Практическое задание

Урок 8. Оконные функции, индексы, работа в графическом клиенте

Задание

Работаем с базой данных учителей teachers.db. Для каждого задания требуется сдать только код, который выполняется для получения результата, в текстовом файле. В качестве отчёта к четвёртому заданию надо приложить скриншот.

1. Найдите общее количество учеников для каждого курса. В отчёт выведите название курса и количество учеников по всем потокам курса. Решите задание с применением оконных функций.

2. Найдите среднюю оценку по всем потокам для всех учителей. В отчёт выведите идентификатор, фамилию и имя учителя, среднюю оценку по всем проведённым потокам. Учителя, у которых не было потоков, также должны попасть в выборку. Решите задание с применением оконных функций.

3. Какие индексы надо создать для максимально быстрого выполнения представленного запроса?

```
SELECT
  surname,
  name,
  number,
  performance
FROM academic_performance
  JOIN teachers
    ON academic_performance.teacher_id = teachers.id
  JOIN streams
    ON academic_performance.stream_id = streams.id
WHERE number >= 200;
```

4. Установите SQLiteStudio, подключите базу данных учителей, выполните в графическом клиенте любой запрос.

5. Дополнительное задание. Для каждого преподавателя выведите имя, фамилию, минимальное значение успеваемости по всем потокам преподавателя, название курса, который соответствует потоку с минимальным значением успеваемости, максимальное значение успеваемости по всем потокам преподавателя, название курса, соответствующий потоку с максимальным значением успеваемости, дату начала следующего потока. Выполните задачу с использованием оконных функций.

```
sqlite> .open teachers8.db
sqlite> .tables
courses  grades   streams  teachers
sqlite> SELECT * FROM courses;
id  name
--  ----------------------
1   Базы данных
2   Основы Python
3   Linux. Рабочая станция
sqlite> SELECT * FROM grades;
id_teacher  id_stream  grade
----------  ---------  -----
3           1.0        4.7
2           2.0        4.9
1           3.0        4.8
1           4.0        4.9
sqlite> SELECT * FROM streams;
id  course_id  number  started_at  finished_at  students_amount
--  ---------  ------  ----------  -----------  ---------------
1   3          165     2020-08-18  0            34
2   2          178     2020-10-02  0            37
3   1          203     2020-11-12  0            35
4   1          210     2020-12-03  0            41
sqlite> SELECT * FROM teachers;
id  surname  name      email
--  -------  --------  --------------------
1   Николай  Савельев  saveliev.n@mail.ru
2   Наталья  Петрова   petrova.n@yandex.ru
3   Елена    Малышева  malisheva.e@google.com
```

1. Найдите общее количество учеников для каждого курса. В отчёт выведите название курса и количество учеников по всем потокам курса. Решите задание с применением оконных функций.

```
sqlite> SELECT
   ...> courses.name,
   ...> SUM(streams.students_amount)
   ...> FROM streams
   ...> LEFT JOIN courses
   ...> ON streams.course_id = courses.id
   ...> GROUP BY courses.name;
name                 SUM(streams.students_amount)
-------------------  ---------------------------
Linux. Рабочая станция  34
Базы данных          76
Основы Python        37


sqlite> SELECT DISTINCT
   ...> courses.name,
   ...> SUM(streams.students_amount) OVER(w_students) AS sum_students
   ...> FROM streams
   ...> LEFT JOIN courses
   ...> ON streams.course_id = courses.id
   ...> WINDOW w_students AS (PARTITION BY courses.name);
name                 sum_students
-------------------  ------------
Linux. Рабочая станция  34
Базы данных          76
Основы Python        37
```

2. Найдите среднюю оценку по всем потокам для всех учителей. В отчёт выведите идентификатор, фамилию и имя учителя, среднюю оценку по всем проведённым потокам. Учителя, у которых не было потоков, также должны попасть в выборку. Решите задание с применением оконных функций.

```
sqlite> SELECT
   ...> teachers.id,
   ...> teachers.name,
   ...> teachers.surname,
   ...> AVG(grades.grade)
   ...> FROM teachers
   ...> LEFT JOIN grades
   ...> ON teachers.id = grades.id_teacher
   ...> GROUP BY id_teacher;
id  name      surname  AVG(grades.grade)
--  --------  -------  -----------------
1   Савельев  Николай  4.85
2   Петрова   Наталья  4.9
3   Малышева  Елена    4.7


sqlite> SELECT DISTINCT
   ...> teachers.id,
   ...> teachers.name,
   ...> teachers.surname,
   ...> AVG(grades.grade) OVER(w_grades) AS avg_grade
   ...> FROM teachers
   ...> LEFT JOIN grades
   ...> ON teachers.id = grades.id_teacher
   ...> WINDOW w_grades AS (PARTITION BY id_teacher);
id  name      surname  avg_grade
--  --------  -------  ---------
1   Савельев  Николай  4.85
```

2 Петрова Наталья 4.9

3 Малышева Елена 4.7

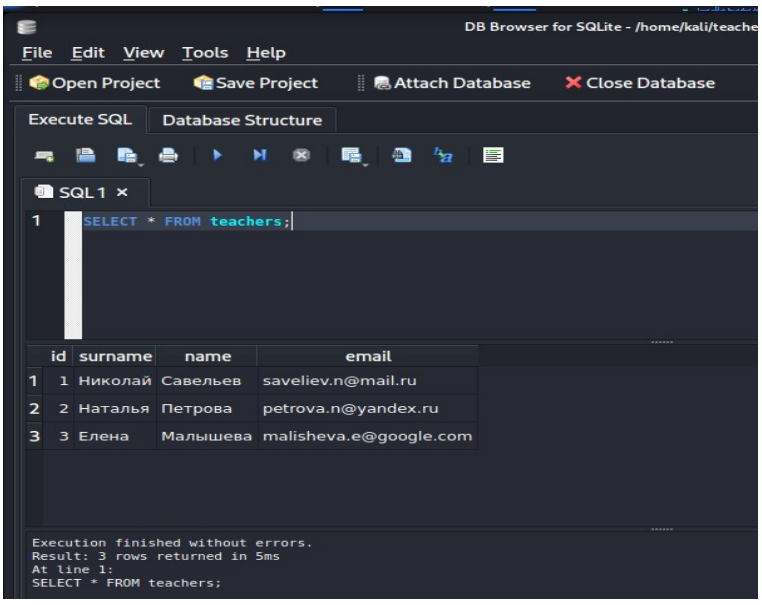
3. Какие индексы надо создать для максимально быстрого выполнения представленного запроса?

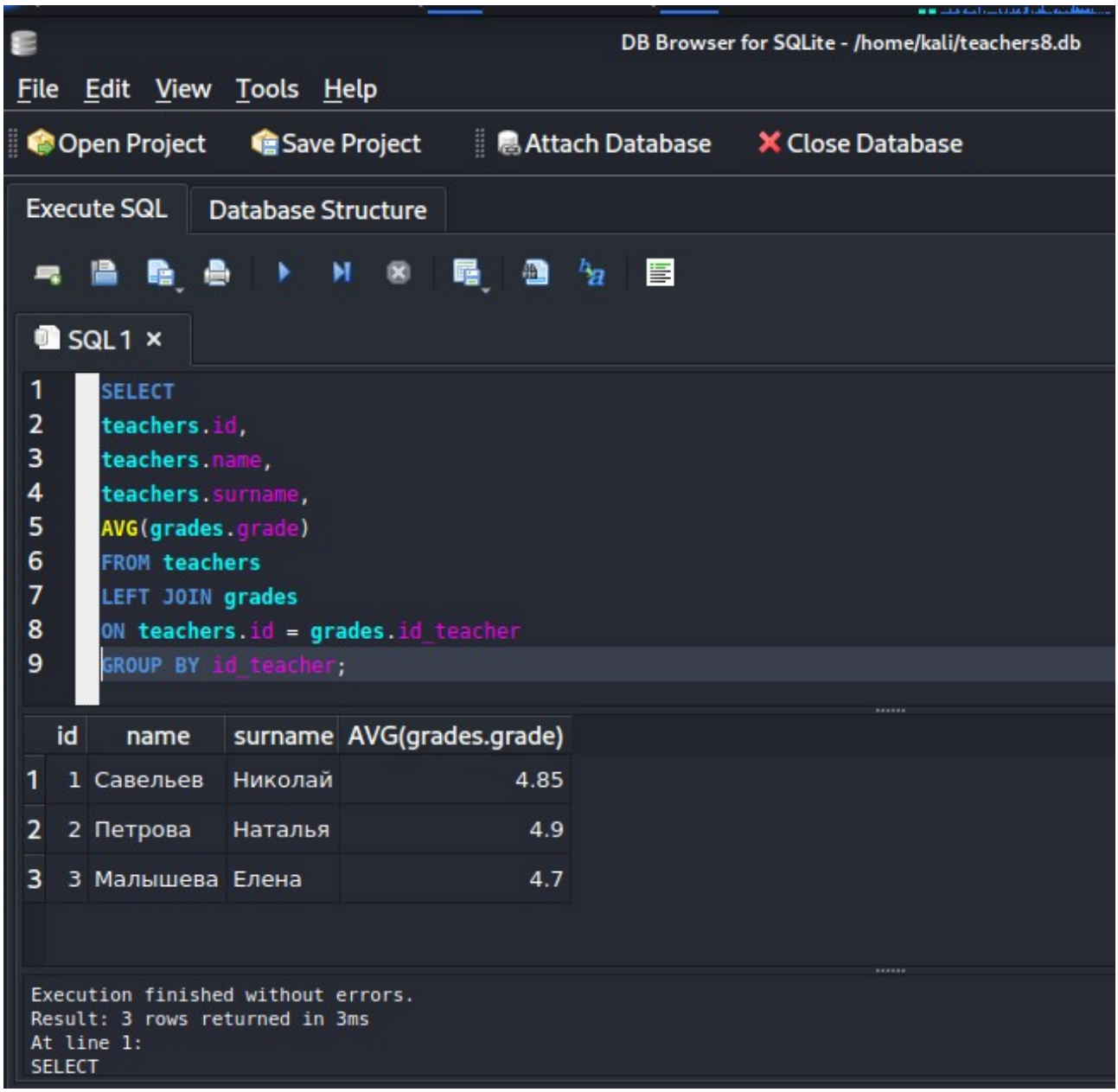
```
SELECT
  surname,
  name,
  number,
  performance
FROM academic_performance
  JOIN teachers
    ON academic_performance.teacher_id = teachers.id
  JOIN streams
    ON academic_performance.stream_id = streams.id
WHERE number >= 200;
```

**

```
sqlite> CREATE INDEX teachers_surname_name_indx ON teachers(surname, name);
sqlite> CREATE INDEX streams_number_indx ON streams(number);
```

4. Установите SQLiteStudio, подключите базу данных учителей, выполните в графическом клиенте любой запрос.





5. Дополнительное задание. Для каждого преподавателя выведите имя, фамилию, минимальное значение успеваемости по всем потокам преподавателя, название курса, который соответствует потоку с минимальным значением успеваемости, максимальное значение успеваемости по всем потокам преподавателя, название курса, соответствующий потоку с максимальным значением успеваемости, дату начала следующего потока. Выполните задачу с использованием оконных функций.