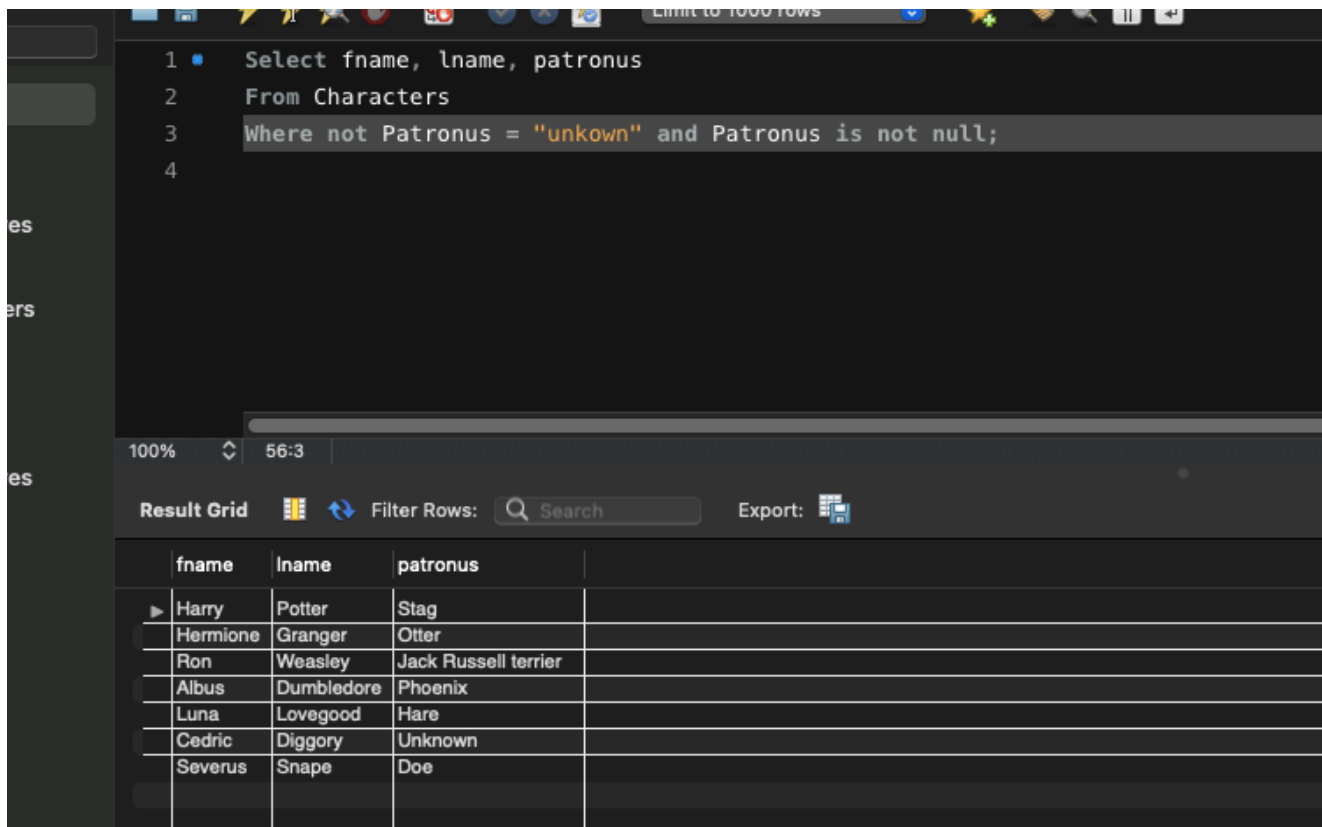


1. Выведите имя, фамилию, патронуса всех персонажей, у которых есть patronus и он известен



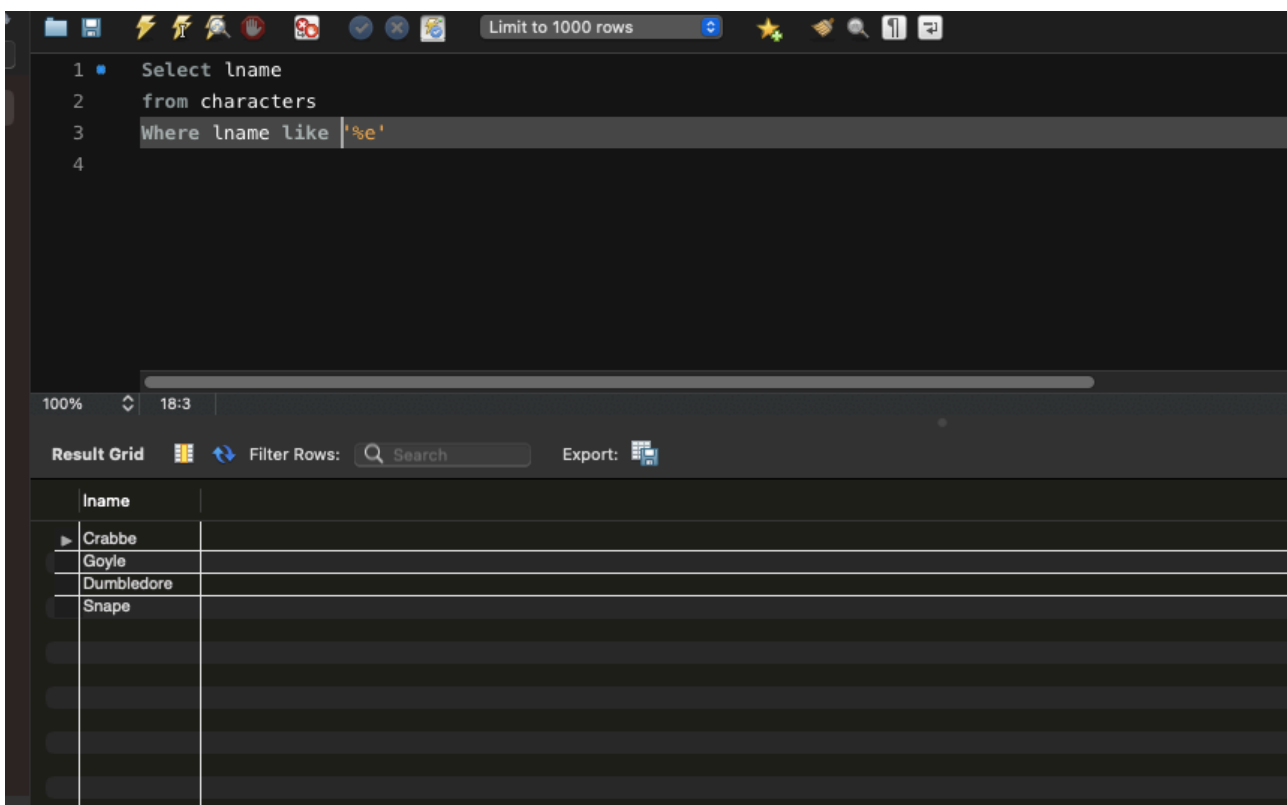
The screenshot shows a database query editor with a SQL query and its results. The query is:

```
1 Select fname, lname, patronus
2 From Characters
3 Where not Patronus = "unkown" and Patronus is not null;
4
```

The results are displayed in a table with columns: fname, lname, patronus.

fname	lname	patronus
Harry	Potter	Stag
Hermione	Granger	Otter
Ron	Weasley	Jack Russell terrier
Albus	Dumbledore	Phoenix
Luna	Lovegood	Hare
Cedric	Diggory	Unknown
Severus	Snape	Doe

2. Выведите фамилию персонажей, у которых последняя буква в фамилии 'е'



The screenshot shows a database query editor with a SQL query and its results. The query is:

```
1 Select lname
2 from characters
3 Where lname like '%e'
4
```

The results are displayed in a table with columns: lname.

lname
Crabbe
Goyle
Dumbledore
Snape

3. Посчитайте общий возраст всех персонажей и выведите это на экран

The screenshot shows a database query editor with a dark theme. The query editor has a toolbar with icons for file operations, execution, and settings. The query text is as follows:

```
1 Select SUM(age)
2 From Characters
```

Below the query editor, there is a 'Result Grid' section. It shows a single row with the result of the query:

S...
257

4. Выведите имя, фамилию и возраст персонажей по убыванию их возраста

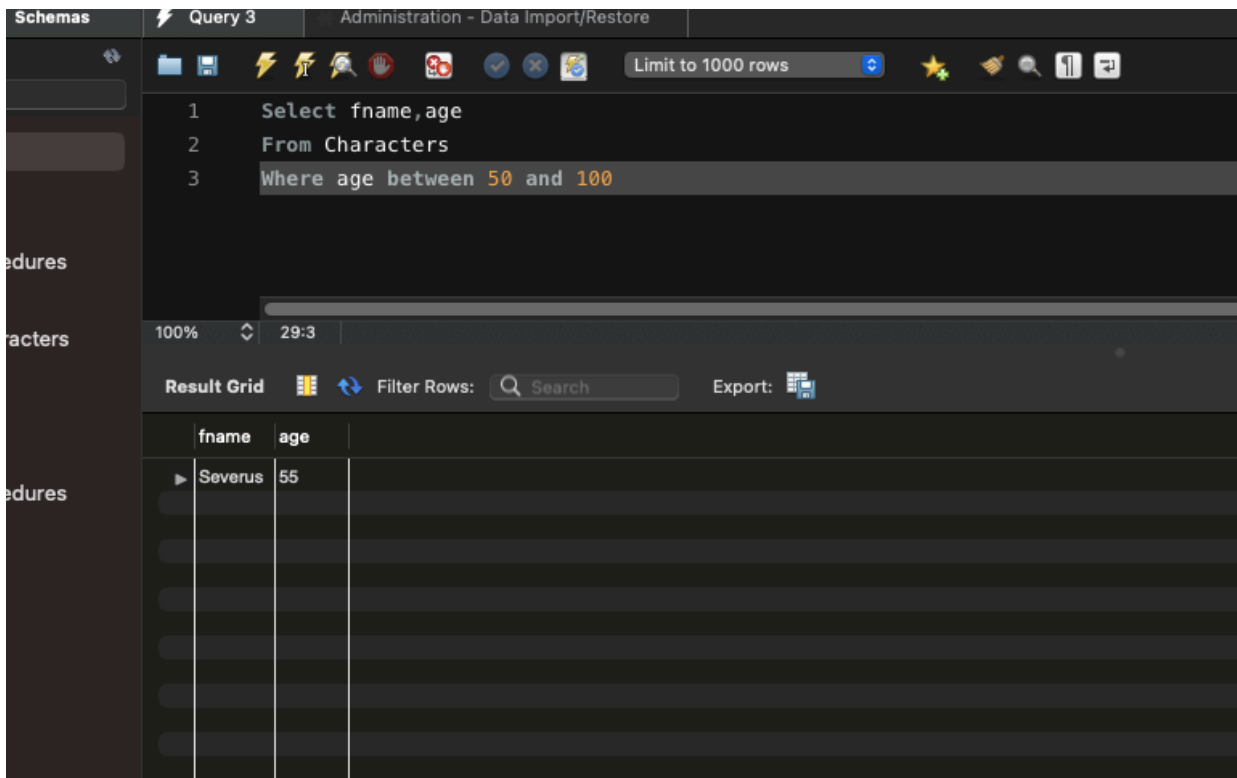
The screenshot shows a database query editor with a dark theme. The query editor has a toolbar with icons for file operations, execution, and settings. The query text is as follows:

```
1 Select fname, lname, age
2 From Characters
3 ORDER by age DESC
```

Below the query editor, there is a 'Result Grid' section. It shows a table with the results of the query:

fname	lname	age
Albus	Dumbledore	111
Severus	Snape	55
Cedric	Diggory	14
Harry	Potter	11
Hermione	Granger	11
Ron	Weasley	11
Draco	Malfoy	11
Vincent	Crabbe	11
Gregory	Goyle	11
Luna	Lovegood	11
Lord	Voldemort	NULL

5. Выведите имя персонажа и возраст, у которых последний находится в диапазоне от 50 до 100 лет



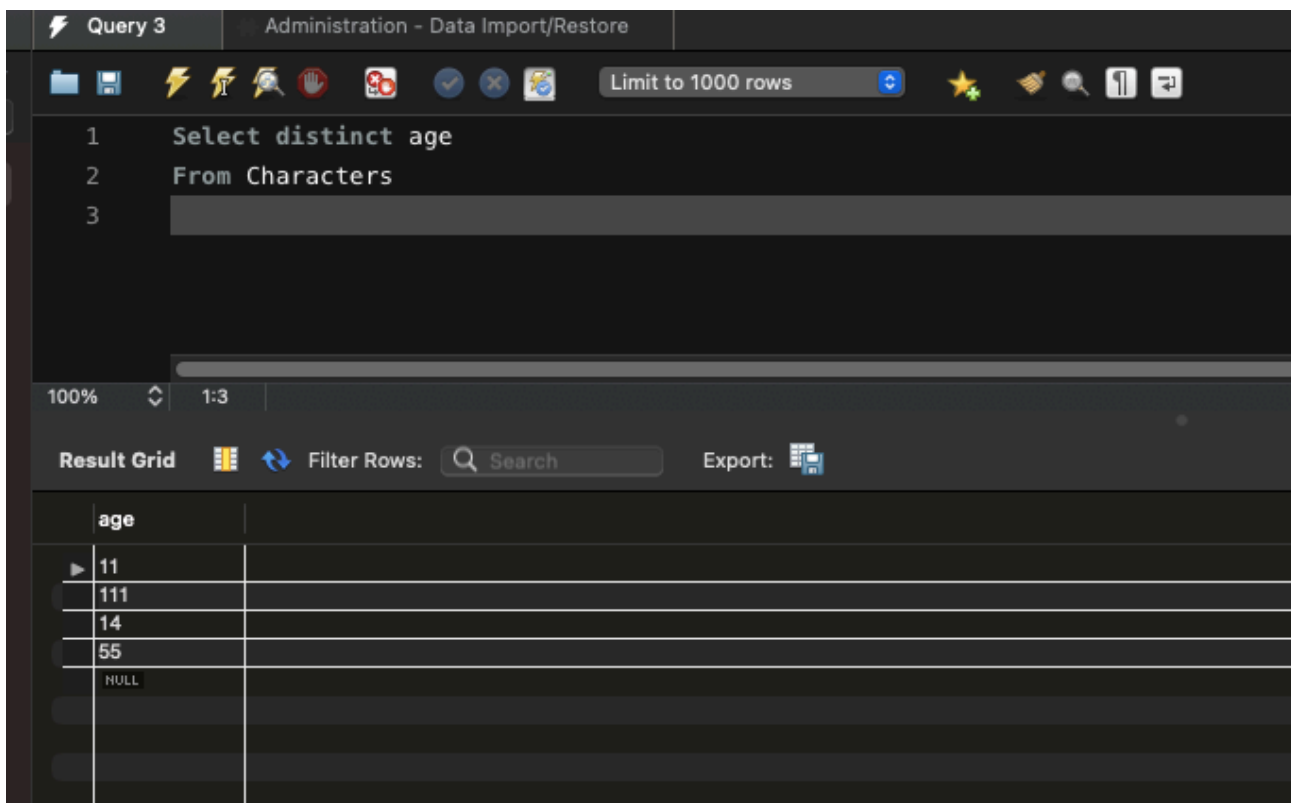
The screenshot shows a database query editor with the following SQL query:

```
1 Select fname,age
2 From Characters
3 Where age between 50 and 100
```

The result grid displays the following data:

fname	age
Severus	55

6. Выведите возраст всех персонажей так, чтобы среди них не было тех, у кого он одинаковый



The screenshot shows a database query editor with the following SQL query:

```
1 Select distinct age
2 From Characters
3
```

The result grid displays the following data:

age
11
111
14
55
NULL

7. Выведите всю информацию о персонажах, у которых faculty = Gryffindor и чей возраст больше 30 лет

The screenshot shows the PostgreSQL Query Editor interface. At the top, the title bar reads "Query 3" and "Administration - Data Import/Restore". The query editor contains the following SQL code:

```
1 select *
2 from characters
3 where faculty = 'Gryffindor' and age >30
4
5
```

Below the query editor, the "Result Grid" is displayed. It shows a table with 8 columns: char_id, fname, lname, age, faculty, patronus, book_id, and an empty column. The first row of data is highlighted in blue and contains the following values:

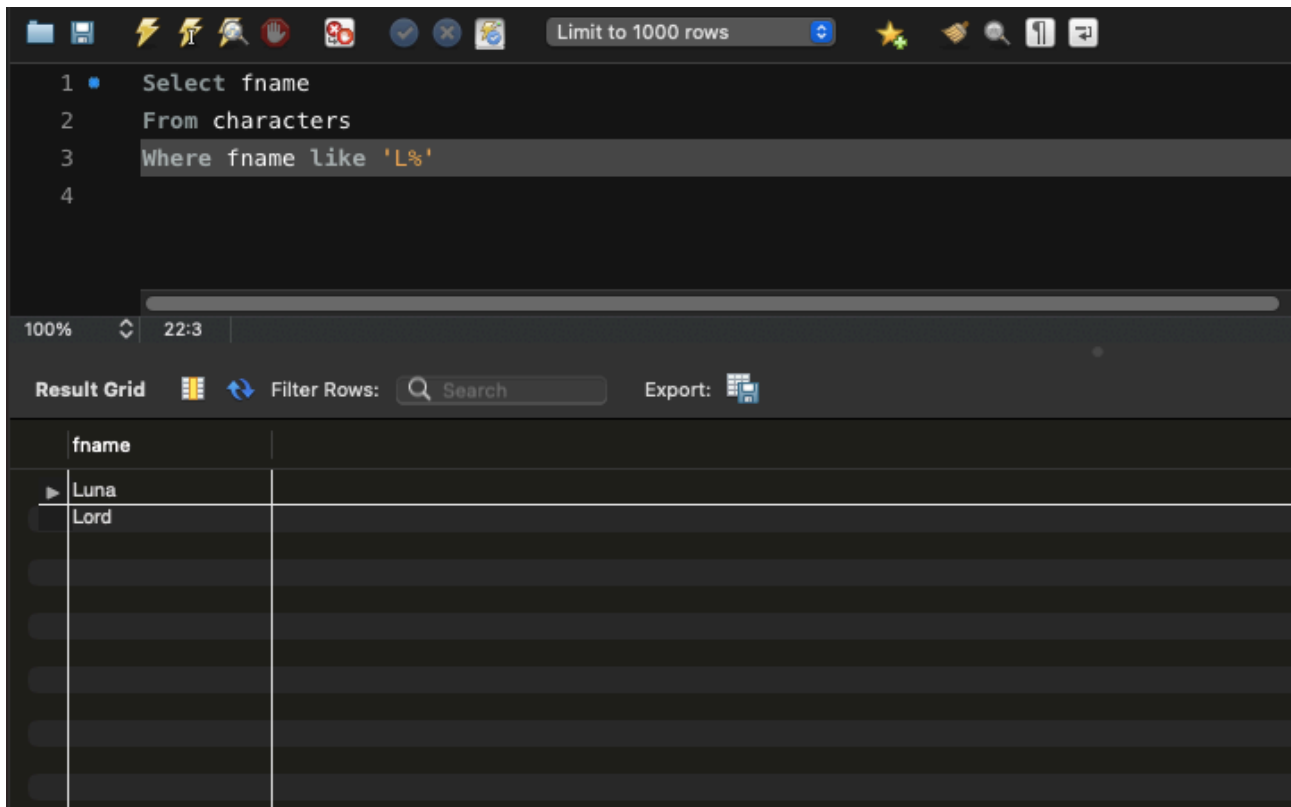
char_id	fname	lname	age	faculty	patronus	book_id	
7	Albus	Dumbledore	111	Gryffindor	Phoenix	2	

The interface also includes a toolbar with various icons for file operations, a "Limit to 1000 rows" dropdown, and a search bar.

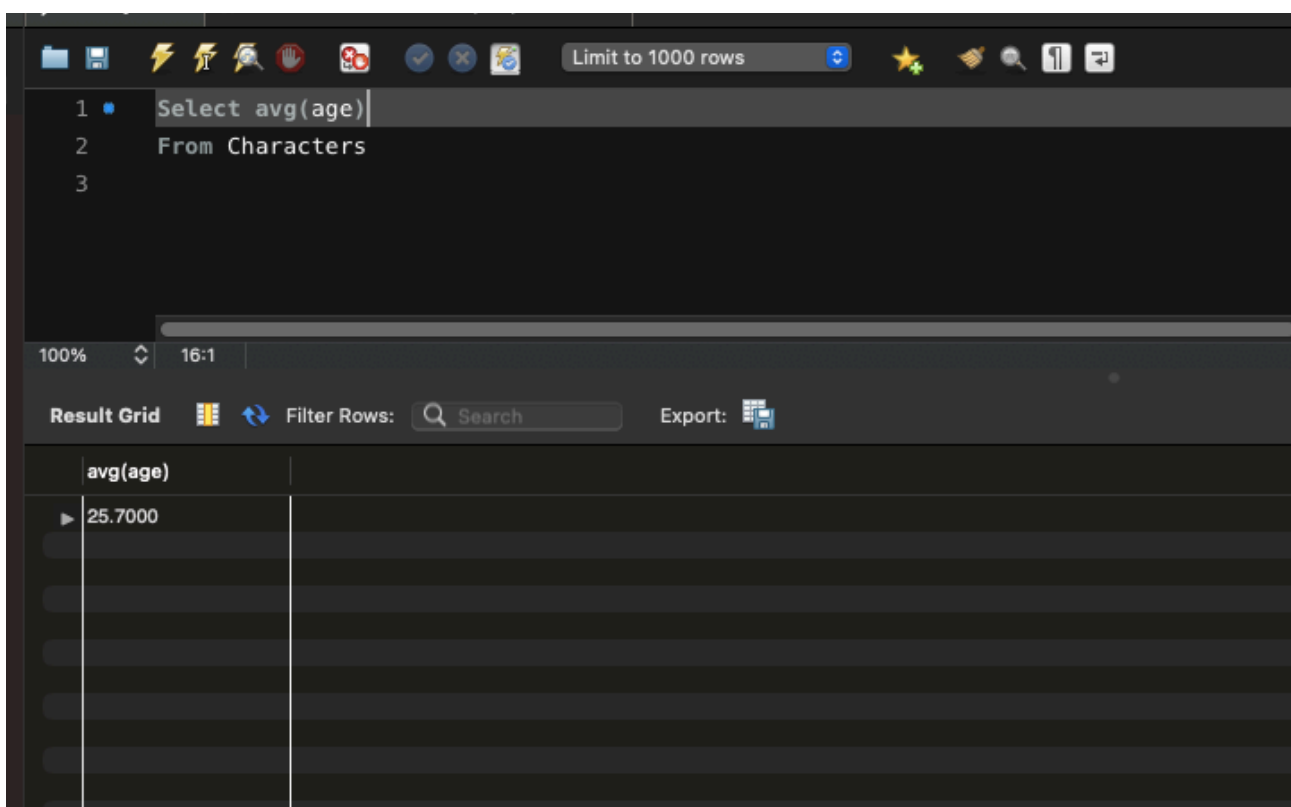
8. Выведите имена первых трех факультетов из таблицы, так чтобы факультеты не повторялись

[illegible]

9. Выведите имена всех персонажей, у которых имя начинается с 'H' и состоит из 5 букв, или чье имя начинается с 'L'



10. Посчитайте средний возраст всех персонажей



11. Удалите персонажа с ID = 11

```
1 Delete from characters
2 Where char_id = 11
```

12. Выведите фамилию всех персонажей, которые содержат в ней букву 'а'

Limit to 1000 rows

```
1 Select lname
2 From characters
3 Where lname like '%a%'
```

100% 24:3

Result Grid Filter Rows: Search Export:

	lname
▶	Granger
	Weasley
	Malfoy
	Crabbe
	Snape

13. Используйте псевдоним для того, чтобы временно заменить название столбца fname на Half-Blood Prince для реального принца-полукровки

```
1 select fname as Half_Blood_Prince
2 From Characters
3 Where fname = 'Severus'
```

100% 24:3

Result Grid Filter Rows: Search Export:

Half_Blood_...
Severus

14. Выведите id и имена всех патронусов в алфавитном порядке, при условии что они есть или известны

```
1 Select char_id, patronus
2 From Characters
3 Where not patronus='unkown' and patronus is not null
4 order by patronus asc
```

100% 22:4

Result Grid Filter Rows: Search Edit: Export/Import:

char_id	patronus
10	Doe
8	Hare
3	Jack Russell terrier
2	Otter
7	Phoenix
1	Stag
9	Unknown
NULL	NULL

15. Используя оператор IN, выведите имя и фамилию тех персонажей, у которых фамилия Crabbe, Granger или Diggory

[illegible]

16. Выведите минимальный возраст персонажа

The screenshot shows a SQL editor interface. The query entered is:

```
1 Select min(age)
2 From Characters
```

Below the query editor, there is a toolbar with options like "Result Grid", "Filter Rows", and "Search". The "Result Grid" is currently selected, displaying a table with the following data:

min(age)
11

SQL Query Editor:

```

1  Select fname, lname,
2  CASE
3  When Faculty = 'Gryffindor' THEN 'Godric'
4  When Faculty = 'Slytherin' THEN 'Salazar'
5  When faculty = 'Ravenclaw' THEN 'Rowena'
6  When faculty = 'Hufflepuff' then 'Helga'
7  Else 'Muggle'
8  END AS FOUNDERS
9  From characters;

```

Result Grid:

	fname	lname	FOUNDERS
▶	Harry	Potter	Godric
▶	Hermione	Granger	Godric
▶	Ron	Weasley	Godric
▶	Draco	Malfoy	Salazar
▶	Vincent	Crabbe	Salazar
▶	Gregory	Goyle	Salazar
▶	Albus	Dumbledore	Godric
▶	Luna	Lovegood	Rowena
▶	Cedric	Diggory	Helga
▶	Severus	Snape	Salazar

Result Grid:

	count(char_id)	faculty
▶	4	Gryffindor
▶	4	Slytherin

17. Используя оператор [UNION](#) выберите имена из таблицы characters и названия книг из таблицы library

Система уходит в ошибку

18. Используя оператор [HAVING](#) посчитайте количество персонажей на каждом факультете, оставив только те факультеты, где количество студентов больше 1

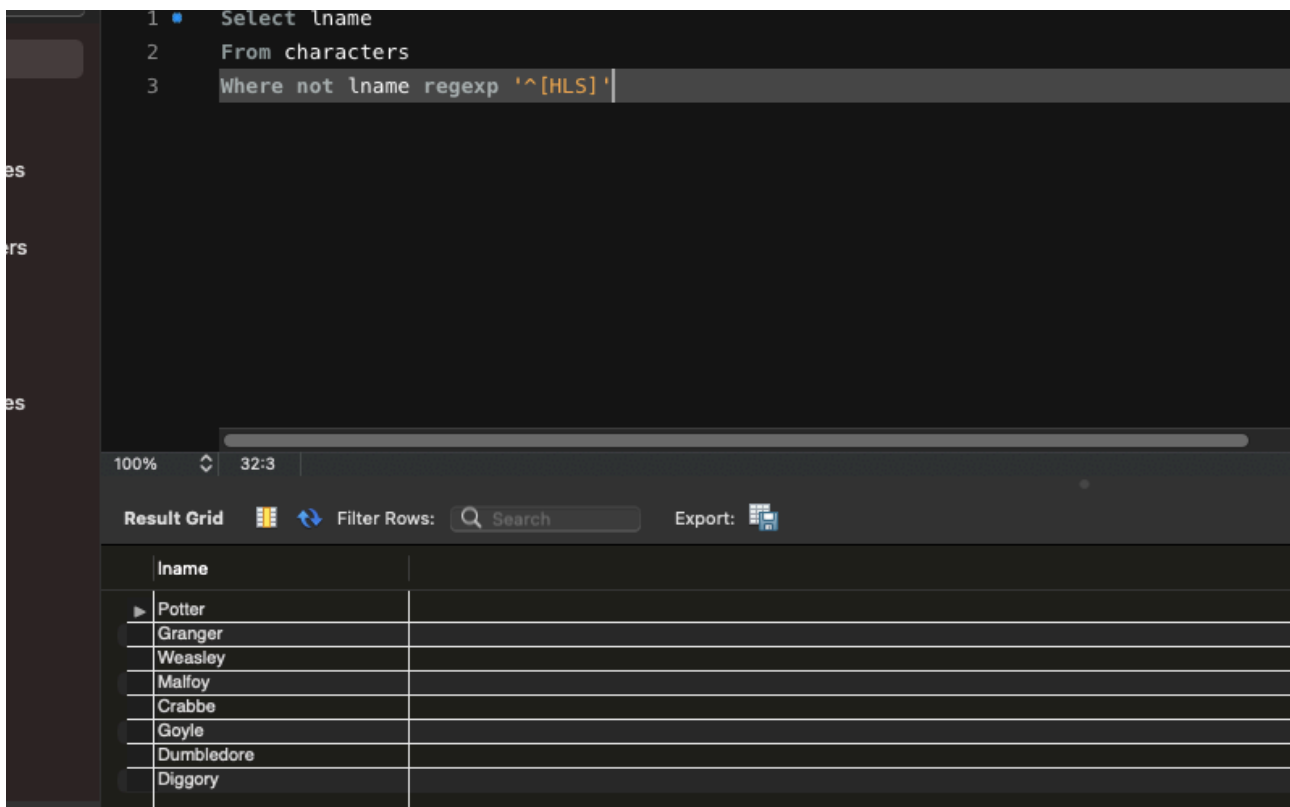
19. Используя оператор [CASE](#) опишите следующую логику:
Выведите имя и фамилию персонажа, а также следующий текстовое сообщение:

Если факультет Gryffindor, то в консоли должно выводиться Godric

Если факультет Slytherin, то в консоли должно вывестись Salazar
Если факультет Ravenclaw, то в консоли должно вывестись Rowena
Если факультет Hufflepuff, то в консоли должно вывестись Helga
Если другая информация, то выводится Muggle

Для сообщения используйте псевдоним Founders

1. [Используя регулярное выражение](#) найдите фамилии персонажей, которые не начинаются с букв H, L или S и выведите их



```
1 • Select lname
2   From characters
3  Where not lname regexp '^[HLS]'
```

100% 32:3

Result Grid Filter Rows: Search Export:

lname	
Potter	
Granger	
Weasley	
Malfoy	
Crabbe	
Goyle	
Dumbledore	
Diggory	