



# Урок 4

## План заняття:

1. Функції агрегування. Необхідність використання
2. Групування даних. Оператори GROUP BY та HAVING
3. Домашнє завдання

## 1. Функції агрегування. Необхідність використання

В SQL, крім звичайних функцій, існує ряд спеціальних функцій, які називають **функціями агрегування** або **групування**. Кожна з цих функцій працює з сукупністю значень поля деякої таблиці, а результатом функцій є єдине значення. Саме від принципу їх роботи і походить назва цих функцій. Прикладів необхідності їх використання можна привести безліч. Наприклад, необхідно вивести суму продажу товарів на кожну дату, загальну кількість проданих будинків, розміщених в окремих районах або ж необхідно визначити мінімальну і максимальну ціни на товари окремої категорії.

Функції агрегації (агрегування) переважно повертають числові результати і зазвичай використовуються для розрахунку певної статистики. В кожній СУБД підтримується свій набір таких функцій, але ряд з них затверджені стандартом SQL та спільні для всіх. Отже, до **функцій агрегування належать**:

- **COUNT** – кількість значень (записів) в полі;
- **SUM** - сума значень поля;
- **AVG** – середнє арифметичне значень поля;
- **MAX** – максимальне значення з сукупності;
- **MIN** – мінімальне значення в сукупності;
- **\*StDev** – середньоквадратичного відхилення значень поля;
- **\*Var** – дисперсія роз приділених значень, які містяться у вказаному полі;
- **\*First** – перший запис з результуючого набору по вказаному полю;
- **\*Last** – останній запис з результуючого набору по вказаному полю.

\* - підтримуються лише Microsoft Access.

Варто також відмітити, що для функцій SUM і AVG поле **ОБОВ'ЯЗКОВО** повинно містити числові значення, в інших випадках поле може містити дані довільного типу. Попрацюємо та розберемо кілька прикладів:

1. Створимо запит на отримання інформації про загальну кількість товарів, їх загальну та середню вартість.

```
SELECT COUNT(Product.IdProduct) as [Кількість товару],
       SUM(Sale.Price*Sale.Quantity) as [Загальна вартість продаж],
       AVG(Sale.Price*Sale.Quantity) AS [Середня вартість продаж]
FROM Product, Sale
WHERE Product.IdProduct=Sale.IdProduct;
```

Результат:

	Кількість товару	Загальна вартість продаж	Середня вартість продаж
▶	15	1057,75	70,5166666666667

Запись: 1 из 1

2. Ніби нічого складного, але розглянемо маленький нюанс. Напишемо запит, що дозволить підрахувати лише загальну кількість записів в таблиці, що містить інформацію про товари:

```
SELECT COUNT(*) as КількістьЗаписів
FROM Product;
```

Результат:

	КількістьЗаписів
▶	17

Запись: 1

Функція COUNT з зірочкою в якості параметра включає при підрахунку як NULL-значення, так і значення, що повторюються. Припустимо, виникла ситуація, коли необхідно вивести список товарів з унікальними назвами (цілком



імовірна ситуація, коли товар з назвою «Банани» поставляється кількома постачальниками і виробляється кількома виробниками). Для здійснення таких підрахунків згідно стандарту SQL в тілі функцій агрегування, зокрема і COUNT можна використовувати ключове слово **DISTINCT** (різний) з метою підрахунку лише унікальних ненулевих значень, а також ключове слово **ALL** (всі) для врахування при підрахунку всіх значень поля, крім NULL. Доречі, ALL неявно підставляється по замовчуванню. В такому разі наш запит переписеться:

```
SELECT COUNT(DISTINCT Name) as [Кількість унікальних назв товарів]
FROM Product;
```

Але, СУБД MS Access не в повній мірі підтримує стандарт ANSI SQL, і тому даний запит викличе помилку. Доречі, явне використання ключового слова ALL в будь-якій функції агрегування також в MS Access викличе помилку.

3. Вивести на екран мінімальну та максимальну ціни на товари категорії «Кондитерські»:

```
SELECT MIN(Product.Price) as [Мінімальна ціна],
       MAX(Product.Price) as [Максимальна ціна]
FROM Product, Category
WHERE Product.IdCategory=Category.IdCategory AND Category.Name = 'Кондитерські';
```

Результат:

Мінімальна ціна	Максимальна ціна
5	19

## 2. Групування даних. Оператори GROUP BY та HAVING

Робота з функціями агрегування відрізняється від звичайної вибірки, оскільки їх результатом є одне єдине значення. В зв'язку з цим при створенні запиту на вибірку неможна одночасно при операторі SELECT використовувати звичайні поля та функції агрегування.

Для реалізації такої можливості використовується поняття групування даних та оператор, що його здійснює - **GROUP BY**. Вказана конструкція дозволяє визначати групи, які володіють спільними характеристиками, а потім генерувати статистику для кожної групи (з використанням необхідної функції агрегування). В результаті СУБД буде повертати стільки рядків, скільки груп ви визначите.

Щоб краще розібратись з роботою даного оператора розглянемо кілька прикладів:

1. Вивести назви товарів та найменші ціни їх продажу:

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

Результат:

Name	Мінімальна ціна
Апельсини "Наколоті"	7 грн.
Банани	2 грн.
Горілка "Чіполіно"	13 грн.
Молоко "Успевайка"	4,5 грн.
Моршинська 0,5л	2,5 грн.
Сухарі "Пісочні"	5,50 грн.

Поле «Name» з таблиці «Product» в списку SELECT являється полем групування. СУБД групує записи по значенню в даному полі, а потім шукає мінімальне значення ціни для кожної окремої групи, а потім виводиться результат.

При роботі з оператором GROUP BY слід пам'ятати про те, що **всі поля групування, які ви задасте в списку SELECT, повинні бути включені і в список GROUP BY**. Якщо полів кілька, тоїх назви розділяються комами.



2. Вивести на екран категорії продуктів, виробників товарів цих категорій та значення середньої ціни в розрізі цих груп (категорій та виробників). Відсортувати вибірку по назві виробника.

```
SELECT Producer.Name as [Виробник], Category.Name as [Категорія],
       AVG(Product.Price) & ' грн.' as [Середня ціна]
FROM Product, Category, Producer
WHERE Product.IdCategory=Category.IdCategory AND Product.IdProducer=Producer.IdProducer
GROUP BY Producer.Name, Category.Name
ORDER BY 1;
```

Результат:

Виробник	Категорія	Середня ціна
Banana Republica	Фрукти	9 грн.
АТ "Русская водка"	Лікери-горілчані	66,5 грн.
ВАТ "Конфеті"	Кондитерські	49 грн.
ВАТ "Росинка"	Бакалія	3 грн.
ВАТ "Сольце України"	Ковбасні	54 грн.
ВАТ "Сольце України"	Мясні	50 грн.
ЗДТ "ППС"	Мясні	42 грн.

3. А тепер виведемо інформацію про товари, мінімальна ціна продажу яких була більше 10 грн.:

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

Результат:

IdSale	IdProduct	Price	Quantity	DateSale
3	Фарш "З15км/год"	50,10р.	10	12.05.2009
1	Фарш "З15км/год"	50,10р.	1	12.02.2008
2	Цукерки "Радощі у козлика"	50,00р.	2	12.02.2008
12	Цукерки "Крабїки"	15,50р.	0	25.05.2009
5	Горїлка "Чіполіно"	15,00р.		
4	Горїлка "Чіполіно"	13,00р.		
15	Апельсини "Наколоті"	7,00р.		
11	Сухаріки "Дубовые дровишки"	5,50р.		
10	Сухаріки "Дубовые дровишки"	5,50р.		

Name	Мінімальна ціна
Горїлка "Чіполіно"	13 грн.
Фарш "З15км/год"	50,1 грн.
Цукерки "Крабїки"	15,5 грн.
Цукерки "Радощі у козлика"	50 грн.

Як Ви помітили, в попередньому запиті використовувався новий оператор **HAVING**. Він дозволяє визначити критерій, згідно якого певні групи не включаються в результуючу вибірку, аналогічно тому, як це робить оператор WHERE для окремих записів. Фактично, оператор HAVING дозволяє накласти умову на групи, тому він розміщується після оператора GROUP BY і вказується лише тоді, коли існує останній.

По суті, умову можна визначати і в операторі WHERE, вказуючи на критерій відбору кожного окремого запису, який входить до складу груп, але це робити недоречно. По-перше, падає швидкодія; по-друге, вказуючи умову в операторі WHERE, ви одразу виключаєте відібрані рядки з розгляду будь-якої операції групування. Оператор HAVING, навпаки, починає діяти лише після того, як групи сформовані, визначаючи, які саме групи будуть відображені в результаті.

Як правило, оператор HAVING містить функції агрегування, які вказані в списку SELECT, але це необов'язково.

4. Вивести кількість товарів кожної категорії, середня ціна поставки яких більше 20 грн.

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



5. Вивести категорії, товари, що до них належать та загальну суму їх продажу. Умова: товари лише категорій «Кондитерські» та «Фрукти».

```
SELECT Product.Name as [Товар], Category.Name as [Категорія],  
       Format(SUM(Sale.Price), '## ###.00 грн.') as [Сума 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 ('Кондитерські', 'Фрукти');
```

### 3. Домашнє завдання

1. Вивести загальну кількість товарів, максимальну та середню ціну їх продажу.
2. Вивести на екран кількість нових товарів в магазині, тобто товарів, які були поставлені на протязі останнього тижня (7 днів).
3. Підрахувати загальну кількість товарів двох виробників, наприклад ПП Іванов і ПП Воробей.
4. Вивести інформацію про найменшу вартість поставки для кожного постачальника товарів за останній місяць (для отримання інформації про останній місяць скористайтесь стандартними функціями) в євро. Відсортувати вибірку по назві постачальника в зростаючому порядку.
5. Підрахувати та вивести інформацію про загальну кількість продажу на кожен день в межах 01/01/2009 по поточний день і вивести їх в спадаючому порядку кількості продаж.
6. Вивести назви товарів виробника ПП Іванов, що поставлялись більш, ніж двома постачальниками.
7. Вивести інформацію про виробників, країни їх розташування, кількість товарів їх виробництва, які наявні в магазині. Умова: загальна вартість продажу товарів кожного виробника повинна бути в межах від 100 до 200 грн. Відсортувати вибірку по країні розташування в зростаючому порядку та кількості товарів виробників в спадаючому.
8. Вирахувати та вивести кількість товарів, кожної категорії, які необхідно списати. До списання підлягають товари, які наявні в магазині і не продавались з дати їх поставки на протязі 3 місяців.
9. Показати на екран кількість наявного товару по кожній категорії, при цьому враховувати лише товари, вартість поставки яких перевищувала 300 грн. Виведена інформація повинна стосуватись лише трьох постачальників: ТзОВ «Титанік», ПП Ваня, ВАТ «Швидкий вітер».
10. Показати категорію, товарів якої в магазині знаходиться найменше.