Профессиональная программа “Анализ данных на языке SQL”

Блок 3. Трансформация таблицы

Содержание блока

- Агрегация
- Группировка
- Фильтрация групп

В конце блока предусмотрено решение ряда практических задач.

1-ый учебный вопрос

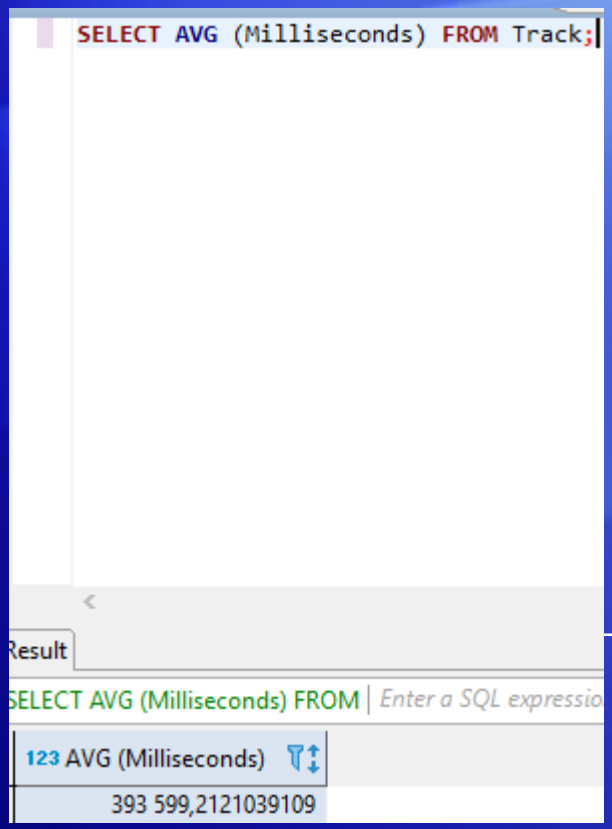
Агрегация данных

Общие сведения: агрегатные функции SQL

Агрегатная функция выполняет вычисление на наборе значений и возвращает одиночное значение. Агрегатные функции, за исключением функции COUNT не учитывают значения NULL.

Все агрегатные функции являются детерминированными, то есть, агрегатные функции возвращают одну и ту же величину при каждом их вызове на одном и том же наборе входных значений.

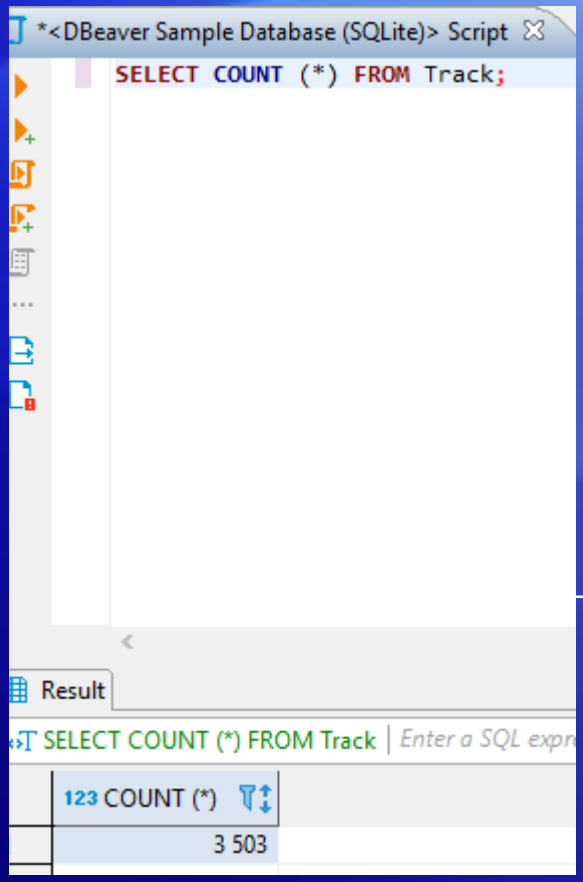
Агрегатная функция AVG



На рисунке показано, как функция AVG позволяет рассчитать среднее время длительности (столбец Milliseconds) всех музыкальных композиций, находящихся в таблице Track.

Функция AVG с целью исключения погрешности, не считает неопределенные значения NULL (NULL не тождественного арифметическому нулю).

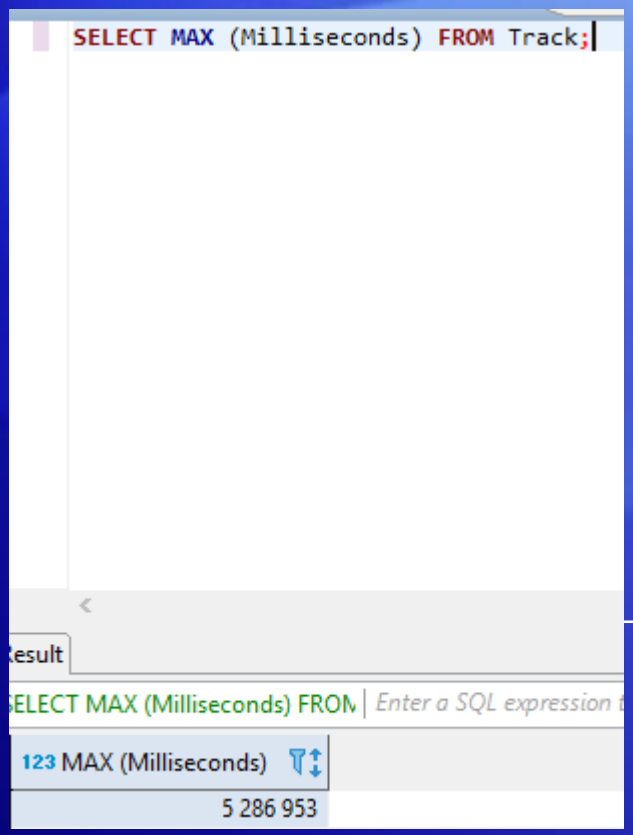
Агрегатная функция COUNT



На рисунке показано, как функция COUNT выдает в результате количество столбцов в таблице Track с учетом неопределенных значений NULL и дубликатов.

Если необходимо посчитать количество столбцов без дубликатов, необходимо в условии COUNT дописать (DISTINCT имя столбца).

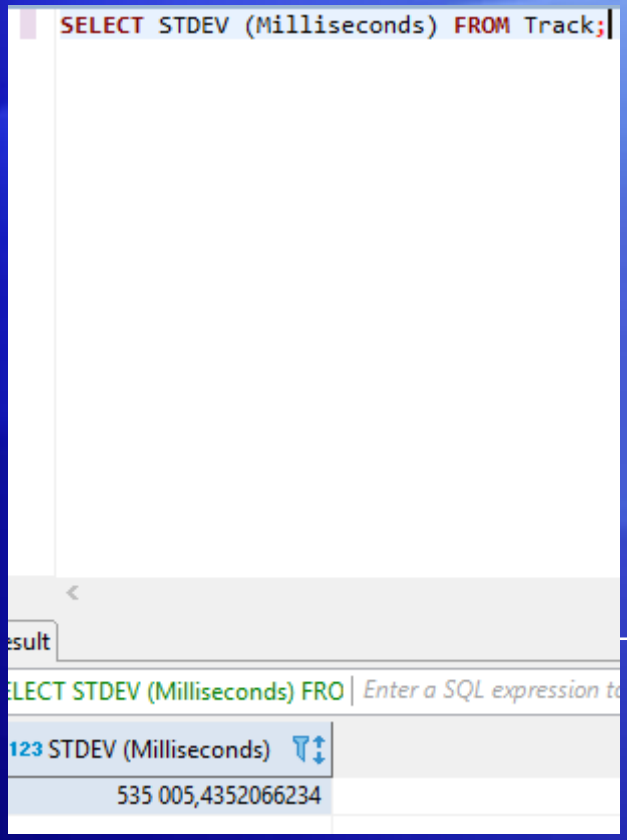
Агрегатные функции MAX/MIN



На рисунке показано, как функция MAX выдает самую длинную по времени музыкальную композицию из таблицы Track. Экстремум по максимуму берется из столбца Milliseconds.

Разумеется, функция MIN для вычисления минимального значения из столбца работает аналогичным образом.

Агрегатная функция STDEV



На рисунке показано, как функция STDEV возвращает статистическое стандартное отклонение по всей выборке значений по указанному столбцу.

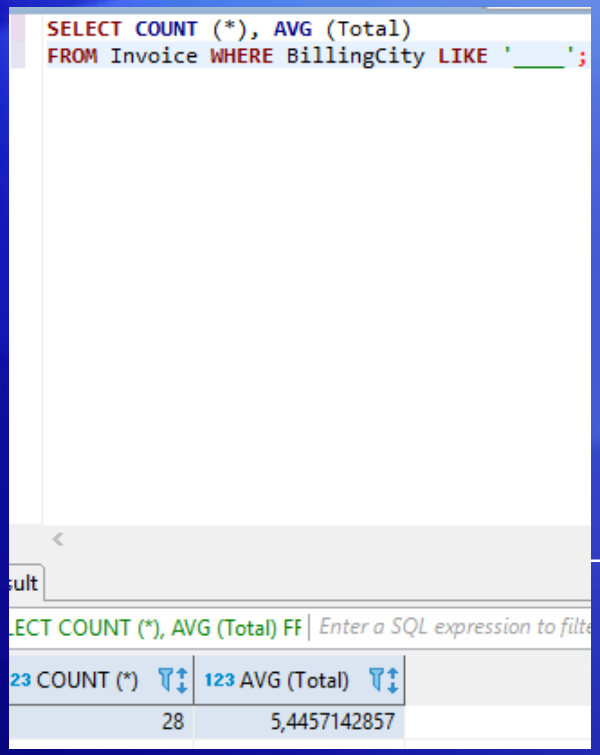
Для выражений с целью расчета отклонения применяется функция STDEVP.

Агрегатная функция SUM

На рисунке показано, как функция SUM суммирует все значения по столбцу Total таблицы Invoice. При этом значения NULL пропускаются.

Для суммирования только уникальных значений, после функции SUM необходимо применение функции DISTINCT.

Сочетание агрегатных функций



На рисунке показано, как функции COUNT и AVG составляют выборку по количеству платежей и среднему чеку (Total) для городов (BillingCity) в названии которых 4 любых буквы (LIKE ‘____’).

2-ый учебный вопрос

Группировка данных

Общие сведения: группировка данных, GROUP BY

Предложение инструкции SELECT, которое разделяет результат запроса на группы строк обычно с целью выполнения одного или нескольких статистических вычислений в каждой группе. Инструкция SELECT возвращает одну строку для каждой группы.

Группировка не подчиняется условиям выборки WHERE, для фильтрации группировки используется особый оператор HAVING.

Группировка без дополнительных опций

```
SELECT BillingCity, SUM (Total)
AS TotalSales
FROM Invoice
GROUP BY BillingCity;
```

Invoice

SELECT BillingCity, SUM (Total) A: | Enter a SQL exp

BillingCity	TotalSales
Amsterdam	40,62
Bangalore	36,64
Berlin	75,24
Bordeaux	39,62
Boston	37,62
Brasília	37,62
Brussels	37,62

На картинке видно, как из таблицы Invoice выбираются столбцы BillingCity и Total (массив данных этого столбца суммируется и помещается во временный столбец с названием TotalSales), после чего вся выборка полученных значений группируется по названию BillingCity.

В результате выводится информация о суммарных продажах по каждому отдельному городу.

3-ий учебный вопрос

Фильтрация групп

Общие сведения: инструкция HAVING

Определяет условие поиска для группы или статистического выражения. Предложение HAVING можно использовать только в инструкции SELECT.

HAVING обычно используется с предложением GROUP BY. Если предложение GROUP BY не используется, имеется одна неявная агрегированная группа.

Группировка с дополнительным условием HAVING

```
SELECT BillingCity, SUM(Total)
AS TotalSales
FROM Invoice
GROUP BY BillingCity
HAVING TotalSales BETWEEN 39 AND 42
ORDER BY TotalSales DESC;
```

Invoice

SELECT BillingCity, SUM(Total) AS | Enter a SQL express

BillingCity	TotalSales
Amsterdam	40,62
Lisbon	39,62
Redmond	39,62
Bordeaux	39,62
Montréal	39,62
Orlando	39,62
Oslo	39,62

На картинке видно, что условие в HAVING задается по тому же принципу, как и условие для WHERE. Условие можно применять и к временному столбцу, который был создан для данной выборки.

Как видно из результата, на временный столбец также можно исполнить условие сортировки GROUP BY.