**Практическая работа № 12**

**Использование агрегатных функций в подзапросах**

**Краткие теоретические сведения**

Вы хотите увидеть все Заказы, имеющие сумму приобретений выше средней на 4-е Октября

SELECT \*  
FROM Orders  
WHERE amt > (SELECT AVG (amt)  
 FROM Orders  
 WHERE odate = 10/04/1990);

=============== SQL Execution Log ==============  
| SELECT \* |  
| FROM Orders |  
| WHERE amt > |  
| (SELECT AVG (amt) |  
| FROM Orders |  
| WHERE odate = 01/04/1990); |  
| =============================================== |  
| onum amt odate cnum snum |  
| ----- -------- ---------- ----- ----- |  
| 3002 1900.10 10/03/1990 2007 1004 |  
| 3005 5160,45 10/03/1990 2003 1002 |  
| 3006 1098.19 10/03/1990 2008 1007 |  
| 3009 1713.23 10/04/1990 2002 1003 |  
| 3008 4723.00 10/05/1990 2006 1001 |  
| 3010 1309.95 10/06/1990 2004 1002 |  
| 3011 9891.88 10/06/1990 2006 1001 |  
=================================================

Средняя сумма приобретений на 4 Октября — 1788.98 (1713.23 + 75.75) делится пополам, что в целом равняется = 894.49. Все строки со значением в поле amt выше этого — являются выбранными.

Имейте в виду, что сгруппированные агрегатные функции, которые являются агрегатными функциями, определенными в терминах предложения GROUP BY, могут производить многочисленные значения. Они, следовательно, не позволительны в подзапросах такого характера. Даже если GROUP BY и HAVING используются таким способом, что только одна группа выводится с помощью подзапроса, команда будет отклонена в принципе. Вы должны использовать одиночную агрегатную функцию с предложением WHERE, что устранит нежелательные группы. Например, следующий запрос, который должен найти среднее значение комиссионных продавца в Лондоне

SELECT AVG (comm)  
FROM Salespeople  
GROUP BY city  
HAVING city = 'London;

не может использоваться в подзапросе! Во всяком случае, это не лучший способ формировать запрос. Другим способом может быть

SELECT AVG (comm)  
FROM Salespeople  
WHERE city = 'London';

**Использование подзапросов, которые выдают много строк с помощью оператора IN**

Вы можете использовать подзапросы, которые производят любое число строк, если вы используете специальный оператор IN (операторы BETWEEN, LIKE, и IS NULL не могут использоваться с подзапросами). Например, мы можем найти все атрибуты таблицы Заказов для продавца в Лондоне

SELECT \*  
FROM Orders  
WHERE snum IN (SELECT snum  
 FROM Salespeople  
 WHERE city = 'London');

=============== SQL Execution Log ==============  
| SELECT \* |  
| FROM Orders |  
| WHERE snum IN |  
| (SELECT snum |  
| FROM Salespeople |  
| WHERE city = 'London'); |  
| =============================================== |  
| onum amt odate cnum snum |  
| ----- -------- ---------- ----- ------ |  
| 3003 767.19 10/03/1990 2001 1001 |  
| 3002 1900.10 10/03/1990 2007 1004 |  
| 3006 1098.19 10/03/1990 2008 1007 |  
| 3008 4723.00 10/05/1990 2006 1001 |  
| 3011 9891.88 10/06/1990 2006 1001 |  
 =================================================

**Подзапросы выбирают одиночные столбцы**

Смысл всех подзапросов обсужденных в этой главе тот, что все они выбирают одиночный столбец. Это обязательно, поскольку выбранный вывод сравнивается с одиночным значением. Подтверждением этому то, что SELECT \* не может использоваться в подзапросе.

**Использование выражений в подзапросах**

Вы можете использовать выражение, основанное на столбце, а не просто сам столбец, в предложении SELECT подзапроса. Это может быть выполнено или с помощью реляционных операторов, или с IN. Например, следующий запрос использует реляционный оператор = (вывод показывается в рисунке):

SELECT \*  
FROM Customers  
WHERE cnum = (SELECT snum + 1000  
 FROM Salespeople  
 WHERE sname = 'Serres');

=============== SQL Execution Log ============  
| SELECT \* |  
| FROM Customers |  
| WHERE cnum = |  
| (SELECT snum + 1000 |  
| WHERE Salespeople |  
| WHERE sname = 'Serres' |  
| ============================================= |  
| cnum cname city rating snum |  
| ----- -------- ---- ------ ----- |  
| 2002 Giovanni Rome 200 1003 |  
 ===============================================

**Подзапросы в предложении HAVING**

Вы можете также использовать подзапросы внутри предложения HAVING. Эти подзапросы могут использовать свои собственные агрегатные функции, если они не производят многочисленных значений или использовать GROUP BY или HAVING. Следующий запрос является этому примером (вывод показывается в рисунке):

SELECT rating, COUNT (DISTINCT cnum)  
FROM Customers  
GROUP BY rating  
HAVING rating > (SELECT AVG (rating)  
 FROM Customers  
 WHERE city = 'San Jose');

=============== SQL Execution Log ============  
| SELECT rating,count (DISTINCT cnum) |  
| FROM Customers |  
| GROUP BY rating |  
| HAVING rating > |  
| (SELECT AVG (rating) |  
| FROM Customers |  
| WHERE city = 'San Jose' |  
|===============================================|  
| rating |  
| -------- -------- |  
| 300 2 |  
 ===============================================

Эта команда подсчитывает заказчиков с оценками выше среднего в San Jose. Так как имеются другие оценки отличные от 300, они должны быть выведены с числом номеров заказчиков, которые имели эту оценку.

**Задание**

**Теоретическая часть**

1. Для чего используют агрегатные функции в подзапросах?

**Практическая часть**

1. Напишите запрос, который вывел бы имена Заказчиков, их рейтинг и среднюю стоимость заказа.
2. Напишите запрос, который вывел бы имена Заказчиков из Рима, рейтинг которых ниже среднего.
3. Напишите запрос, который выводит комиссионные (comm) всех Продавцов, обслуживавших Заказчиков в Париже.
4. Написать запрос, который подсчитывает количество Заказов с суммой приобретения ниже среднего, совершенных 10/03/1990.
5. Написать запрос, который выводит Продавца, уникальный номер которого на 1000 меньше уникального номера Заказчика, проживающего в Берлине.
6. Напишите запрос, который бы выбрал общую сумму всех приобретений в Заказах для каждого продавца, у которого эта общая сумма больше, чем сумма наибольшего Заказа в таблице.

Покажите работу преподавателю.

**ВНИМАНИЕ** В тетради Вы должны законспектировать все созданные и выполненные запросы по заданию.