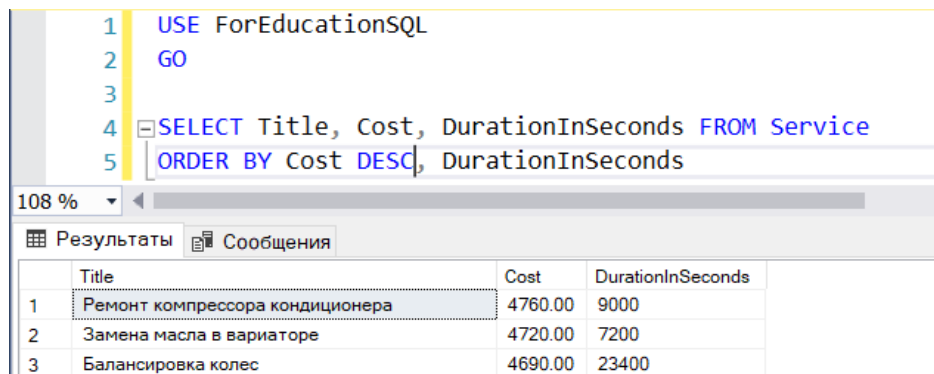
Below the query window, the 'Results' tab is active, displaying a table with 3 columns: Title, Cost, and DurationInSeconds. The table contains 5 rows of data, sorted by Cost in ascending order, and then by DurationInSeconds in ascending order for rows with the same Cost.

	Title	Cost	DurationInSeconds
1	Ремонт бензонасоса	500.00	12600
2	Ремонт и замена катализатора	500.00	16200
3	Замена фильтров	530.00	12600
4	Замена рулевой тяги	570.00	10800
5	Замена лямбда зонда	770.00	16200

Важно понимать, что при сортировке по нескольким столбцам порядок сортировки будет таким, который указан в запросе. Другими словами, в примере, приведенном выше, услуги сортируются по столбцу `DurationInSeconds`, только если существует несколько строк с одинаковыми значениями `Cost`. Если никакие значения столбца `Cost` не совпадают, данные по столбцу `DurationInSeconds` сортироваться не будут.

Указание направления сортировки

В предложении **ORDER BY** можно также использовать порядок сортировки по убыванию. Для этого необходимо указать ключевое слово **DESC**. Ключевое слово **DESC** применяется только к тому столбцу, после которого оно указано.



```
1 USE ForEducationSQL
2 GO
3
4 SELECT Title, Cost, DurationInSeconds FROM Service
5 ORDER BY Cost DESC, DurationInSeconds
```

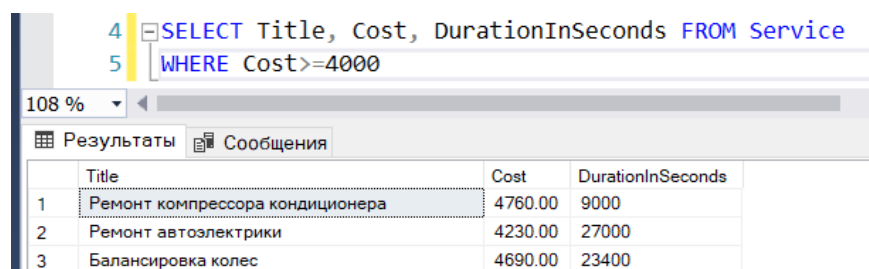
108 %

Результаты Сообщения

	Title	Cost	DurationInSeconds
1	Ремонт компрессора кондиционера	4760.00	9000
2	Замена масла в вариаторе	4720.00	7200
3	Балансировка колес	4690.00	23400

Фильтрация данных

В таблицах баз данных обычно содержится много информации и довольно редко возникает необходимость выбирать все строки таблицы. Выборка только необходимых данных включает в себя критерий поиска, также известный под названием предложение фильтрации. В операторе **SELECT** данные фильтруются путем указания критерия поиска в предложении **WHERE**. Предложение **WHERE** указывается сразу после названия таблицы (предложения **FROM**) следующим образом:



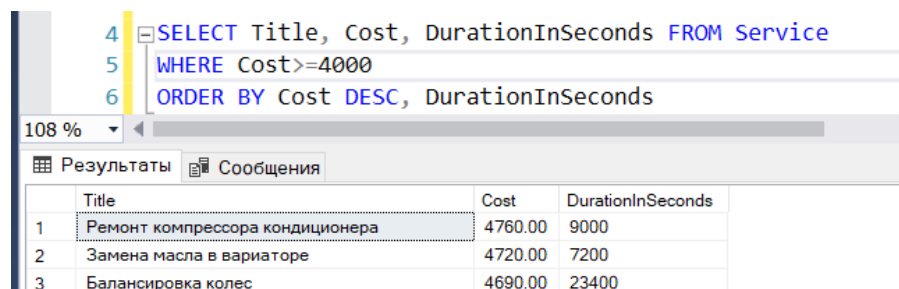
```
4 SELECT Title, Cost, DurationInSeconds FROM Service
5 WHERE Cost >= 4000
```

108 %

Результаты Сообщения

	Title	Cost	DurationInSeconds
1	Ремонт компрессора кондиционера	4760.00	9000
2	Ремонт автоэлектрики	4230.00	27000
3	Балансировка колес	4690.00	23400

При совместном использовании предложений **ORDER BY** и **WHERE**, предложение **ORDER BY** должно следовать после **WHERE**.



```
4 SELECT Title, Cost, DurationInSeconds FROM Service
5 WHERE Cost >= 4000
6 ORDER BY Cost DESC, DurationInSeconds
```

108 %

Результаты Сообщения

	Title	Cost	DurationInSeconds
1	Ремонт компрессора кондиционера	4760.00	9000
2	Замена масла в вариаторе	4720.00	7200
3	Балансировка колес	4690.00	23400

SQL поддерживает весь спектр условных (логических) операций, которые приведены в следующей таблице.

Обозначение	Значение
+	Сложения
-	Вычитание
*	Умножение
/	Деление
%	Остаток от деления
=	Присвоение
&	Побитовая операция «И»
	Побитовая операция «ИЛИ»
^	Побитовая операция «Исключающее ИЛИ»
=	Оператор проверки равенства
>	Больше
<	Меньше
>=	Больше равно
<=	Меньше равно
<>	Не равно
!=	Не равно
!>	Не больше чем
!<	Не меньше чем
ALL	Истина, если каждое сравнение из набора истина
AND	Истина, если два булевых выражения истинны
ANY	Истина, если любое сравнения из набора истинно
BETWEEN	Истина, если операнд находится внутри диапазона
EXISTS	Истина, если подчиненный запрос содержит какие либо строки
IN	Истина, если операнд находится в списке
LIKE	Истина, если операнд соответствует шаблону
NOT	Изменяет значение других булевых операторов на противоположное
OR	Истина, если любая из пар булевых выражений истинна
SOME	Истина, если какое-либо сравнения из набора истинно
+	Конкатенация строк
-	Отрицательное число
~	Дополнение до единицы

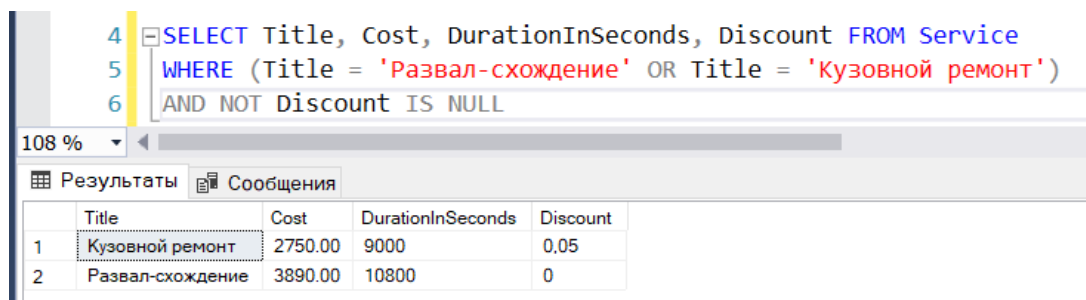
Для поиска диапазона значений можно использовать операцию ***BETWEEN***. Ее синтаксис немного отличается от других операций предложения WHERE, так как для нее требуются два значения: начальное и конечное.

4	=	SELECT Title, Cost, DurationInSeconds FROM Service
5		WHERE Cost BETWEEN 3000 AND 4000
6		

108 %

Результаты		Сообщения	
	Title	Cost	DurationInSeconds
1	Замена сальника привода	3820.00	34200
2	Диагностика инжектора	3390.00	19800
3	Замена кулисы АКПП	3250.00	21600

Предложения WHERE могут содержать любое количество логических операторов AND и OR. Комбинируя их можно создавать сложные фильтры. Однако при комбинировании ключевых слов AND и OR необходимо учитывать, что оператор AND выполняется раньше оператора OR, т.е. имеет более высокий приоритет. Изменить приоритет можно с помощью круглых скобок.



Для фильтрации данных по критерию соответствия определенной символьной строки заданному шаблону используется оператор LIKE. Шаблон может включать обычные символы и символы-шаблоны. Во время сравнения с шаблоном необходимо, чтобы его обычные символы в точности совпадали с символами, указанными в строке. Символы шаблоны могут совпадать с произвольными элементами символьной строки. Использование символов-шаблонов с оператором **LIKE** предоставляет больше возможностей, чем использование обычных операторов сравнения. Шаблон может включать в себя следующие символы-шаблоны

Символ шаблон	Описание	Пример
%	Любое количество символов	Инструкция WHERE Title LIKE 'A%' выполняет поиск и выдает все услуги, названия которых начинаются на букву 'А'.
_	Любой одиночный символ	Инструкция WHERE LastName LIKE '_етров' выполняет поиск и выдает всех клиентов, фамилии которых состоят из шести букв и заканчиваются сочетанием 'етров' (Петров, Ветров и т.п.).
[]	Любой символ, указанный в квадратных скобках	Инструкция WHERE LastName LIKE '[ЛС]омов' выполняет поиск и выдает всех клиентов, фамилии которых заканчиваются на 'омов' и начинаются на любую букву в промежутке от 'Л' до 'С', например Ломов, Ромов, Сомов и т.п.

[^]	Любой символ, кроме перечисленных в квадратных скобках	Инструкция WHERE LastName LIKE 'ив[^а]%' выполняет поиск и выдает всех клиентов, фамилии которых начинаются на 'ив' и третья буква отличается от 'а'.
-----	--	---

Создание вычисляемых полей

Конструкция SELECT кроме имен столбцов таблиц может также включать так называемые вычисляемые поля. В отличие от всех выбранных нами ранее столбцов, вычисляемых полей на самом деле в таблицах базы данных нет. Они создаются "на лету" SQL-оператором SELECT. Рассмотрим следующий пример.

```

4 SELECT LastName + ' ' + FirstName + ' ' + Patronymic AS 'Client'
5 FROM Client

```

108 %

Результаты Сообщения

	Client
1	Голубев Иосиф Тимофеевич
2	Ермакова Алла Мироновна

Здесь создается вычисляемое поле, которому с помощью ключевого слова AS дан псевдоним 'Фамилия и имя клиента'. Оно позволяет объединить (произвести конкатенацию) с помощью оператора + фамилию, пробел и имя клиента в одно поле (столбец).

Еще одним способом использования вычисляемых полей является выполнение математических операций над выбранными данными. Рассмотрим пример.

```

4 SELECT Title, Cost, Cost - (Cost * Discount) AS 'Price'
5 FROM Service

```

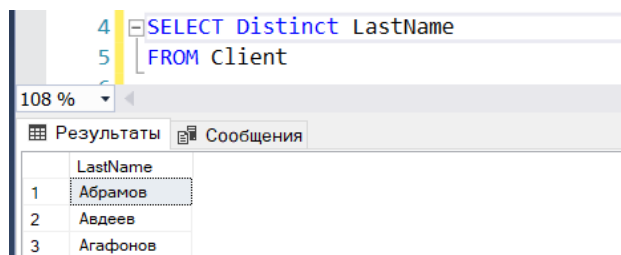
108 %

Результаты Сообщения

	Title	Cost	Price
1	Замена сальника привода	3820.00	3247
2	Ремонт компрессора кондиционера	4760.00	3570

Исключение дублирующих записей

Для исключения из результата выборки повторяющихся строк используется ключевое слово ***DISTINCT***, которое указывается сразу после SELECT. В следующем примере осуществляется вывод всех фамилий клиентов. Даже если среди них есть однофамильцы, каждая фамилия будет выведена только один раз.



ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ:

Сформулируйте на языке SQL запросы на выборку следующих данных:

- Выведите список всех предоставленных услуг за определенный период времени (сентябрь 2019 года), отсортированный по дате оказания услуги.
- Список всех услуг, содержащие слово «замена» с указанием их стоимости.
- Список продаж товара, наименование которого начинается на «Учебник английского языка», с указанием даты продажи, отсортированный по дате продажи и количеству.

Использование агрегатных функций

В SQL определено множество встроенных функций различных категорий, среди которых особое место занимают агрегатные функции, оперирующие значениями столбцов множества строк и возвращающие одно значение. Аргументами агрегатных функций могут быть как столбцы таблиц, так и результаты выражений над ними. Агрегатные функции и сами могут включаться в другие арифметические выражения. В следующей таблице приведены наиболее часто используемые стандартные унарные агрегатные функции.

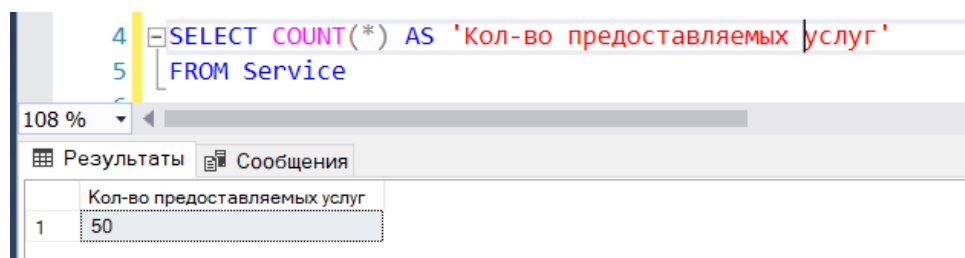
Функция	Возвращаемое значение
COUNT	Количество значений в столбце или строк в таблице
SUM	Сумма
AVG	Среднее
MIN	Минимум
MAX	Максимум

Функция **COUNT** имеет два формата. В первом случае возвращается количество строк входной таблицы, во втором случае — количество значений аргумента во входной таблице:

- COUNT(*)
- COUNT([DISTINCT | ALL] выражение)

Простейший способ использования этой функции — подсчет количества строк в таблице (всех или удовлетворяющих указанному условию). Для этого используется первый вариант синтаксиса.

Запрос: Количество видов продукции, информация о которых имеется в базе данных.



```
4 SELECT COUNT(*) AS 'Кол-во предоставляемых услуг'
5 FROM Service
```

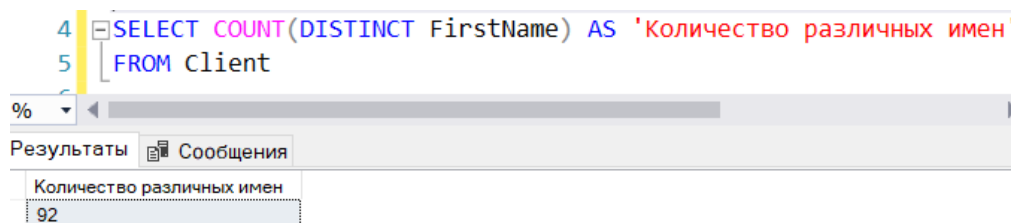
108 %

Результаты Сообщения

	Кол-во предоставляемых услуг
1	50

Во втором варианте синтаксиса функции COUNT в качестве аргумента может быть использовано имя отдельного столбца. В этом случае подсчитывается количество либо всех значений в этом столбце входной таблицы, либо только неповторяющихся (при использовании ключевого слова DISTINCT).

Запрос: Количество различных имен, содержащихся в таблице Client.



```
4 SELECT COUNT(DISTINCT FirstName) AS 'Количество различных имен'
5 FROM client
```

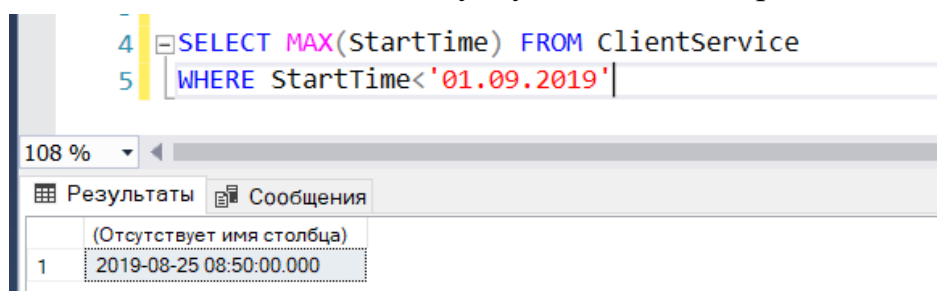
%

Результаты Сообщения

	Количество различных имен
	92

Использование остальных унарных агрегатных функции аналогично COUNT за тем исключением, что для функций MIN и MAX использование ключевых слов DISTINCT и ALL не имеет смысла. С функциями COUNT, MAX и MIN кроме числовых могут использоваться и символьные поля. Если аргумент агрегатной функции не содержит значений, функция COUNT возвращает 0, а все остальные - значение NULL.

Запрос: Дата последней оказанной услуги до 1 сентября 2019 года.



ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ:

Сформулируйте на языке SQL запросы на выборку следующих данных:

- Выведите среднюю продолжительность услуги в минутах;
- Количество различных поставщиков, осуществляющих поставки товаров (из таблицы Product);
- Дату рождения самого молодого и самого взрослого клиента компании;
- Количество проданных товаров за ноябрь 2019 года;
- Выведите количество товаров, стоимость которых более 1500 руб. с учетом скидки;
- Выведите количество клиентов мужского и женского пола;
- Выведите средний возраст клиентов.

Запросы с группировкой строк

Описанные выше агрегатные функции применялись ко всей таблице. Однако часто при создании отчетов появляется необходимость в формировании промежуточных итоговых значений, то есть относящихся к данным не всей таблицы, а ее частей. Для этого предназначена фраза **GROUP BY**. Она позволяет все множество строк таблицы разделить на группы по признаку равенства значений одного или нескольких столбцов (и выражений над ними). Фраза **GROUP BY** должна располагаться вслед за фразой **WHERE** (если она отсутствует, то за фразой **FROM**). При наличии фразы **GROUP BY** фраза **SELECT** применяется к каждой группе, сформированной фразой группировки. В этом случае и действие агрегатных функций, указанных во фразе **SELECT**, будет распространяться не на всю результирующую таблицу, а только на строки в пределах каждой группы.

Каждое выражение в списке фразы SELECT должно принимать единственное значение для группы, то есть оно может быть:

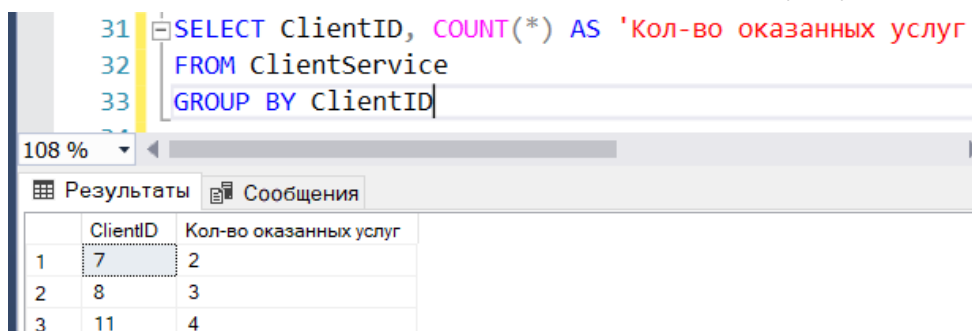
- константой;
- агрегатной функцией, которая оперирует всеми значениями аргумента в пределах группы и агрегирует их в одно значение (например, в сумму);
- выражением, идентичным стоящему во фразе GROUP BY;
- выражением, объединяющим приведенные выше варианты.

Для отбора строк среди полученных групп применяется фраза **HAVING**. Она играет такую же роль для групп, что и фраза WHERE для исходных таблиц, и может использоваться лишь при наличии фразы GROUP BY. В предложении SELECT фразы WHERE, GROUP BY и HAVING обрабатываются в следующем порядке.

1. Фразой WHERE отбираются строки, удовлетворяющие указанному в ней условию;
2. Фраза GROUP BY группирует отобранные строки;
3. Фразой HAVING отбираются группы, удовлетворяющие указанному в ней условию.

Значение условия, указываемого во фразе HAVING, должно быть уникальным для всех строк каждой группы. Поэтому правила использования имен столбцов и агрегатных функций во фразе HAVING такие же, как и для фразы SELECT при наличии фразы GROUP BY. Это значит, что во фразе HAVING в качестве операндов сравнения можно использовать только группируемые столбцы или агрегатные функции.

Запрос: Список клиентов, и количество оказанных им услуг.



```
31 SELECT ClientID, COUNT(*) AS 'Кол-во оказанных услуг'
32 FROM ClientService
33 GROUP BY ClientID
```

	ClientID	Кол-во оказанных услуг
1	7	2
2	8	3
3	11	4

Запрос: Список клиентов, которым оказано более двух услуг.

31	SELECT ClientID, COUNT(*) AS 'Кол-во оказанных услуг'
32	FROM ClientService
33	GROUP BY ClientID
34	HAVING COUNT(*)>2

108 %	
Результаты	Сообщения
	ClientID Кол-во оказанных услуг
1	8 3
2	11 4
3	17 5

ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ:

Сформулируйте на языке SQL запросы на выборку следующих данных:

- Выведите список товаров (ID и дату последней продажи), которые были проданы 2 и более раз за 2019 год.
- Выведите список поставщиков (ID), которые поставляют более одного товара. Отсортируйте по количеству поставляемых товаров в порядке убывания.