* **Агрегатные функции SQL:** SUM, MIN, MAX, AVG и COUNT.
* COUNT (\*) - Пока \* является единственым аргументом который включает NULL значения, и он используется только с COUNT; функции отличные от COUNT игнорируют значения NULL в любом случае
* DISTINCT работает только с одним аргументом (полем таблицы)
* GROUP BY позволяет определять подмножество значений в особом поле в терминах другого поля, и применять функцию агрегата к подмножеству.

SELECT snum, MAX (amt)

FROM Orders GROUP BY snum;

SELECT snum, odate, MAX (amt)

FROM Orders GROUP BY snum, odate;

* Предложение HAVING определяет критерии используемые чтобы удалять определенные группы из вывода, точно также как предложение WHERE делает это для индивидуальных строк.

SELECT snum, odate, **MAX ((amt))**

FROM Orders

GROUP BY snum, odate

HAVING **MAX ((amt))** > 3000.00;

**HAVING может использовать только аргументы, которые имеют одно значение на группу вывода**

Для решения данной задачи, так же можно было воспользоваться подзапросом

**Оператор HAVING используется для обработки условий с агрегатными функциями, если использовать условие WHERE с агрегатной функией, будет ошибка и отбор не сработает.**

Если необходимо сделать отбор с группировкой с условием по одиночному полю, а не агрегатору, необходимо использовать следующую конструкцию с использованием WHERE

SELECT snum, MAX (amt)

FROM Orders

WHERE odate=10/03/1990

GROUP BY snum;

HAVING может использовать только аргументы которые имеют одно значение на группу вывода. Практически, ссылки на агрегатные функции - наиболее общие, но и поля, выбранные с помощью GROUP BY также допустимы.

SELECT snum, MAX (amt) FROM Orders

GROUP BY snum HAVING snum IN (1002, 1007 );

* ORDER BY. Столбцы, которые упорядочиваются(сортируются), должны быть указаны в выборе SELECT.

ORDER BY может использоваться с GROUP BY для упорядочения групп. Если это так, то ORDER BY всегда приходит последним.

SELECT snum, odate, MAX (amt) FROM Orders

GROUP BY snum, odate ORDER BY snum

* Вап
* Вап
* Вап