Групповые, или агрегатные, функции используются для получения итоговой, сводной

информации на основе значений, хранящихся в столбце таблицы. В этом разделе вы узнаете

об этих функциях, а также об особенностях синтаксиса запросов, использующих эти функ-

ции.

***SUM ()***

Данная функция возвращает сумму значений в столбце. Неопределенные значения при

этом не учитываются. Если запросом не найдено ни одной строки или все значения в столбце

равны NULL, то функция возвращает значение NULL.

Например, запрос

***SELECT*** ***SUM***(rating) ***FROM*** Customers;

возвращает сумму рейтингов клиентов – величину, полученную при сложении значе-

ний 1000 + 1500 + 1000.

Исключить повторяющиеся значения при подсчете суммы можно с помощью параме-

тра DISTINCT. Если указан этот параметр, то каждое значение столбца будет учтено в сумме

только один раз, даже если в столбце оно встречается несколько раз.

Например, запрос

***SELECT*** ***SUM***(***DISTINCT*** rating) ***FROM*** Customers;

подсчитывает сумму только несовпадающих рейтингов.

Например, запрос

***SELECT*** ***SUM***(amount) ***FROM*** Orders ***WHERE*** customer\_id = 533;

вычисляет общую сумму заказов клиента с идентификатором 533

***AVG()***

Данная функция возвращает среднее арифметическое значений в столбце (сумму зна-

чений, деленную на количество значений). Неопределенные значения при этом не учитыва-

ются. Если в запросе вы укажете какое-либо условие отбора, то суммирование произойдет

только по тем строкам, которые удовлетворяют условию отбора. Если запросом не найдено

ни одной строки или все значения в столбце равны NULL, то функция возвращает значение

NULL.

Например, запрос

***SELECT*** ***AVG***(rating) ***FROM*** Customers;

возвращает средний рейтинг клиентов – величину (1000 + 1500 + 1000) /3.

Исключить повторяющиеся значения при подсчете среднего можно с помощью пара-

метра DISTINCT.

Например, запрос

***SELECT*** ***AVG*** (***DISTINCT*** rating) ***FROM*** Customers;

подсчитывает среднее только несовпадающих рейтингов – величину (1000 + + 1500) /

2; еще одно значение 1000, имеющееся в столбце rating, запросом игнорируется.

Функцию AVG () можно использовать для отбора тех значений, которые больше сред-

него, или тех, которые меньше среднего.

Например, запрос

***SELECT*** \* ***FROM*** Customers

***WHERE*** rating > (***SELECT*** ***AVG***(rating) ***FROM*** Customers);

выводит информацию о клиентах, чей рейтинг выше среднего.

***MAX ()***

Данная функция возвращает максимальное значение в столбце. Если в запросе вы ука-

жете какое-либо условие отбора, то максимальное значение выбирается из строк, удовлетво-

ряющих условию отбора. Если запросом не найдено ни одной строки или все значения в

столбце равны NULL, то функция возвращает значение NULL.

Например, запрос

***SELECT*** ***MAX***(rating) ***FROM*** Customers;

возвращает наибольший из рейтингов клиентов – 1500.

Функцию MAX() можно использовать для поиска строк, в которых достигается мак-

симальное значение столбца.

Например, запрос

***SELECT*** \* ***FROM*** Customers

***WHERE*** rating = (***SELECT*** ***MAX***(rating) ***FROM*** Customers);

выводит информацию о клиентах, чей рейтинг равен максимальному.

***MIN ()***

Данная функция возвращает минимальное значение в столбце. Если в запросе вы ука-

жете какое-либо условие отбора, то минимальное значение выбирается из строк, удовлетво-

ряющих условию отбора. Если запросом не найдено ни одной строки или все значения в

столбце равны NULL, то функция возвращает значение NULL.

Например, запрос

***SELECT*** ***MIN***(rating) ***FROM*** Customers;

возвращает наименьший из рейтингов клиентов – 1000.

Функцию MIN() можно использовать для поиска строк, в которых достигается мини-

мальное значение столбца.

Например, запрос

***SELECT*** \* ***FROM*** Customers

***WHERE*** rating = (***SELECT*** ***MIN***(rating) ***FROM*** Customers);

выводит информацию о клиентах, чей рейтинг равен минимальному.

***COUNT ()***

Данная функция возвращает количество отличных от NULL значений, содержащихся

в столбце. Если в запросе вы укажете какое-либо условие отбора, то в подсчете участвуют

только строки, удовлетворяющие условию отбора. Если не найдено ни одного отличного от

NULL значения, то функция возвращает значение 0.

Например, запрос

***SELECT*** ***COUNT***(rating) ***FROM*** Customers;

возвращает количество отличных от NULL значений в столбце rating таблицы

Customers.

Параметр DISTINCT позволяет подсчитать количество различных (уникальных) зна-

чений в столбце (при этом неопределенные значения также игнорируются).

Например, запрос

***SELECT*** ***COUNT***(***DISTINCT*** rating) ***FROM*** Customers;

подсчитывает количество различных значений рейтинга в таблице Customers.

В таблице есть две строки с одинаковым рейтингом – 1000, поэтому результат под-

счета будет меньше, чем в предыдущем запросе.

Если в качестве аргумента функции COUNT() указать не имя столбца, а звездочку,

то функция возвращает общее число строк, удовлетворяющих условию отбора, включая

строки, содержащие неопределенные значения. Так, если столбец rating содержит неопре-

деленные значения, то значение, выводимое запросом

***SELECT*** ***COUNT***(\*) ***FROM*** Customers;

Функцию COUNT() можно использовать для отбора тех строк родительской таблицы,

с которыми связано заданное количество строк дочерней таблицы.

Например, запрос

***SELECT*** \* ***FROM*** Customers

***WHERE*** 2 < = (***SELECT*** ***COUNT***(\*) ***FROM*** Orders

***WHERE*** Customers.id = customer\_id);

выводит список клиентов, сделавших не менее двух заказов.

Для каждого клиента вложенный запрос выдает количество заказов этого кли-

ента, и если это количество не меньше двух, то текущая запись о клиенте включается в

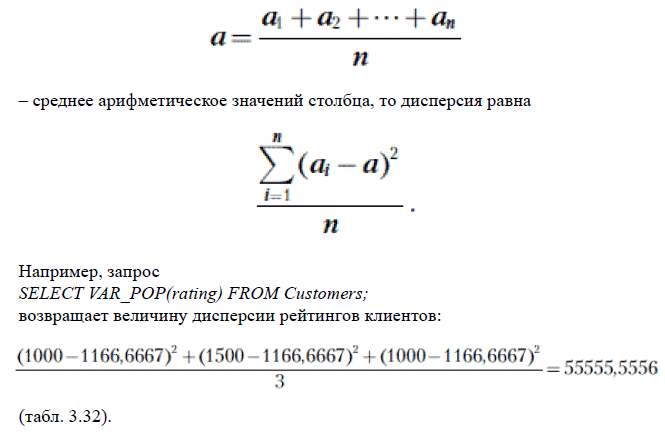
результат, выводимый внешним запросом.

***Функции VAR\_POP(), VARIANCE(), VAR\_SAMP(),***

***STDDEV\_POP(), STD(), STDDEV() и STDDEV\_SAMP()***

***Функция VAR\_POP ()*** вычисляет дисперсию значений столбца. Дисперсия характери-

зует колебание значений от среднего. Если a1, a2,…, an – значения столбца,



Функция ***VARIANCE ()*** является синонимом функции VAR\_POP ().

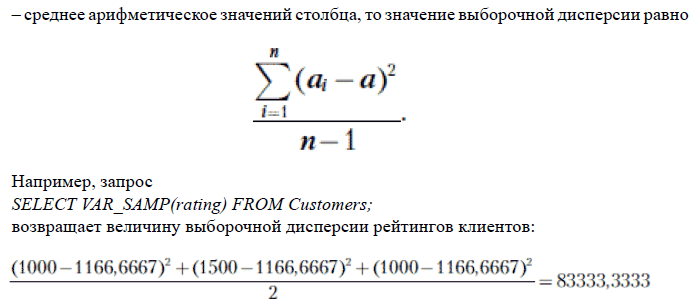
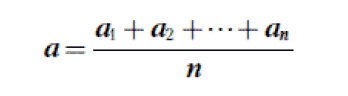
***Функция*** ***VAR\_SAMP ()*** возвращает величину выборочной, или несмещенной, диспе-

рсии (в математической статистике выборочная дисперсия является оценкой дисперсии всей

изучаемой совокупности значений, при этом значения, по которым вычисляется несмещен-

ная дисперсия, рассматриваются как выборка из изучаемой совокупности). Если a1, a2,… an

– значения столбца,



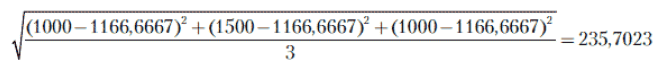
***Функция STDDEV\_POP ()*** вычисляет среднеквадратичное отклонение значений

столбца, которое является квадратным корнем из дисперсии.

Например, запрос

***SELECT*** ***STDDEV***\_***POP***(rating) ***FROM*** Customers;

возвращает величину



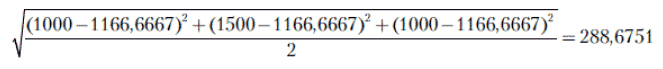
Функции STD() и STDDEV() являются синонимами функции STDDEV\_POP().

***Функция STDDEV\_SAMP ()*** вычисляет квадратный корень из выборочной дисперсии.

Например, запрос

***SELECT*** ***STDDEV***\_***SAMP***(rating) ***FROM*** Customers;

возвращает величину



При вычислении всех вышеперечисленных функций неопределенные значения не учи-

тываются. Если в запросе вы укажете какое-либо условие отбора, то в вычислениях уча-

ствуют только те строки, которые удовлетворяют условию отбора. Если запросом не найдено

ни одной строки или все значения в столбце равны NULL, то все эти функции возвращают

значение NULL.

***GROUP\_CONCAT ()***

Функция GROUP\_CONCAT () объединяет в одну строку значения столбца. При этом

неопределенные значения не учитываются. Если в запросе вы укажете какое-либо усло-

вие отбора, то объединятся значения только из тех строк, которые удовлетворяют условию

отбора. Если запросом не найдено ни одной строки или все значения в столбце равны NULL,

то функция возвращает значение NULL.

Например, запрос

***SELECT*** ***GROUP***\_***CONCAT***(name) ***FROM*** Customers;

возвращает строку, содержащую имена клиентов.

При использовании функции GROUP\_CONCAT() вы также можете указать дополни-

тельные параметры:

• DISTINCT – исключает при объединении повторяющиеся значения;

• ORDER BY – упорядочивает объединяемые значения;

• SEPARATOR – задает разделитель значений.

Например, запрос

***SELECT*** ***GROUP***\_***CONCAT***(***DISTINCT*** name

***ORDER*** ***BY*** name ***ASC*** ***SEPARATOR*** ';') ***FROM*** Customers;

возвращает строку, содержащую имена клиентов без повторений, упорядоченные по

алфавиту и разделенные точкой с запятой.

***Параметр GROUP BY***

Параметр GROUP BY позволяет объединять строки в группы, для каждой из кото-

рых групповая функция вычисляется отдельно. Для этого в параметре GROUP BY нужно

указать столбец или несколько столбцов: в одну группу попадут строки с одинаковым набо-

ром значений в этих столбцах.

Например, запрос

***SELECT*** customer\_id, ***SUM***(amount) ***FROM*** Orders

***GROUP*** ***BY*** customer\_id;

возвращает общую сумму заказов отдельно для каждого клиента (табл. 3.38). В этом

запросе заказы сгруппированы по значению столбца customer\_id (клиент), поэтому каждая

группа состоит из заказов одного клиента, а функция SUM(amount) вычисляет сумму заказов

в каждой из групп.

Таким же образом можно подсчитать количество заказов каждого клиента, максималь-

ную, минимальную и среднюю сумму заказа и др.

Другой пример – запрос, возвращающий имена клиентов с одинаковым значением рей-

тинга:

***SELECT*** ***GROUP***\_***CONCAT***(name), rating ***FROM*** Customers

***GROUP*** ***BY*** rating;

Этот запрос группирует клиентов по значению рейтинга и выводит имена клиентов в

каждой группе.

Если указано ключевое слово WITH ROLLUP, то обобщенные значения выводятся как

для отдельных групп строк, так и для всех в совокупности строк.

Например, запрос

***SELECT*** customer\_id, ***SUM***(amount) ***FROM*** Orders

***GROUP*** ***BY*** customer\_id ***WITH*** ***ROLLUP***;

возвращает, помимо общей суммы заказов каждого клиента, сумму всех заказов.

(появляется одна итоговая нуль строка с суммой всех заказов).

В запросе с параметром GROUP BY вы можете использовать условия как для отбора

отдельных строк перед группировкой, так и для отбора групп строк. Если требуется выбрать

из таблицы строки, удовлетворяющие какому-либо критерию, а затем объединить в группы

только эти строки, то применяется параметр WHERE, который должен быть указан перед

параметром GROUP BY.

Например, запрос

***SELECT*** customer\_id, ***COUNT***(amount) ***FROM*** Orders

***WHERE*** amount>5000

***GROUP*** ***BY*** customer\_id;

позволяет подсчитать, сколько заказов на сумму более 5000 сделал каждый клиент

(табл. 3.41). Сначала выбираются строки таблицы Orders (Заказы), для которых выполнено

условие amount>5 0 0 0, далее эти строки группируются по значению столбца customer\_id

(клиент), и после этого вычисляется количество строк в каждой из групп.

***Параметр HAVING***

Параметр HAVING позволяет задать условие отбора для групп строк. Он аналогичен

параметру WHERE, но указывается после параметра GROUP BY и применяется к агрегиро-

ванным строкам. В условии отбора параметра HAVING можно использовать значения столб-

цов, выводимых запросом, в том числе значения агрегатных функций.

Например, если требуется вывести общую сумму заказов для каждого клиента, кроме

тех клиентов, для кого эта сумма меньше 20 000, выполните запрос

***SELECT*** customer\_id, ***SUM***(amount) ***FROM*** Orders

***GROUP*** ***BY*** customer\_id

***HAVING*** ***SUM***(amount)> = 20000;

Условие SUM(amount) > = 20 000 позволяет отобрать только те группы строк, в которых

общая сумма заказа равна или превышает 20 000