**Тема лр:** Выполнение индивидуальных заданий по формированию SQL-запросов к БД в среде sqlite.

**Таблица студенты**

create table students (id serial primary key, syname varchar(30), name varchar(30), birthday date);

|  |  |  |  |
| --- | --- | --- | --- |
|  | **фамилия** | **имя** | **День рождение** |
| **1** | Фролов | Данил | 2001-06-22 |
| **2** | Китов | Виктор | 1998-11-03 |
| **3** | Тюрина | Александра | 1998-03-30 |
| **4** | Петров | Иван | 2000-01-04 |
| **5** | Чащина | Татьяна | 1999-06-06 |
| **6** | Лосенкова | Александра | 1999-04-08 |

Insert into students (syname, name, birthday)values ( ‘Фролов’, ‘Данил’, ‘2001-06-22’);

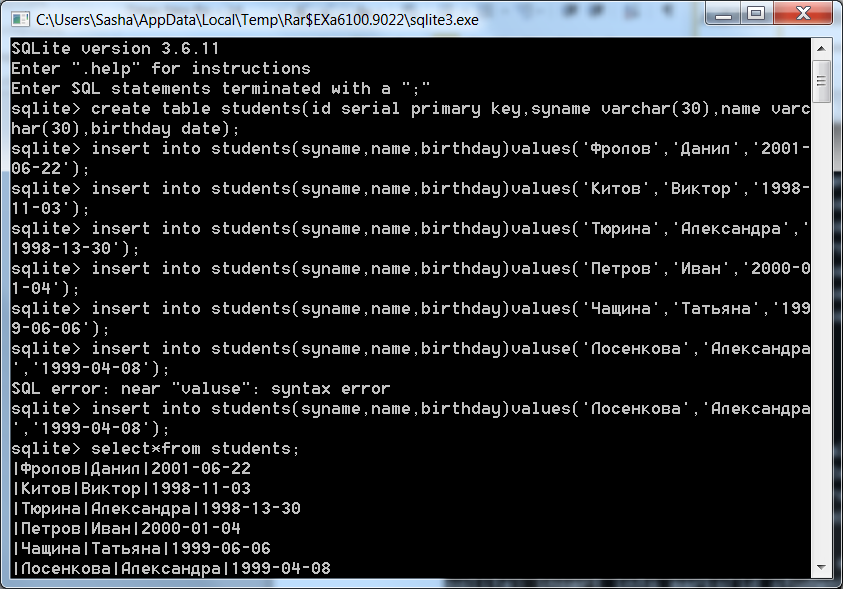
Insert into students (syname, name, birthday)values ( ‘Китов’, ‘Виктор’, ‘1998-11-03’);

Insert into students (syname, name, birthday)values ( ‘Тюрина’, ‘Александра’, ‘1998-13-30’);

Insert into students (syname, name, birthday)values ( ‘Петров’, ‘Иван’, ‘2000-01-04’);

Insert into students (syname, name, birthday)values ( ‘Чащина’, ‘Татьяна’, ‘1999-06-06’);

Insert into students (syname, name, birthday)values ( ‘Лосенкова’, ‘Александра’, ‘1999-04-08’);



**Студенты предметы**

create table subjects (id serial primary key, name varchar(30));

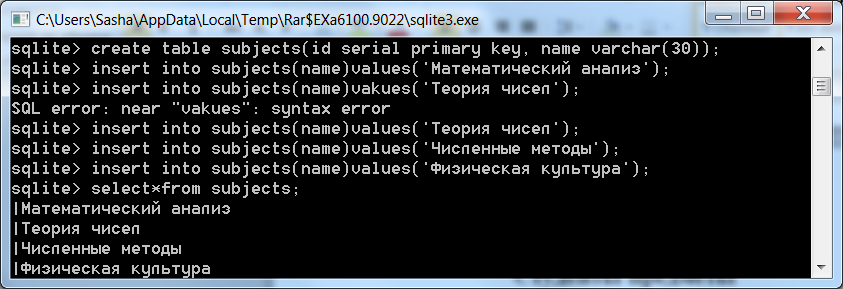
|  |  |
| --- | --- |
| **Ид предмета** | **Название** |
| 1 | Математический анализ |
| 2 | Теория чисел |
| 3 | Численные методы |
| 4 | Физическая культура |

Insert into subjects (name)values ( ‘Математичсекий анализ’);

Insert into subjects (name)values ( ‘Теория чисел’);

Insert into subjects (name)values ( ‘Численные методы’);

Insert into subjects (name)values ( ‘Физическая культура’);



**Оценки**

create table marks (id serial primary key, id\_students integer, id\_subjects integer, mark integer, mark\_date date, foreign key (id\_students) references students(id), foreign key (id\_subjects) references subjects(id));

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ид оценки** | **Ид студента** | **Ид предмета** | **Оценка** | **Дата** |
| 1 | 1 | 1 | 2 | 2020-03-14 |
| 2 | 2 | 2 | 3 | 2019-09-22 |
| 3 | 3 | 3 | 4 | 2020-04-10 |
| 4 | 4 | 4 | 5 | 2018-10-25 |
| 5 | 5 | 2 | 2 | 2018-12-03 |
| 6 | 6 | 4 | 5 | 2019-06-30 |
| 7 | 2 | 2 | 3 | 2019-09-22 |

Insert into marks (id\_students, id\_subjects,mark,mark\_date)values ( ‘1’,’1’,’2’,’2020-03-14’);

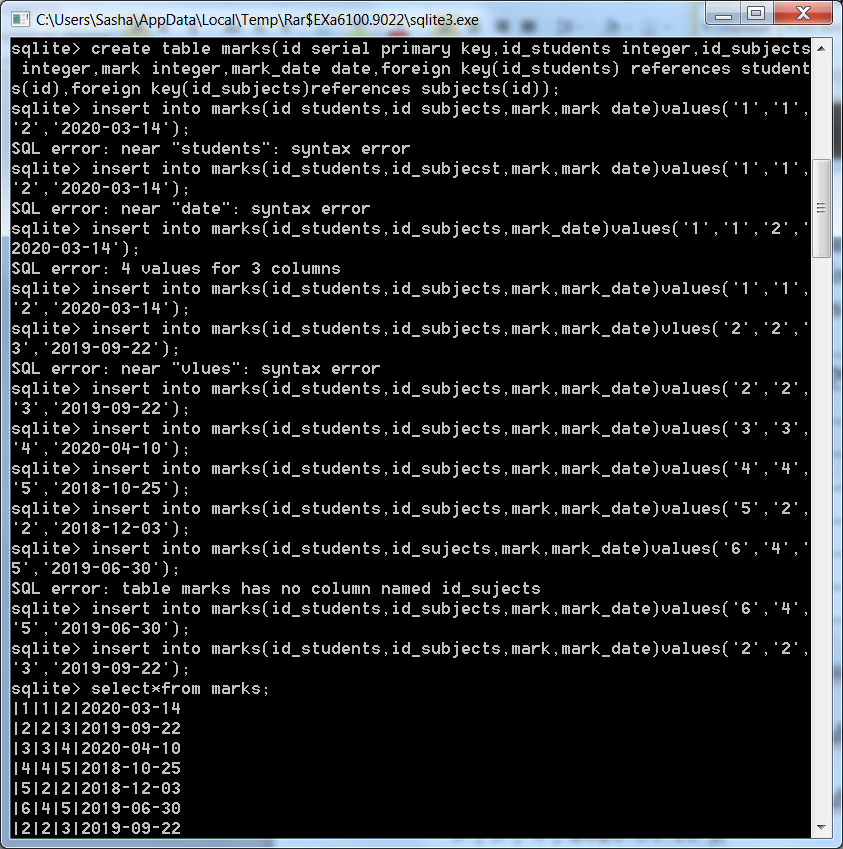
Insert into marks (id students, id subjects,mark,mark date)values ( ‘2’,’2’,’3’,’2019-09-22’);

Insert into marks (id students, id subjects,mark,mark date)values ( ‘3’,’3’,’4’,’2020-04-10’);

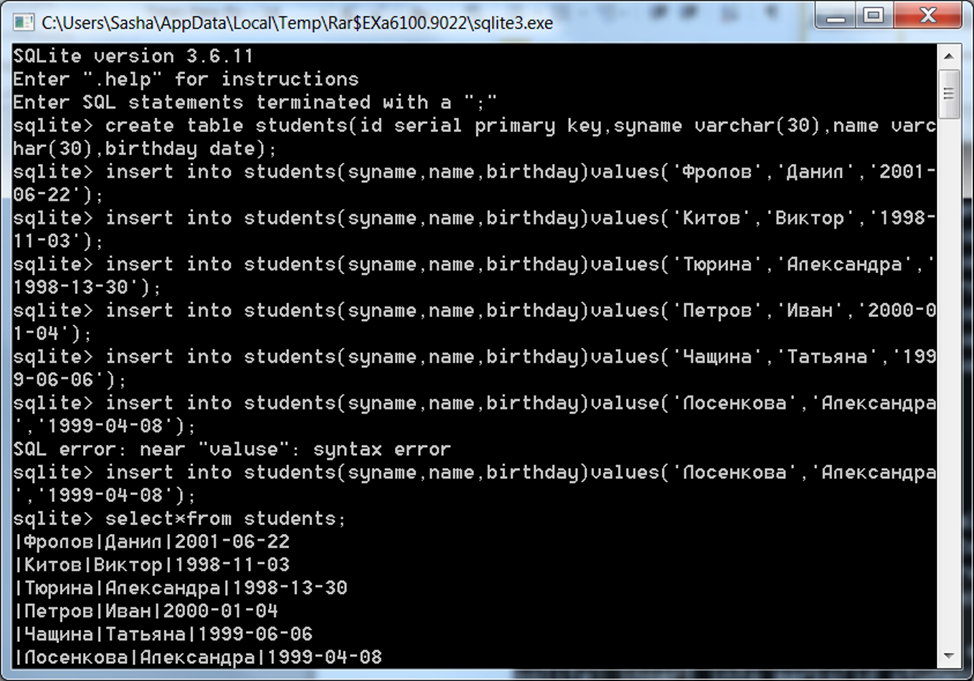
Insert into marks (id students, id subjects,mark,mark date)values ( ‘4’,’4’,’5’,’2018-10-25’);  
Insert into marks (id students, id subjects,mark,mark date)values ( ‘5’,’2’,’2’,’2018-12-03’);

Insert into marks (id students, id subjects,mark,mark date)values ( ‘6’,’4’,’5’,’2019-06-30’);

Insert into marks (id students, id subjects,mark,mark date)values ( ‘2’,’2’,’3’,’2019-09-22’);

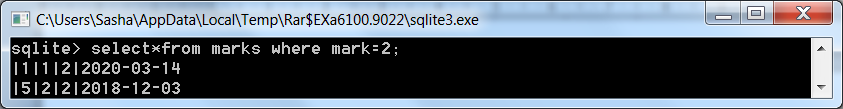


**Пример 1** Вывод таблицы студентыselect \* from students;



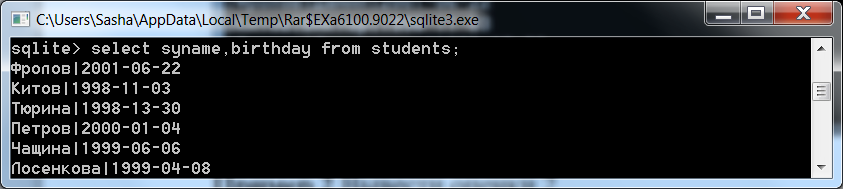
**Пример 2** Вывести оценки 2

select \* from marks where mark=2;



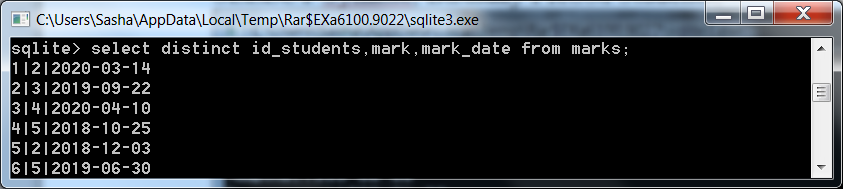
**Пример 3.** Выбрать колонки фамилии и дата рождения из исходной таблицы Students:

SELECT Syname, Birthday FROM Students;



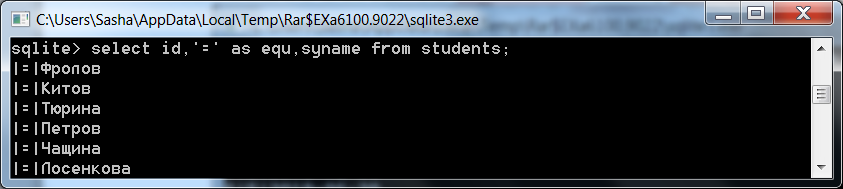
**Пример 4.** Выбрать некоторые колонки из исходной таблицы, удалив из результата повторяющиеся строки:

SELECT DISTINCT id\_students, mark, mark\_date FROM Marks;



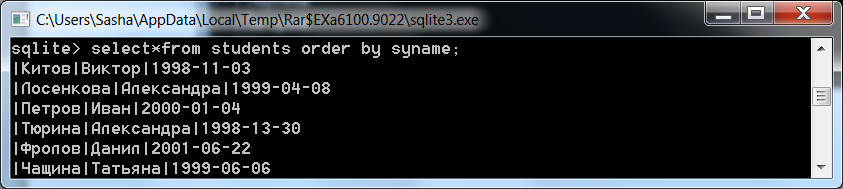
**Пример 5.** Использование переименования колонок в запросах:

SELECT id, ‘=’ AS Equ, Syname FROM Students;



**Пример 6.** Упорядочение результатов запроса:

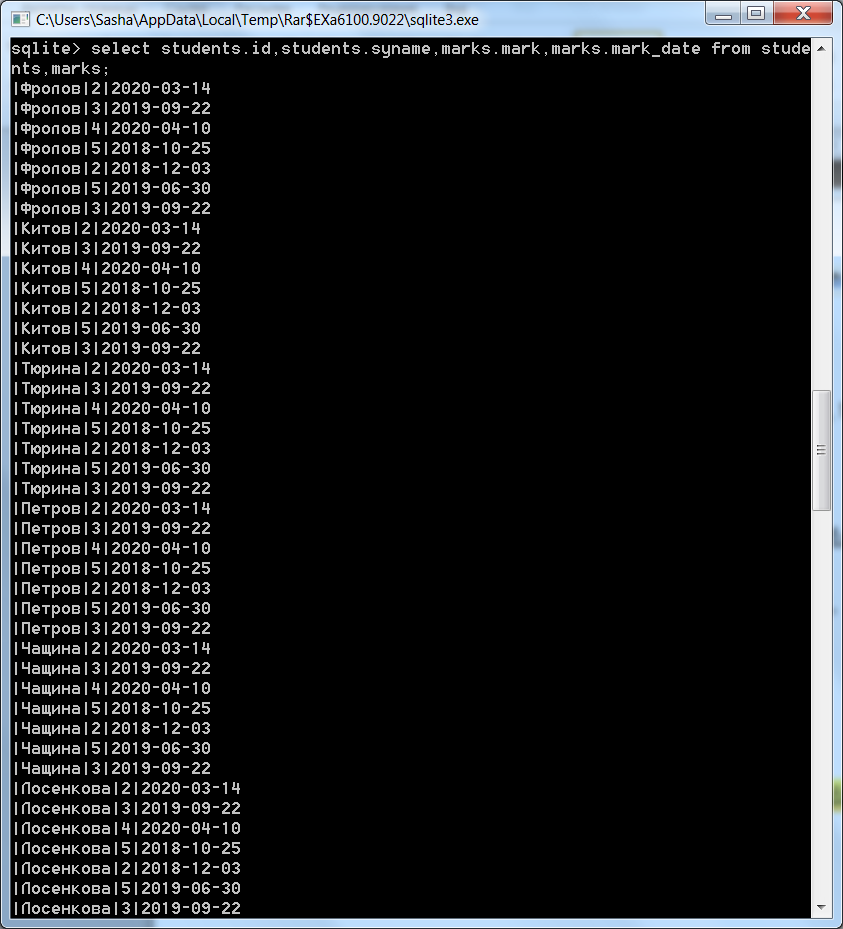
SELECT \* FROM Students ORDER BY Syname;



**Пример 7.** Прямое произведение таблиц:

SELECT Students.id, Students.Syname, Marks.mark, Marks.mark\_date

FROM Students, Marks;

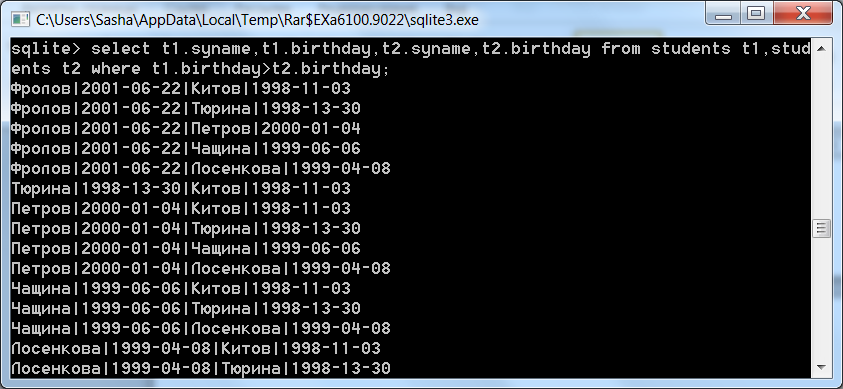


**Пример 8.** Отобрать все пары учеников таким образом, чтобы первый ученик в паре был младше второго:

SELECT t1.Syname, t1.Birthday, t2.Syname, t2.Birthday

FROM Students t1, Students t2

WHERE t1.Birthday>t2.Birthday;



**Пример 9.** Получить общее количество оценок:

SELECT COUNT(\*) AS kol\_mark FROM Marks;



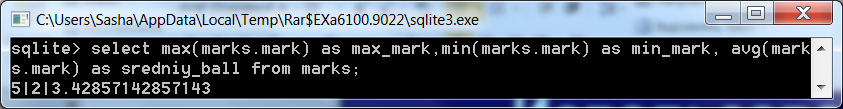
**Пример 10.** Получить максимальную, минимальную и среднюю оценку:

SELECT MAX (marks.mark) AS Max\_mark,

MIN(marks.mark) AS Min\_mark,

AVG(marks.mark) AS Sredniy\_ball

FROM Marks;



**Пример 11.** Получить список студентов, возраст которых больше минимального возраста в таблице студентов:

SELECT \* FROM Students s

WHERE s.Birthday<(SELECT MAX(s.birthday) FROM Students s);

