Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Базы данных

Лабораторная работа № 2

«Выборка данных с использованием предложения SELECT»

Выполнил:

ст.гр. 113802

Разумов Д.А.

Проверила:

Василькова А.Н.

Минск 2022

Итак, в нашей БД Автогараж есть три таблицы: routs (маршруты), drivers (водители) и traffics (машины). И мы хотим посмотреть, какие данные в них содержатся. Для этого в SQL существует оператор SELECT. Синтаксис его использования, следующий:

SELECT что\_выбрать FROM откуда\_выбрать;

Вместо "что\_выбрать" мы должны указать либо имя столбца, значения которого хотим увидеть, либо имена нескольких столбцов через запятую, либо символ звездочки (\*), означающий выбор всех столбцов таблицы. Вместо "откуда\_выбрать" следует указать имя таблицы.

На рисунке 1 приведен пример запроса с оператором SELECT.

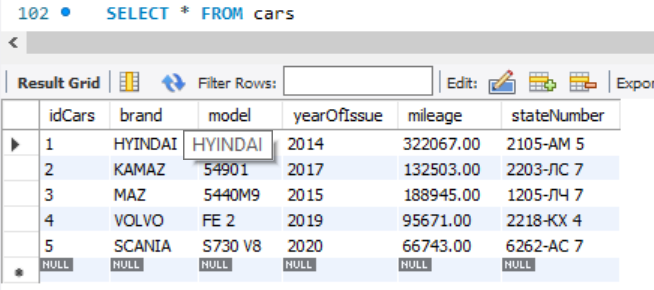


Рисунок 1 – Пример запроса с SELECT

Мы увидели все данные, которые вносили в эту таблицу. Но предположим, что мы хотим посмотреть только столбец brand. Для этого в запросе мы укажем имя этого столбца:

SELECT brand FROM routs;

На рисунке 2 выведен только столбец brand.

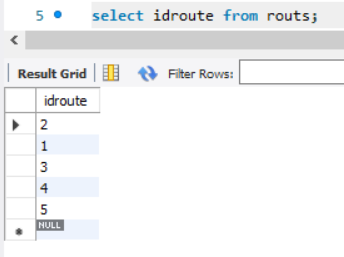


Рисунок 2 – Пример запроса SELECT только для одного столбца

Если мы захотим посмотреть, например, brand и model маршрутов, то перечислим интересующие столбцы через запятую:

SELECT brand, model FROM routs;

На рисунке 3 приведен данный пример.

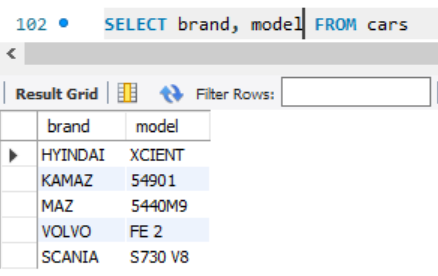


Рисунок 3 - Пример запроса SELECT для двух определенных столбцов

В настоящее время имеется 4 темы, а если их будет 100? Хотелось бы, чтобы они выводились, например, по алфавиту. Для этого в SQL существует ключевое слово ORDER BY после которого указывается имя столбца, по которому будет происходить сортировка. Пример работы данного оператора приведен на рисунке 4.

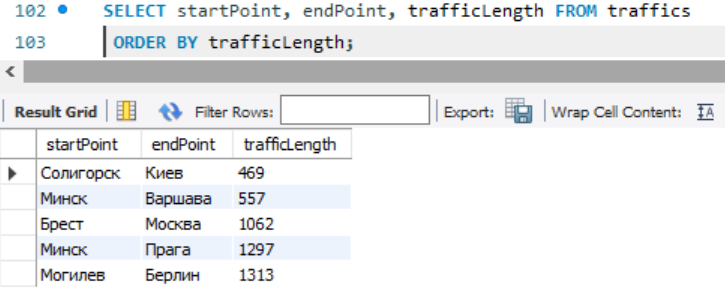


Рисунок 4 – Пример работы оператора ORDER BY

По умолчанию сортировка идет по возрастанию. Для сортировки по убыванию значений – необходимо использовать ключевое слово DESC. Пример работы данного оператора представлен на рисунке 5.

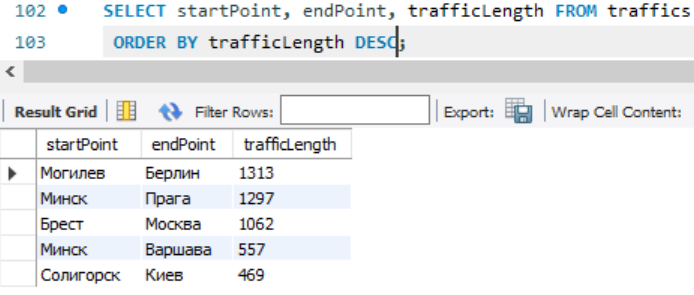


Рисунок 5 – Пример работы операторов ORDER BY и DESC

Например, мы хотим узнать, у какого маршрута trafficLength> 400 или trafficLength <600. Для этого в SQL есть ключевое слово WHERE, синтаксис у такого запроса, следующий:

SELECT имя\_столбца FROM имя\_таблицы WHERE условие;

Пример работы данного оператора показан на рисунке 6 - 8.

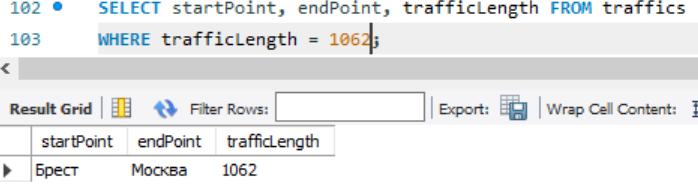


Рисунок 6 – Пример работы оператора ‘=’ и оператора WHERE

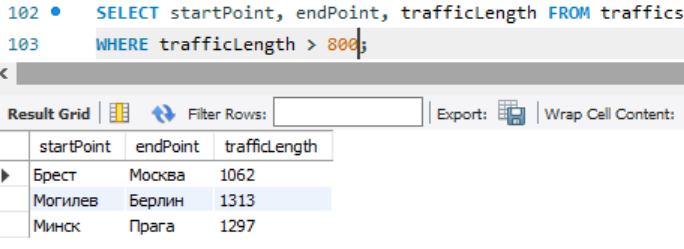


Рисунок 7 – Пример работы оператора ‘≥’ и оператора WHERE

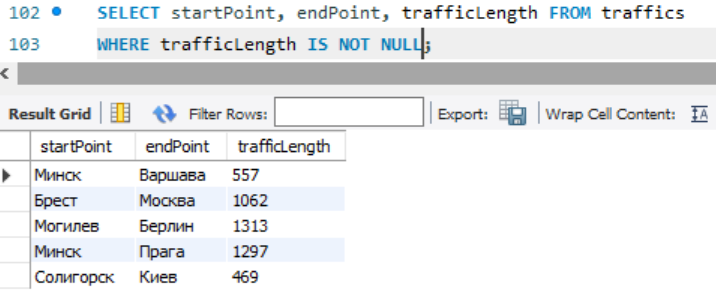


Рисунок 8 – Пример работы оператора IS NOT NULL и оператора WHERE

Поиск с использованием метасимволов может осуществляться только в текстовых полях. Самый распространенный метасимвол - %. Он означает любые символы. Например, если нам надо найти слова, начинающиеся с букв "вел", то мы напишем LIKE 'вел%', а если мы хотим найти слова, которые содержат символы "клуб", то мы напишем LIKE '%клуб%'. Пример показан на рисунке 9.

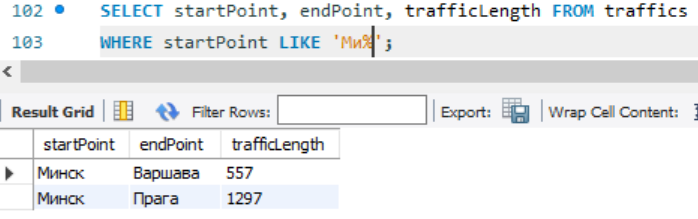


Рисунок 9 – Пример работы оператора LIKE и оператора WHERE

Агрегатные функции вычисляют некоторые скалярные значения в наборе строк. В MySQL есть следующие агрегатные функции:

− AVG: вычисляет среднее значение

* SUM: вычисляет сумму значений
* MIN: вычисляет наименьшее значение
* MAX: вычисляет наибольшее значение
* COUNT: вычисляет количество строк в запросе

Все агрегатные функции принимают в качестве параметра выражение, которое представляет критерий для определения значений. Зачастую, в качестве выражения выступает название столбца, над значениями которого надо проводить вычисления. Выражения в функциях AVG и SUM должно представлять числовое значение (например, столбец, который хранит числовые значения). Выражение в функциях MIN, MAX и COUNT может представлять числовое или строковое значение или дату. Все агрегатные функции за исключением COUNT (\*) игнорируют значения NULL.

С помощью оператора AVG вычислим среднее значение величины mileage. Результата показан на рисунке 10.

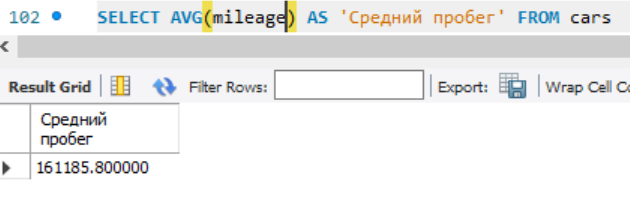


Рисунок 10 – Пример работы оператора AVG

С помощью операторов MIN и MAX вычислим максимальное и минимальное значение величины year\_of\_issue. Результата показан на рисунке 11.

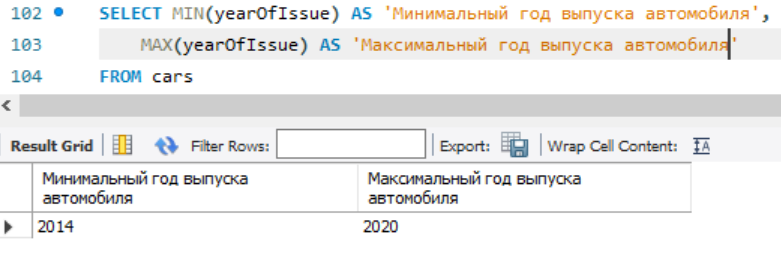


Рисунок 11 – Пример работы операторов MIN и MAX

Подсчитаем общее количество кортежей в таблице drivers c помощью оператора COUNT. Результат показан на рисунке 12.

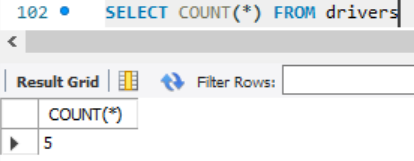


Рисунок 12 – Пример работы оператора COUNT

Подсчитаем сумму значений величины mileage c помощью оператора SUM. Результат показан на рисунке 13.

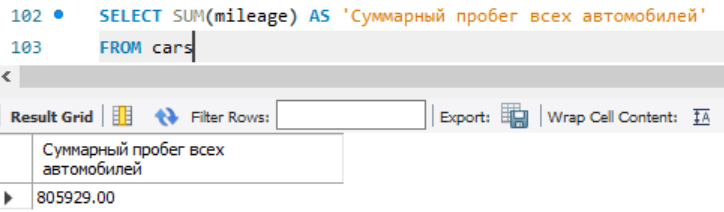


Рисунок 13 – Пример работы оператора SUM

По умолчанию все вышеперечисленных пять функций учитывают все строки выборки для вычисления результата. Но выборка может содержать повторяющие значения. Если необходимо выполнить вычисления только над уникальными значениями, исключив из набора значений повторяющиеся данные, то для этого применяется оператор DISTINCT.

Подсчитаем количество уникальных значений начальной точки отправления startPoint. Результат показан на рисунке 14.

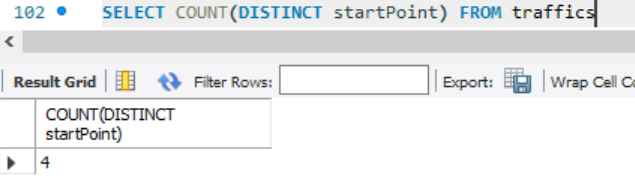


Рисунок 14 – Пример работы оператора DISTINCT

Операторы GROUP BY и HAVING позволяют сгруппировать данные.

Они употребляются в рамках команды SELECT. Оператор GROUP BY определяет, как строки будут группироваться. Например, сгруппируем маршруты по точке отправления. Первый столбец в выражении SELECT – startPoint представляет название группы, а второй столбец – startPointCount представляет результат функции Count, которая вычисляет количество строк в группе. Результат показан на рисунке 15.

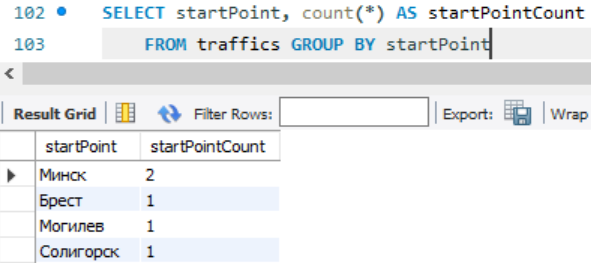


Рисунок 15 – Пример работы оператора GROUP BY

Оператор HAVING позволяет выполнить фильтрацию групп, то есть определяет, какие группы будут включены в выходной результат. Использование HAVING во многом аналогично применению WHERE. Только есть WHERE применяется для фильтрации строк, то HAVING - для фильтрации групп. Например, найдем все маршруты, для которых одинаковое количество остановок меньше 2. Результат показан на рисунке 16.

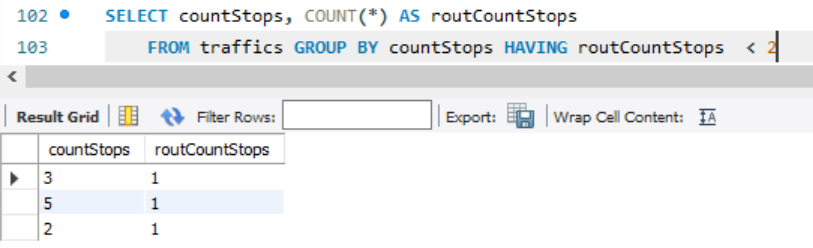


Рисунок 16 – Пример работы операторов GROUP BY и HAVING

Для этого же подсчета выведем данные по убыванию с помощью операторов ORDER BY и DESC. Результат показан на рисунке 17.

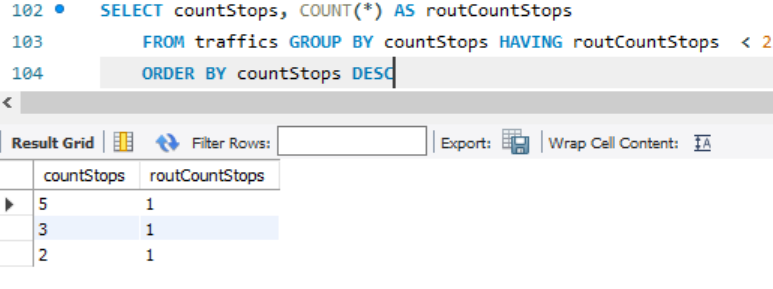


Рисунок 17 – Пример работы операторов GROUP BY, HAVING, ORDER BY

Таким образом, исходя полученных результатов, мы ознакомились с основными операторами и агрегатными функциями, с сортировкой и группировкой в языке SQL в среде разработки MySQL Workbench.