

## Урок №4

### Теория баз данных

## Содержание

Функции агрегирования.....	3
Группировка данных. ....	7
Домашнее задание .....	12

## Функции агрегирования.

### Необходимость использования

В SQL, кроме обычных функций, существует ряд специальных функций, которые называют *функциями агрегирования* или группировки. Каждая из этих функций работает с совокупностью значений поля некоторой таблицы, а результатом функций является единственное значение. Именно от принципа их работы и происходит название этих функций. Примеров необходимости их использования можно привести множество. Например, необходимо вывести сумму продажи товаров на каждую дату, общее количество проданных домов, расположенных в отдельных районах или же необходимо определить минимальную и максимальную цены на товары отдельной категории.

Функции агрегации (агрегирования) преимущественно возвращают числовые результаты и обычно используются для расчета определенной статистики. В каждой СУБД поддерживается свой набор таких функций, но ряд из них утверждены стандартом SQL и общие для всех. Итак, к *функциям агрегирования относятся*:

- **COUNT** — количество значений (записей) в поле;
- **SUM** — сумма значений поля;
- **AVG** — среднее арифметическое значений поля;
- **MAX** — максимальное значение в совокупности;
- **MIN** — минимальное значение в совокупности;

- **\* StDev** — среднеквадратическое отклонение значений поля;
  - **\* Var** — дисперсия распределенных значений, содержащихся в указанном поле;
  - **\* First** — первая запись из результирующего набора по указанному полю;
  - **\* Last** — последняя запись из результирующего набора по указанному полю.
- \* — Поддерживаются только Microsoft Access.

Стоит также отметить, что для функций **SUM** и **AVG** поле **ОБЯЗАТЕЛЬНО** должно содержать числовые значения, в других случаях поля могут содержать данные произвольного типа. Поработаем и разберем несколько примеров:

1. Создадим запрос на получение информации об общем количестве товаров, их общую и среднюю стоимость.

```
SELECT COUNT (Product.IdProduct) as [Количество товара],
SUM (Sale.Price*Sale.Quantity) as
[Общая стоимость продаж],
AVG (Sale.Price*Sale.Quantity) AS
[Средняя стоимость продажи]
FROM Product, Sale
WHERE Product.IdProduct=Sale.IdProduct;
```

Результат:

Количество товара	Общая стоимость продаж	Средняя стоимость продаж
15	1057,75	70,5166666666667

2. Вроде бы ничего сложного, но рассмотрим маленький нюанс. Напишем запрос, который позволит подсчитать только общее количество записей в таблице, содержащей информацию о товарах:

```
SELECT COUNT (*) as Количество Записей
FROM Product;
```

Результат:

Количество записей
17

Функция **COUNT** со звездочкой в качестве параметра включает при подсчете как NULL-значения, так и значения, которые повторяются. Допустим, возникла ситуация, когда необходимо вывести список товаров с уникальными названиями (вполне вероятно ситуация, когда товар под названием «Бананы» поставляется несколькими поставщиками и производится несколькими производителями). Для осуществления таких расчетов, согласно стандарту SQL, в теле функции агрегирования, в том числе и **COUNT**, можно использовать ключевое слово **DISTINCT** (разный) с целью подсчета только уникальных ненулевых значений, а также ключевое слово **ALL** (все) для учета при подсчете всех значений поля, кроме NULL. Кстати, **ALL** неявно подставляется по умолчанию. В таком случае наш запрос переписывается

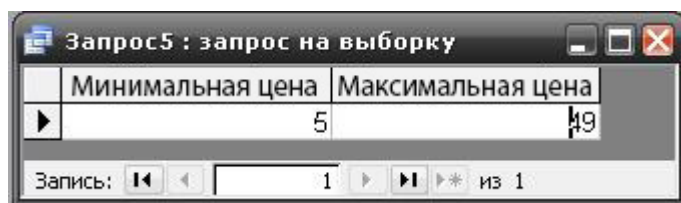
```
SELECT COUNT (DISTINCT Name) as [Количество  
уникальных наименований товаров]  
FROM Product;
```

Но СУБД MS Access не в полной мере поддерживает стандарт ANSI SQL, и поэтому данный запрос вызовет ошибку. Кстати, явное использование ключевого слова ALL в любой функции агрегирования также в MS Access вызовет ошибку.

### 3. Вывести на экран минимальные и максимальные цены на товары категории «Кондитерские»:

```
SELECT MIN(Product.Price) as [Минимальная цена],  
       MAX(Product.Price) as [Максимальная цена]  
FROM Product, Category  
WHERE Product.IdCategory=Category.IdCategory AND  
       Category.Name = 'Кондитерские';
```

Результат:



Минимальная цена	Максимальная цена
5	19

Запись: 1 из 1

## Группировка данных. Операторы GROUP BY и HAVING

Работа с функциями агрегирования отличается от обычной выборки, поскольку их результатом является одно единственное значение. В связи с этим при создании запроса на выборку нельзя одновременно при операторе **SELECT** использовать обычные поля и функции агрегирования.

Для реализации такой возможности используется понятие группировки данных и оператор, который их осуществляет — **GROUP BY**. Указанная конструкция позволяет определять группы, которые обладают общими характеристиками, а затем генерировать статистику для каждой группы (с использованием необходимой функции агрегирования). В результате СУБД будет возвращать столько строк, сколько групп вы определите.

Чтобы лучше разобраться с работой данного оператора рассмотрим несколько примеров:

### 1. Вывести названия товаров и наименьшие цены их продажи:

```
SELECT Product.Name, MIN (Sale.Price) & 'грн.'  
       as [Минимальная цена]  
FROM Product, Sale  
WHERE Product.IdProduct=Sale.IdProduct  
GROUP BY Product.Name;
```

Скриншоты из базы данных. Слева: таблица «Sale». Справа: результат запроса «Запрос5: запрос на выборку».

IdSale	IdProduct	Price	Quantity	DateSale
15	Апельсины «Наколотые»	7	0,7	27.07/2009
9	Бананы	3	9,5	05/06/2009
8	Бананы	2	8,5	04/06/2009
5	Водка «Чиполино»	15	5	02.02/2008
4	Водка «Чиполино»	13	7	
7	Молоко «Успевайка»	5	5	
6	Молоко «Успевайка»	4,5	10	

Name	Минимальная цена
Апельсины «Наколотые»	7 грн.
Бананы	2 грн.
Водка «Чиполино»	13 грн.
Молоко «Успевайка»	4,5 грн.
Моршинская 0,5л	2,5 грн.

Поле «Name» из таблицы «Product» в списке **SELECT** является полем группировки. СУБД группирует записи по значению в данном поле, а затем ищет минимальное значение цены для каждой отдельной группы, а затем выводится результат.

При работе с оператором **GROUP BY** следует помнить о том, что *все поля группировки, которые вы задаете в списке SELECT, должны быть включены и в список GROUP BY*. Если полей несколько, то их названия разделяются запятыми.

- Вывести на экран категории продуктов, производителей товаров этих категорий и значение средней цены в разрезе этих групп (категорий и производителей). Отсортировать выборку по названию производителя.

```
SELECT Producer.Name as [Производитель],
       Category.Name as [Категория],
       AVG(Product.Price) & ' грн.' as [Средняя цена]
FROM Product, Category, Producer
WHERE Product.IdCategory=Category.IdCategory AND
       Product.IdProducer=Producer.IdProducer
GROUP BY Producer.Name, Category.Name
ORDER BY 1;
```

Результат:

Запрос5: запрос на выборку

Производитель	Категория	Средняя цена
Banana Republic	Фрукты	9 грн.
АТ «Русская водка»	Ликеро-водочные	66,5 грн.
BAT «Конфеты»	Кондитерские	49 грн.
BAT «Росинка»	Бакалея	3 грн.
BAT «Сальце Украины»	Колбасные	54 грн.
BAT «Сальце Украины»	Мясные	50 грн.

- А теперь выведем информацию о товарах, минимальная цена продажи которых была больше 10 грн.:

```
SELECT Product.Name, MIN(Sale.Price)&' грн.',
       as [Минимальная цена]
FROM Product, Sale
WHERE Product.IdProduct=Sale.IdProduct
GROUP BY Product.Name
HAVING MIN(Sale.Price) > 10;
```

Результат:

Скриншоты из базы данных. Слева: таблица «Sale». Справа: результат запроса «Запрос6: запрос на выборку».

IdSale	IdProduct	Price	Quantity	DateSale
3	Фарш «315км/час»	50,10р.	10	12.05.2009
1	Фарш «315км/час»	50,10р.	1	12.02.2008
2	Конфеты «Радости у козлика»	50,00р.	2	12.02.2008
12	Конфеты «Крабик»	15,50р.	0	25.05.2009
5	Водка «Чиполино»	15,00р.		
4	Водка «Чиполино»	13,00р.		
15	Апельсины «Наколотые»	7,00р.		
11	Сухари «Дубовые дровишки»	5,50р.		
10	Сухари «Дубовые дровишки»	5,50р.		

Name	Минимальная цена
Водка «Чиполино»	13 грн.
Фарш «315км/час»	50,1 грн.
Конфеты «Крабик»	15,5 грн.
Конфеты «Радости у козлика»	50 грн.

Как Вы заметили, в предыдущем запросе использовался новый оператор **HAVING**. Он позволяет определить критерий, согласно которому определенные группы не включаются в результирующую выборку, аналогично тому, как это де-

дает оператор **WHERE** для отдельных записей. Фактически, оператор **HAVING** позволяет наложить условие на группы, поэтому он размещается после оператора **GROUP BY** и указывается только тогда, когда существует последний.

По сути, условие можно определять и в операторе **WHERE**, указывая на критерий отбора каждой отдельной записи, входящей в состав групп, но это делать неуместно. Во-первых, падает быстродействие; во-вторых, указывая условие в операторе **WHERE**, вы сразу исключаете отобранные строки из рассмотрения любой операции группировки. Оператор **HAVING**, наоборот, начинает действовать только после того, как группы сформированы, определяя, какие именно группы будут отражены в результате.

Как правило, оператор **HAVING** содержит функции агрегирования, которые указаны в списке **SELECT**, но это необязательно.

**4. Вывести количество товаров каждой категории, средняя цена поставки которых более 20 грн.**

```
SELECT Category.Name as [Категория],
       COUNT(Product.IdProduct) as [Количество]
FROM Product, Delivery, Category
WHERE Product.IdCategory=Category.IdCategory AND
       Product.IdProduct=Delivery.IdProduct
GROUP BY Category.Name
HAVING AVG(Delivery.Price) > 20;
```

**5. Вывести категории, товары, которые к ним относятся и общую сумму их продажи. Условие: товары только категорий «Кондитерские» и «Фрукты».**

```
SELECT Product.Name as [Товар],
       Category.Name as [Категория],
       Format (SUM(Sale.Price), '## ###.00 грн.') as
       [Сумма продаж]
FROM Product, Sale, Category
WHERE Product.IdProduct=Sale.IdProduct AND
       Product.IdCategory=Category.IdCategory
GROUP BY Product.Name, Category.Name
HAVING Category.Name IN ('Кондитерские', 'Фрукты');
```



# Домашнее задание

1. Вывести общее количество товаров, максимальную и среднюю цену их продажи.
2. Вывести на экран количество новых товаров в магазине, то есть товаров, которые были поставлены в течение последней недели (7 дней).
3. Подсчитать общее количество товаров двух производителей, например ЧП Иванов и ЧП Воробей.
4. Вывести информацию о наименьшей стоимости поставки для каждого поставщика товаров за последний месяц (для получения информации о последнем месяце воспользуйтесь стандартными функциями) в евро. Отсортировать выборку по названию поставщика в возрастающем порядке.
5. Подсчитать и вывести информацию об общем количестве продаж на каждый день в пределах 01/01/2009 по текущий день и вывести их в убывающем порядке количества продаж.
6. Вывести названия товаров производителя ЧП Иванов, которые поставлялись более чем двумя поставщиками.
7. Вывести информацию о производителях, страны их расположения, количество товаров их производства, имеющиеся в магазине. Условие: общая стоимость продажи товаров каждого производителя должна быть в пределах от 100 до 200 грн. Отсортировать выборку

по стране расположения в возрастающем порядке и количестве товаров производителей в ниспадающем.

8. Вычислить и вывести количество товаров, каждой категории, которые необходимо списать. К списанию подлежат товары, которые есть в наличии в магазине и не продавались с даты их поставки в течение 3 месяцев.
9. Вывести на экран количество имеющегося товара по каждой категории, при этом учитывать только товары, стоимость поставки которых превышала 300 грн. Выведенная информация должна касаться только трех поставщиков: ООО «Титаник», ООО Ваня, ОАО «Быстрый ветер».
10. Показать категорию, товаров которой в магазине находится меньше всего.