

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

The screenshot shows the PostgreSQL query editor interface. The query is as follows:

```
1 SELECT * FROM characters;
2
3 SELECT fname, lname, patronus
4 FROM characters
5 WHERE patronus IS NOT NULL AND NOT patronus = 'Unknown';
```

The result grid shows the following data:

fname	lname	patronus
Harry	Potter	Stag
Hermione	Granger	Otter
Ron	Weasley	Jack Russell terrier
Albus	Dumb...	Phoenix
Luna	Loveg...	Hare
Severus	Snape	Do

The bottom of the interface shows the Action Output with the following details:

	Time	Action	Response	Duration / Fetch Time
✓ 43	21:54:45	SELECT char_id, patronus FROM characters WHERE patronus IS NOT NULL AND NOT patronus = 'Unknown' LIMIT...	6 row(s) returned	0.00072 sec / 0.0000...
✓ 44	21:55:54	SELECT fname, lname, patronus FROM characters WHERE patronus IS NOT NULL AND NOT patronus = 'Unknown'...	6 row(s) returned	0.00061 sec / 0.0000...

Query Completed

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

The screenshot shows the PostgreSQL query editor interface. The query is as follows:

```
1 SELECT lname
2 FROM characters
3 WHERE lname LIKE '%e';
```

The result grid shows the following data:

lname
Crabbe
Goyle
Dumbledore
Snape

The bottom of the interface shows the Action Output with the following details:

	Time	Action	Response	Duration / Fetch Time
✓ 43	21:54:45	SELECT char_id, patronus FROM characters WHERE patronus IS NOT NULL AND NOT patronus = 'Unknown' LIMIT...	6 row(s) returned	0.00072 sec / 0.0000...
✓ 44	21:55:54	SELECT fname, lname, patronus FROM characters WHERE patronus IS NOT NULL AND NOT patronus = 'Unknown'...	6 row(s) returned	0.00061 sec / 0.0000...

Query Completed

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

The screenshot shows a database query editor interface. The query is as follows:

```
1 SELECT SUM(Age)
2 FROM characters;
```

The result grid shows a single row with the value 257.

SU...
257

Query Completed

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

The screenshot shows a database query editor interface. The query is as follows:

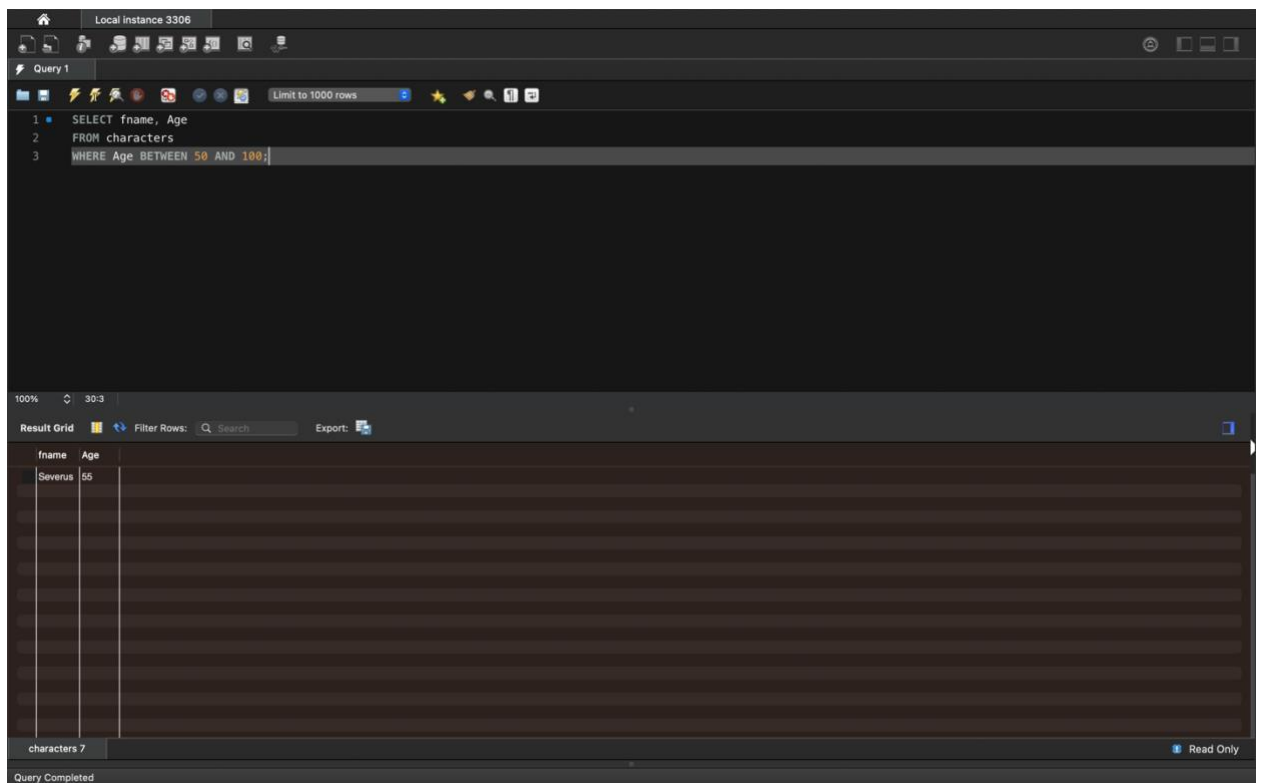
```
1 SELECT fname, lname, Age
2 FROM characters
3 ORDER BY Age DESC;
```

The result grid shows a list of characters ordered by age in descending order.

fname	lname	Age
Albus	Dumbledore	111
Severus	Snape	50
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

Query Completed

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



The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL query:

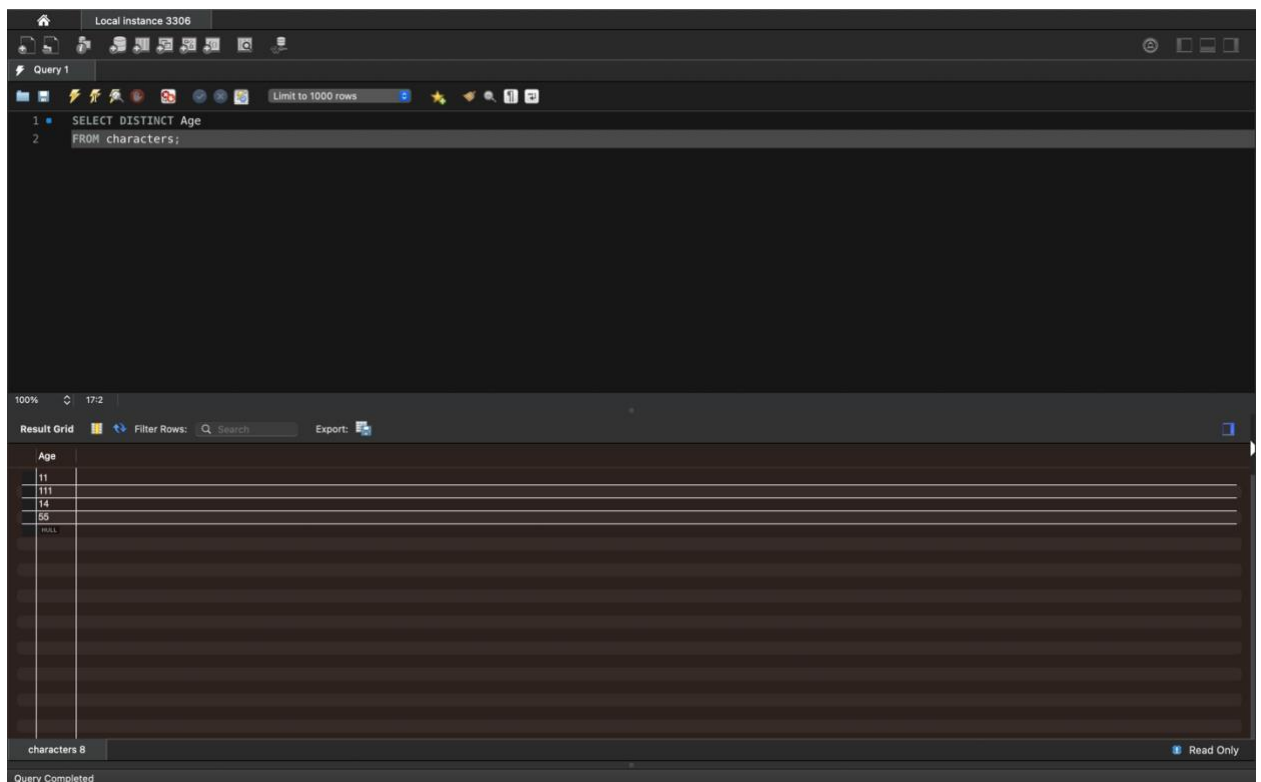
```
1 SELECT fname, Age
2 FROM characters
3 WHERE Age BETWEEN 50 AND 100;
```

The results grid shows the following data:

fname	Age
Sevens	55

The interface also shows a status bar at the bottom indicating "Query Completed" and a "Read Only" button.

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



The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL query:

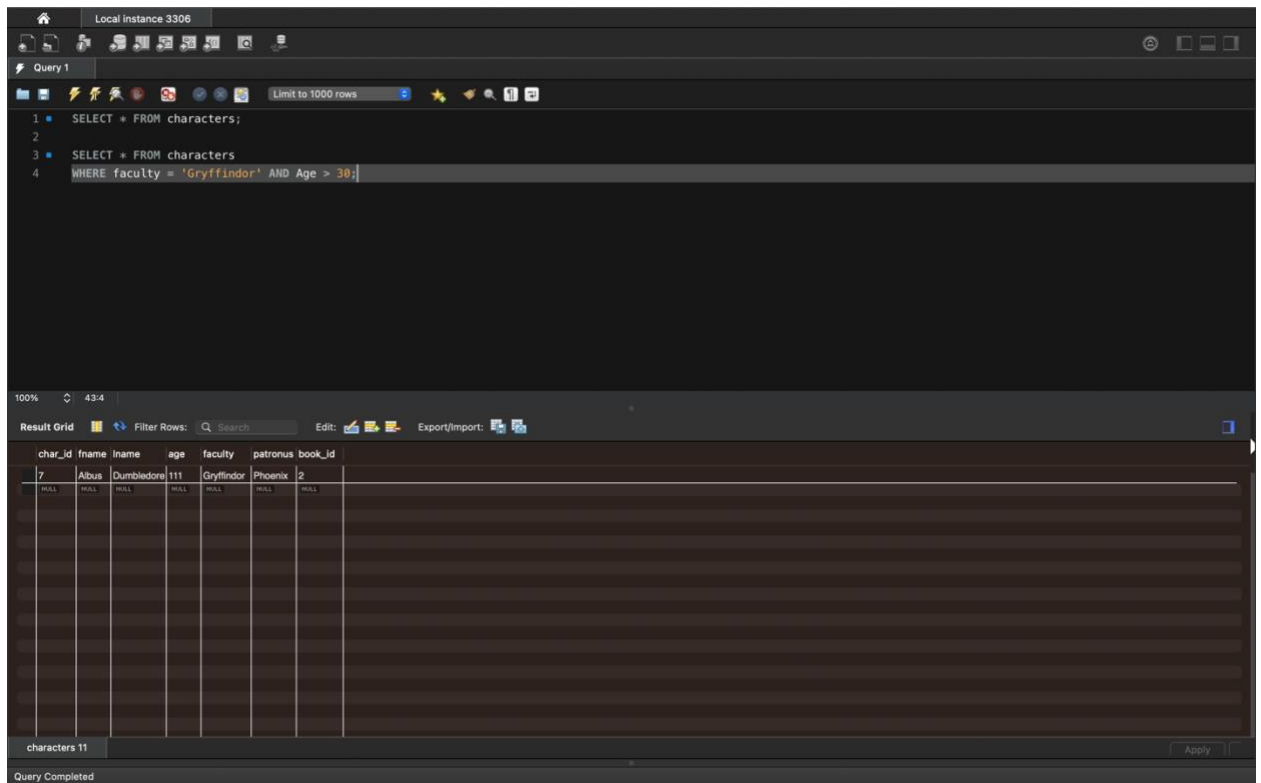
```
1 SELECT DISTINCT Age
2 FROM characters;
```

The results grid shows the following data:

Age
11
111
14
55
NULL

The interface also shows a status bar at the bottom indicating "Query Completed" and a "Read Only" button.

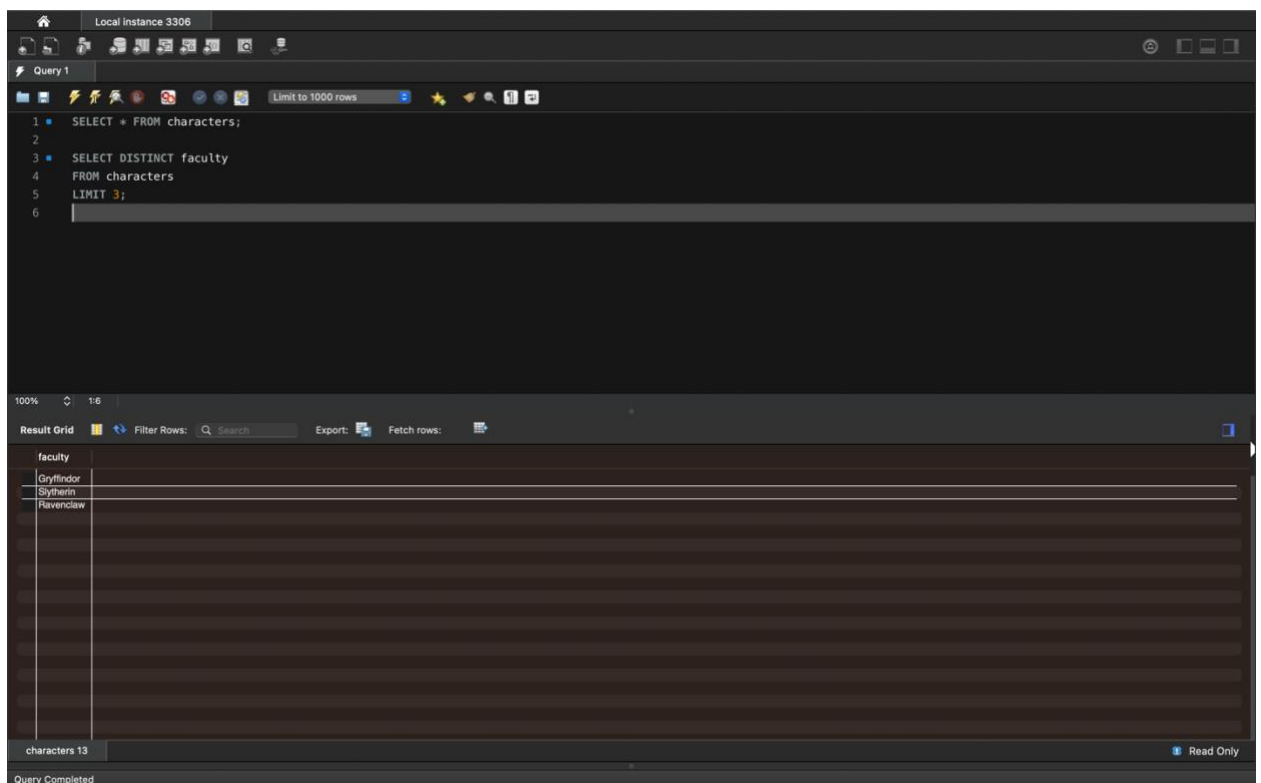
7. Выведите всю информацию о персонажах, у которых faculty = Gryffindor и чей возраст больше 30 лет



The screenshot shows a database client interface with a query editor and a result grid. The query is: `SELECT * FROM characters; WHERE faculty = 'Gryffindor' AND Age > 30;`. The result grid displays a table with 7 columns: char_id, fname, lname, age, faculty, patronus, and book_id. The first row shows a character with char_id 7, fname Albus, lname Dumbledore, age 111, faculty Gryffindor, patronus Phoenix, and book_id 2. The table is labeled 'characters 11' at the bottom.

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

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



The screenshot shows a database client interface with a query editor and a result grid. The query is: `SELECT DISTINCT faculty FROM characters LIMIT 3;`. The result grid displays a table with 1 column: faculty. The first three rows show the faculties Gryffindor, Slytherin, and Ravenclaw. The table is labeled 'characters 13' at the bottom.

faculty
Gryffindor
Slytherin
Ravenclaw

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

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```
1 SELECT * FROM characters;
2
3 SELECT fname
4 FROM characters
5 WHERE fname LIKE 'H____' OR fname LIKE 'L%';
6
```

The results grid displays the following data:

fname
Harry
Luna
Lord

The bottom panel shows the Action Output with the following entries:

	Time	Action	Response	Duration / Fetch Time
23	21:36:22	SELECT fname FROM characters WHERE fname LIKE 'H____' OR fname LIKE 'L%'	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to you...	0.00030 sec
24	21:37:08	SELECT fname FROM characters WHERE fname LIKE 'H____' OR fname LIKE 'L%' LIMIT 0, 1000	3 row(s) returned	0.00076 sec / 0.0000...

Query Completed

10. Посчитайте средний возраст всех персонажей

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```
1 SELECT * FROM characters;
2
3 SELECT AVG(Age)
4 FROM characters;
5
```

The results grid displays the following data:

AVG(Age)
25.7000

The bottom panel shows the Action Output with the following entries:

	Time	Action	Response	Duration / Fetch Time
24	21:37:08	SELECT fname FROM characters WHERE fname LIKE 'H____' OR fname LIKE 'L%' LIMIT 0, 1000	3 row(s) returned	0.00076 sec / 0.0000...
25	21:38:43	SELECT AVG(Age) FROM characters LIMIT 0, 1000	1 row(s) returned	0.00058 sec / 0.0000...

Query Completed

11. Удалите персонажа с ID = 11

The screenshot shows a database management tool interface with a SQL query editor and a results pane. The query is as follows:

```
1 SELECT * FROM characters;
2
3 SELECT * FROM characters
4 WHERE char_id = 11;
5
6 DELETE FROM characters
7 WHERE char_id = 11;
8
9 SELECT * FROM characters
10 WHERE char_id = 11;
```

The results pane shows a table with 7 columns: char_id, fname, lname, age, faculty, patronus, and book_id. The table is currently empty.

The Action Output pane shows the following actions:

Time	Action	Response	Duration / Fetch Time
29 21:40:33	DELETE FROM characters WHERE char_id = 11	1 row(s) affected	0.0098 sec
30 21:40:45	SELECT * FROM characters WHERE char_id = 11 LIMIT 0, 1000	0 row(s) returned	0.0011 sec / 0.00001...

Query Completed

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

The screenshot shows a database management tool interface with a SQL query editor and a results pane. The query is as follows:

```
1 SELECT * FROM characters;
2
3 SELECT lname
4 FROM characters
5 WHERE lname LIKE '%a%';
```

The results pane shows a table with 1 column: lname. The table contains 5 rows of data:

lname
Granger
Weasley
Malfoy
Crabbe
Snape

The Action Output pane shows the following actions:

Time	Action	Response	Duration / Fetch Time
32 21:41:51	SELECT * FROM characters WHERE lname LIKE '%a%' LIMIT 0, 1000	5 row(s) returned	0.00051 sec / 0.0000...
33 21:42:05	SELECT lname FROM characters WHERE lname LIKE '%a%' LIMIT 0, 1000	5 row(s) returned	0.00055 sec / 0.0000...

Query Completed

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

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

```
1 SELECT * FROM characters;  
2  
3 SELECT fname AS 'Half-Blood Prince'  
4 FROM characters  
5 WHERE fname = 'Severus';
```

The query is executed, and the result grid shows the following data:

Half-Blood Prince
Severus

The action output shows the following results:

Time	Action	Response	Duration / Fetch Time
21:46:44	SELECT fname AS 'Half-Blood Prince' FROM characters LIMIT 0, 1000	10 row(s) returned	0.00040 sec / 0.000...
21:46:55	SELECT fname AS 'Half-Blood Prince' FROM characters WHERE fname = 'Severus' LIMIT 0, 1000	1 row(s) returned	0.00042 sec / 0.0000...

Query Completed

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

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

```
1 SELECT * FROM characters;  
2  
3 SELECT char_id, patronus  
4 FROM characters  
5 WHERE patronus IS NOT NULL AND NOT patronus = 'Unknown'  
6 ORDER BY patronus ASC;
```

The query is executed, and the result grid shows the following data:

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

The action output shows the following results:

Time	Action	Response	Duration / Fetch Time
21:57:01	SELECT char_id, patronus FROM characters WHERE patronus IS NOT NULL AND NOT patronus = 'Unknown' LIMIT...	6 row(s) returned	0.00048 sec / 0.000...
21:57:50	SELECT char_id, patronus FROM characters WHERE patronus IS NOT NULL AND NOT patronus = 'Unknown' ORDE...	6 row(s) returned	0.0023 sec / 0.00001...

Query Completed

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

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

```
1 SELECT * FROM characters;
2
3 SELECT fname, lname
4 FROM characters
5 WHERE lname IN ('Crabbe', 'Granger', 'Diggory');
```

The query results are displayed in a table with the following data:

fname	lname
Hermione	Granger
Vincent	Crabbe
Cedric	Diggory

The bottom of the screenshot shows the Action Output table with the following entries:

	Time	Action	Response	Duration / Fetch Time
49	22:00:19	SELECT fname, lname FROM characters WHERE lname IN (Crabbe, Granger, Diggory) LIMIT 0, 1000	Error Code: 1054. Unknown column 'Crabbe' in 'where clause'	0.00040 sec
50	22:01:04	SELECT fname, lname FROM characters WHERE lname IN ('Crabbe', 'Granger', 'Diggory') LIMIT 0, 1000	3 row(s) returned	0.00062 sec / 0.0000...

16. Выведите минимальный возраст персонажа

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

```
1 SELECT * FROM characters;
2
3 SELECT MIN(Age)
4 FROM characters;
```

The query results are displayed in a table with the following data:

MIN(Age)
11

The bottom of the screenshot shows the Action Output table with the following entries:

	Time	Action	Response	Duration / Fetch Time
50	22:01:04	SELECT fname, lname FROM characters WHERE lname IN ('Crabbe', 'Granger', 'Diggory') LIMIT 0, 1000	3 row(s) returned	0.00062 sec / 0.0000...
51	22:01:58	SELECT MIN(Age) FROM characters LIMIT 0, 1000	1 row(s) returned	0.00032 sec / 0.00000...

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

The screenshot shows a SQL query in a database client. The query is as follows:

```
1 SELECT * FROM characters;
2 SELECT * FROM library;
3
4 SELECT fname FROM characters
5 UNION
6 SELECT book_name FROM library;
```

The result grid displays the following data:

fname
Harry
Hermione
Ron
Draco
Vincent
Gregory
Albus
Luna
Cedric
Severus
Hogwarts: A History
Quidditch Through The Ages
The Lockhart Collection
Moste Potente Potions
The Life And Lies Of Albus Dumbledore
Fantastic Beasts And Where To Find Them
The Tales Of Beedle The Bard
Advanced Potion-Making
A History Of Magic
Magical Water Plants Of The Highland Rocks

The Action Output shows the following results:

Time	Action	Response	Duration / Fetch Time
52 22:02:31	SELECT * FROM library LIMIT 0, 1000	13 row(s) returned	0.012 sec / 0.000011...
53 22:03:50	SELECT fname FROM characters UNION SELECT book_name FROM library	20 row(s) returned	0.0030 sec / 0.00001...

Query Completed

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

The screenshot shows a SQL query in a database client. The query is as follows:

```
1 SELECT * FROM characters;
2 SELECT * FROM library;
3
4 SELECT COUNT(faculty), faculty
5 FROM characters
6 GROUP BY faculty
7 HAVING COUNT(faculty) > 1;
```

The result grid displays the following data:

COUNT(faculty)	faculty
4	Gryffindor
4	Slytherin

The Action Output shows the following results:

Time	Action	Response	Duration / Fetch Time
55 22:07:39	SELECT COUNT(faculty) FROM characters GROUP BY faculty HAVING COUNT(faculty) > 1 LIMIT 0, 1000	2 row(s) returned	0.0028 sec / 0.00000...
56 22:07:58	SELECT COUNT(faculty), faculty FROM characters GROUP BY faculty HAVING COUNT(faculty) > 1 LIMIT 0, 1000	2 row(s) returned	0.00060 sec / 0.000...

Query Completed

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 выводиться Salazar

Если факультет Ravenclaw, то в консоли должно выводиться Rowena

Если факультет Hufflepuff, то в консоли должно выводиться Helga

Если другая информация, то выводится Muggle

Для сообщения используйте псевдоним Founders

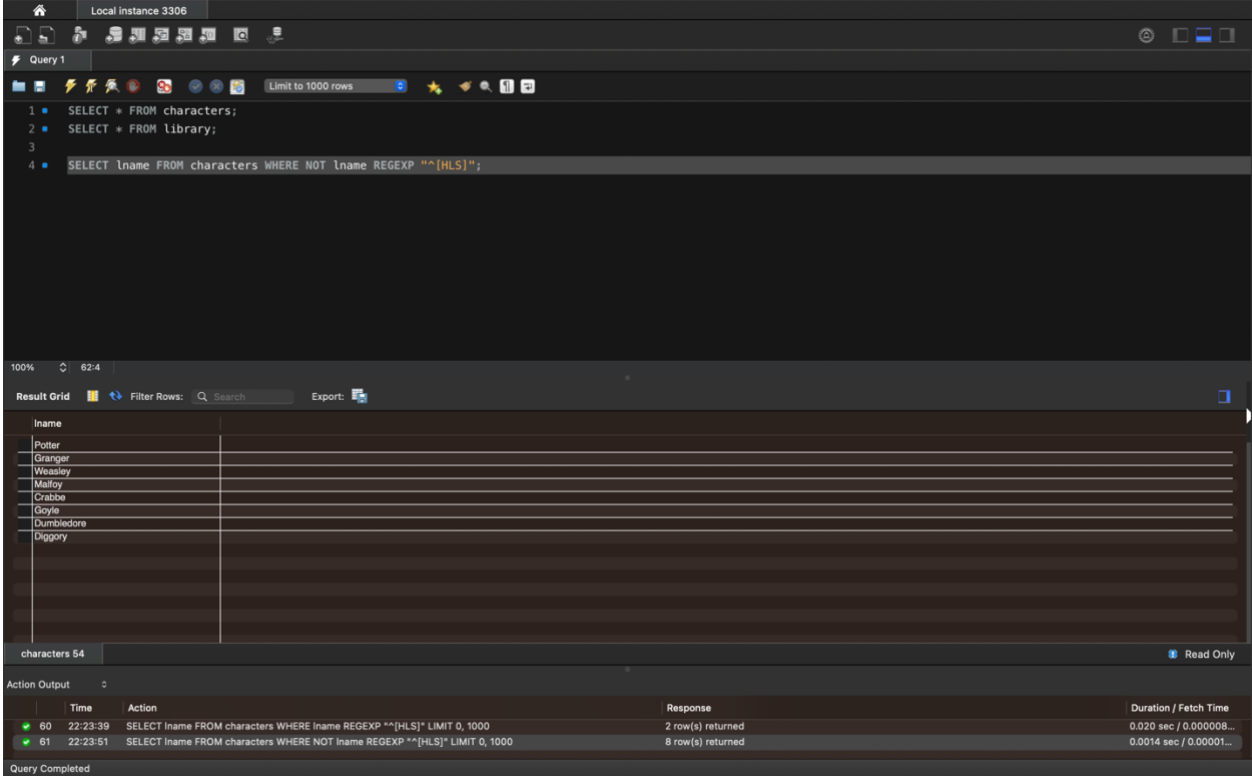
```
1 SELECT * FROM characters;
2 SELECT * FROM library;
3
4 SELECT fname, lname,
5 CASE
6   WHEN faculty = 'Gryffindor' THEN 'Godric'
7   WHEN faculty = 'Slytherin' THEN 'Salazar'
8   WHEN faculty = 'Ravenclaw' THEN 'Rowena'
9   WHEN faculty = 'Hufflepuff' THEN 'Helga'
10  ELSE 'Muggle'
11 END AS Founders
12 FROM characters;
```

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

Time	Action	Response	Duration / Fetch Time
58 22:15:42	SELECT fname, lname, CASE WHEN faculty = 'Gryffindor' THEN 'Godric' WHEN faculty = 'Slytherin' THEN 'Salazar'	10 row(s) returned	0.0016 sec / 0.00001...
59 22:16:05	SELECT fname, lname, CASE WHEN faculty = 'Gryffindor' THEN 'Godric' WHEN faculty = 'Slytherin' THEN 'Salazar'	10 row(s) returned	0.00048 sec / 0.000...

Query Completed

20. Используя регулярное выражение найдите фамилии персонажей, которые не начинаются с букв H, L или S и выведите их



The screenshot shows a database query interface with a dark theme. The query editor at the top contains four lines of SQL code:

```
1 SELECT * FROM characters;  
2 SELECT * FROM library;  
3  
4 SELECT lname FROM characters WHERE NOT lname REGEXP "^[HLS]";
```

Below the query editor, the "Result Grid" displays the results of the fourth query. The first column is labeled "lname" and contains the following names:

lname
Potter
Granger
Weasley
Malfoy
Crabbe
Coyne
Dumbledore
Diggory

At the bottom, the "Action Output" section shows a log of the executed queries:

	Time	Action	Response	Duration / Fetch Time
✓ 60	22:23:39	SELECT lname FROM characters WHERE lname REGEXP "^[HLS]" LIMIT 0, 1000	2 row(s) returned	0.020 sec / 0.000008...
✓ 61	22:23:51	SELECT lname FROM characters WHERE NOT lname REGEXP "^[HLS]" LIMIT 0, 1000	8 row(s) returned	0.0014 sec / 0.00001...

The interface also includes a "Query Completed" status at the bottom left and a "Read Only" button at the bottom right.