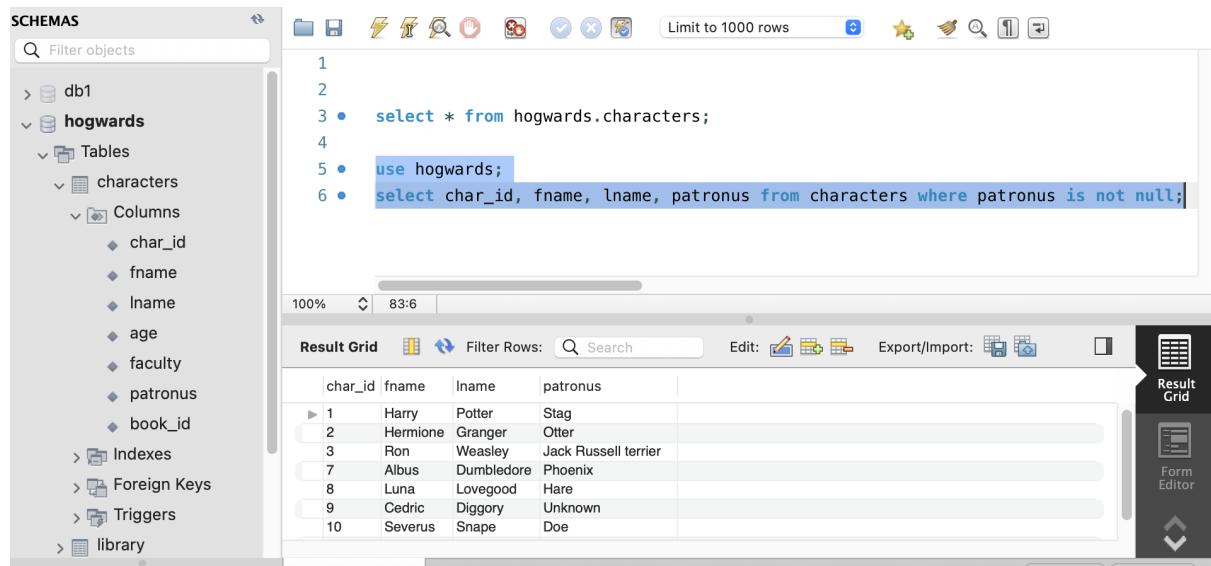


## Задание 1

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



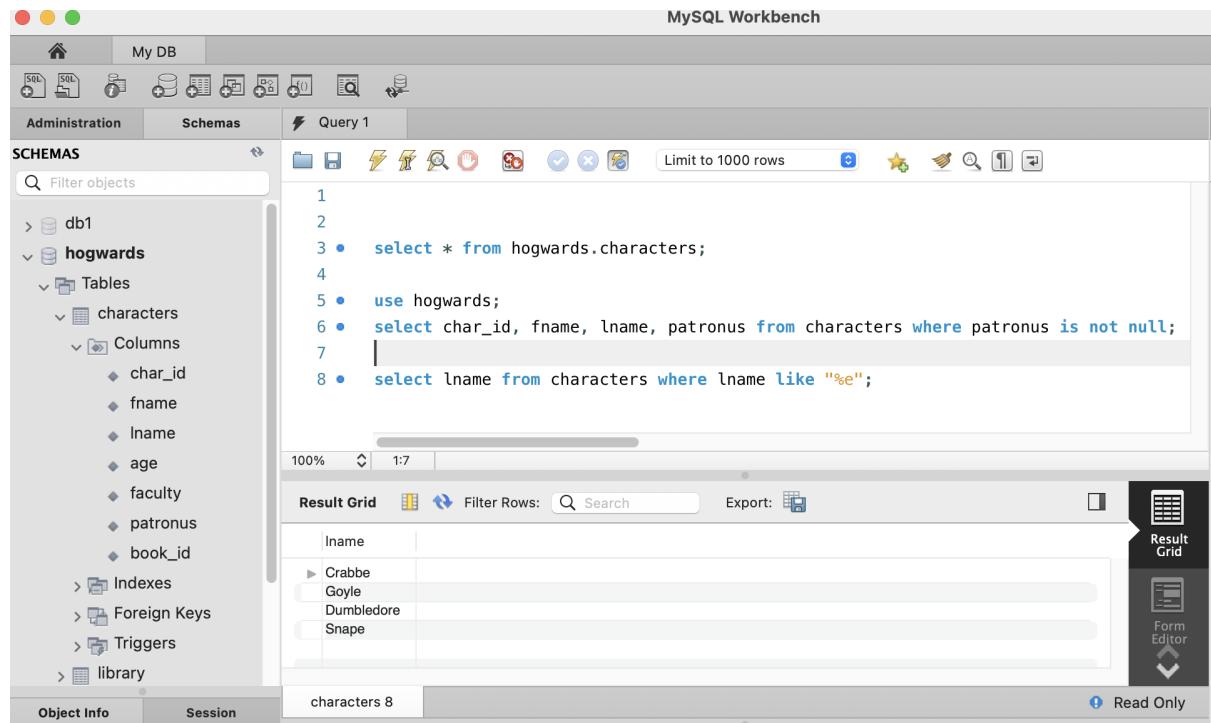
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** db1, hogwarts
- Tables:** characters
- Columns:** char\_id, fname, lname, patronus
- Query:**

```
1
2
3 • select * from hogwarts.characters;
4
5 • use hogwarts;
6 • select char_id, fname, lname, patronus from characters where patronus is not null;
```
- Result Grid:** Displays 10 rows of data:

char_id	fname	lname	patronus
1	Harry	Potter	Stag
2	Hermione	Granger	Otter
3	Ron	Weasley	Jack Russell terrier
7	Albus	Dumbledore	Phoenix
8	Luna	Lovegood	Hare
9	Cedric	Diggory	Unknown
10	Severus	Snape	Doe

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



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** db1, hogwarts
- Tables:** characters
- Columns:** char\_id, fname, lname, patronus
- Query:**

```
1
2
3 • select * from hogwarts.characters;
4
5 • use hogwarts;
6 • select char_id, fname, lname, patronus from characters where patronus is not null;
7
8 • select lname from characters where lname like "%e";
```
- Result Grid:** Displays 4 rows of data:

lname
Crabbe
Goyle
Dumbledore
Snape

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

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing 'db1' and 'hogwarts'. Under 'hogwarts', 'Tables' is expanded, showing 'characters'. 'Columns' is also expanded under 'characters', listing 'char\_id', 'fname', 'lname', 'age', 'faculty', 'patronus', and 'book\_id'. The main area contains a query editor with the following SQL code:

```

1
2
3 • select * from hogwarts.characters;
4
5 • use hogwarts;
6 • select char_id, fname, lname, patronus from characters where patronus is not null;
7
8 • select lname from characters where lname like "%e";
9 • select sum(age) from hogwarts.characters;

```

The results grid below shows one row with the value '257'.

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

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing 'db1' and 'hogwarts'. Under 'hogwarts', 'Tables' is expanded, showing 'characters'. 'Columns' is also expanded under 'characters', listing 'char\_id', 'fname', 'lname', 'age', 'faculty', 'patronus', and 'book\_id'. The main area contains a query editor with the following SQL code:

```

4
5 • use hogwarts;
6 • select char_id, fname, lname, patronus from characters where patronus is not null;
7
8 • select lname from characters where lname like "%e";
9 • select sum(age) from hogwarts.characters;
10 • select fname, lname, age from characters order by age desc;

```

The results grid below shows the following data:

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 the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view shows 'hogwarts' selected. In the main area, a query editor displays the following SQL code:

```

5 • use hogwarts;
6 • select char_id, fname, lname, patronus from characters where patronus is not null;
7
8 • select lname from characters where lname like "%e";
9 • select sum(age) from hogwarts.characters;
10 • select fname, lname, age from characters order by age desc;
11 • select fname, age from characters where age between 50 and 100;

```

The results grid shows one row of data:

fname	age
Severus	55

At the bottom right of the interface, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'.

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

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view shows 'hogwarts' selected. In the main area, a query editor displays the following SQL code:

```

6 • select char_id, fname, lname, patronus from characters where patronus is not null;
7
8 • select lname from characters where lname like "%e";
9 • select sum(age) from hogwarts.characters;
10 • select fname, lname, age from characters order by age desc;
11 • select fname, age from characters where age between 50 and 100;
12 • select distinct age from characters;

```

The results grid shows the distinct ages found in the characters table:

age
11
111
14
55
NULL

At the bottom right of the interface, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'.

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema for 'hogwarts'. The 'Tables' section contains a single table named 'characters'. Below it, the 'Columns' section lists the following columns: char\_id, fname, lname, age, faculty, patronus, and book\_id. The main area shows a query editor with the following SQL code:

```
7  
8 • select lname from characters where lname like "%e";  
9 • select sum(age) from hogwarts.characters;  
10 • select fname, lname, age from characters order by age desc;  
11 • select fname, age from characters where age between 50 and 100;  
12 • select distinct age from characters;  
13 • select * from characters where faculty = "Gryffindor" and age >= 30;
```

The status bar at the bottom indicates a result count of 14 rows.

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

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view is open, showing the 'hogwarts' schema with its tables, columns, indexes, foreign keys, and triggers. The 'characters' table is selected, and its columns (char\_id, fname, lname, age, faculty, patronus, book\_id) are listed. The main area contains a query editor titled 'Query 1' with the following SQL code:

```
8 • select lname from characters where lname like "%e";
9 • select sum(age) from hogwarts.characters;
10 • select fname, lname, age from characters order by age desc;
11 • select fname, age from characters where age between 50 and 100;
12 • select distinct age from characters;
13 • select * from characters where faculty = "Gryffindor" and age >= 30;
14 • select distinct faculty from characters limit 3;
```

The result grid below shows the output of the last query, which is 'faculty'. The results are:

faculty
Gryffindor
Slytherin
Ravenclaw

On the right, there are several tabs: 'Result Grid' (selected), 'Form Editor', and 'Field'. There are also icons for saving, running, and canceling the query.

9. Выведите имена всех персонажей, у которых имя начинается с ‘Н’ и состоит из 5 букв, или чье имя начинается с ‘Л’

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view shows the 'hogwarts' schema selected, with its tables ('characters'), columns ('char\_id', 'fname', 'lname', 'age', 'faculty', 'patronus', 'book\_id'), and other objects like indexes, foreign keys, triggers, and a library. The main area contains a query editor with the following SQL code:

```
9 • select sum(age) from hogwarts.characters;
10 • select fname, lname, age from characters order by age desc;
11 • select fname, age from characters where age between 50 and 100;
12 • select distinct age from characters;
13 • select * from characters where faculty = "Gryffindor" and age >= 30;
14 • select distinct faculty from characters limit 3;
15 • select fname from characters where fname like "H___" or "L%";
```

The result grid below shows the output for the 15th query, which retrieves names starting with 'H' or 'L'. The first row is 'Harry'.

## 10. Посчитайте средний возраст всех персонажей

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view shows the 'hogwarts' schema selected, with its tables ('characters'), columns ('char\_id', 'fname', 'lname', 'age', 'faculty', 'patronus', 'book\_id'), and other objects like indexes, foreign keys, triggers, and a library. The main area contains a query editor with the following SQL code:

```
10 • select fname, lname, age from characters order by age desc;
11 • select fname, age from characters where age between 50 and 100;
12 • select distinct age from characters;
13 • select * from characters where faculty = "Gryffindor" and age >= 30;
14 • select distinct faculty from characters limit 3;
15 • select fname from characters where fname like "H___" or "L%";
16 • select avg(age) from hogwarts.characters;
```

The result grid below shows the output for the 16th query, which calculates the average age of all characters. The result is 25.7000.

## 11. Удалите персонажа с ID = 11

```

Administration Schemas Query 1
SCHEMAS
  Filter objects
  db1
  hogwards
    Tables
      characters
        Columns
          char_id
          fname
          lname
          age
          faculty
          patronus
          book_id
      Indexes
      Foreign Keys
      Triggers
      library
Object Info Session
No object selected
  Time Action Response Duration / Fetch Time
  34 13:50:23 select fname, age from... 1 row(s) returned 0.0057 sec / 0.00001...
  35 13:53:20 select distinct age from... 5 row(s) returned 0.0061 sec / 0.0001...
  36 13:56:44 select * from characters... 1 row(s) returned 0.0058 sec / 0.00001...
  37 13:59:06 select * from characters... 3 row(s) returned 0.0035 sec / 0.00002...
  38 14:00:20 select distinct faculty fr... 3 row(s) returned 0.0019 sec / 0.00002...
  39 14:04:36 select fname from chara... 1 row(s) returned 0.0058 sec / 0.00001...
  40 14:06:14 select avg(age) from ho... 1 row(s) returned 0.0026 sec / 0.00001...
  41 14:34:00 delete from characters... 1 row(s) affected 0.016 sec
  42 14:35:06 select * from characters... 0 row(s) returned 0.0019 sec / 0.00001...

```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema structure for 'hogwards'. The main area contains a query editor with the following SQL code:

```

12 • select distinct age from characters;
13 • select * from characters where faculty = "Gryffindor" and age >= 30;
14 • select distinct faculty from characters limit 3;
15 • select fname from characters where fname like "H___" or "L%";
16 • select avg(age) from hogwarts.characters;
17 • delete from characters where char_id = 11; (arrow points here)
18 • select * from characters where char_id = 11;

```

The result grid shows a single row with all columns set to NULL. The action output table highlights the delete statement (row 41) and its execution details.

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

```

Administration Schemas Query 1
SCHEMAS
  Filter objects
  db1
  hogwards
    Tables
      characters
        Columns
          char_id
          fname
          lname
          age
          faculty
          patronus
          book_id
      Indexes
      Foreign Keys
      Triggers
      library
Object Info Session
No object selected
  Time Action Response Duration / Fetch Time
  13 13:50:23 select * from characters where faculty = "Gryffindor" and age >= 30; 0.0057 sec / 0.00001...
  14 13:53:20 select distinct faculty from characters limit 3; 0.0061 sec / 0.0001...
  15 13:56:44 select fname from characters where fname like "H___" or "L%"; 0.0058 sec / 0.00001...
  16 13:59:06 select avg(age) from hogwarts.characters; 0.0035 sec / 0.00002...
  17 14:00:20 delete from characters where char_id = 11; 0.0019 sec / 0.00002...
  18 14:04:36 select * from characters where char_id = 11; 0.0058 sec / 0.00001...
  19 14:34:00 select lname from characters where lname like "%a%"; 0.0026 sec / 0.00001...

```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema structure for 'hogwards'. The main area contains a query editor with the following SQL code:

```

13 • select * from characters where faculty = "Gryffindor" and age >= 30;
14 • select distinct faculty from characters limit 3;
15 • select fname from characters where fname like "H___" or "L%";
16 • select avg(age) from hogwarts.characters;
17 • delete from characters where char_id = 11;
18 • select * from characters where char_id = 11;
19 • select lname from characters where lname like "%a%"; (arrow points here)

```

The result grid shows the following data:

lname
Granger
Weasley
Malfoy
Crabbe
Snape

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

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree shows the 'hogwarts' schema selected. In the main area, a query editor displays the following SQL code:

```

14 • select distinct faculty from characters limit 3;
15 • select fname from characters where fname like "H___" or "L%";
16 • select avg(age) from hogwarts.characters;
17 • delete from characters where char_id = 11;
18 • select * from characters where char_id = 11;
19 • select lname from characters where lname like "%a%";
20 • select fname as "Half-Blood Prince" from characters where fname = "Severus";

```

The result grid below shows two rows:

	Half-Blood Prince
▶ Severus	

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

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree shows the 'hogwarts' schema selected. In the main area, a query editor displays the following SQL code:

```

15 • select fname from characters where fname like "H___" or "L%";
16 • select avg(age) from hogwarts.characters;
17 • delete from characters where char_id = 11;
18 • select * from characters where char_id = 11;
19 • select lname from characters where lname like "%a%";
20 • select fname as "Half-Blood Prince" from characters where fname = "Severus";
21 • select char_id, patronus from characters where patronus is not null order by patronus

```

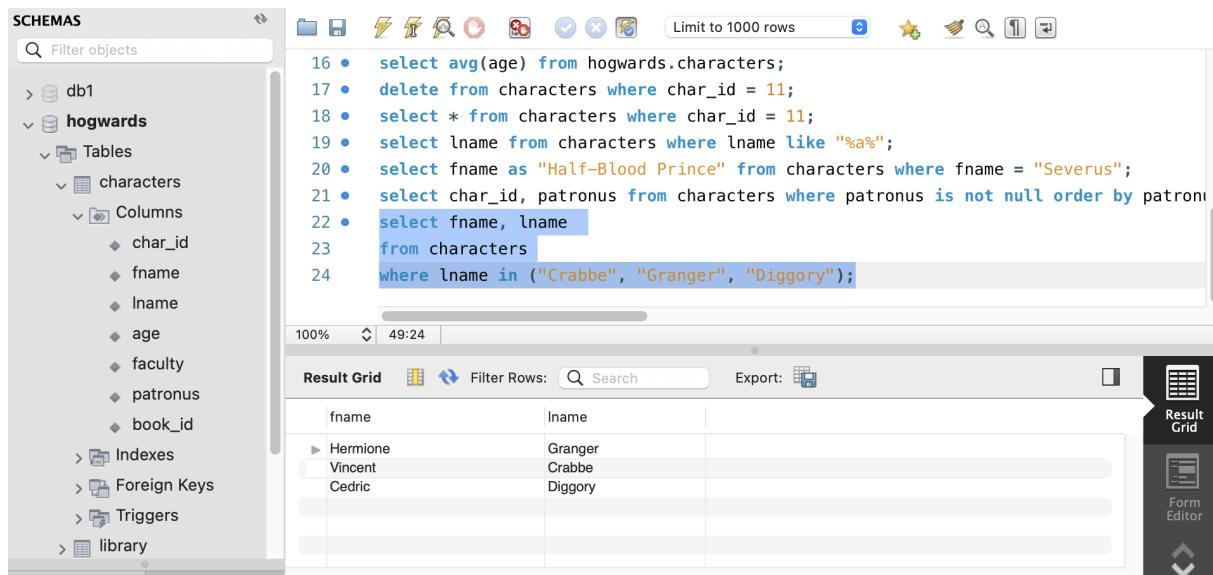
The result grid below shows the following data:

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

At the bottom, the 'Action Output' pane shows the following log:

Time	Action	Response	Duration / Fetch Time
39 14:04:36	select fname from characters	1 row(s) returned	0.0058 sec / 0.00001...
40 14:06:14	select avg(age) from hogwarts.characters	1 row(s) returned	0.0026 sec / 0.00001...
41 14:34:00	delete from characters...	1 row(s) affected	0.016 sec
42 14:35:06	select * from characters...	0 row(s) returned	0.0019 sec / 0.00001...
43 14:37:37	select lname from characters...	0 row(s) returned	0.0075 sec / 0.00001...
44 14:37:58	select fname from characters...	5 row(s) returned	0.0025 sec / 0.00001...
45 14:41:00	select fname from characters...	10 row(s) returned	0.012 sec / 0.000015...
46 14:42:26	select fname as "Half-Blood Prince" from characters where fname = "Severus"	1 row(s) returned	0.0083 sec / 0.00002...
47 14:45:23	select char_id, patronus...	7 row(s) returned	0.0057 sec / 0.00001...

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



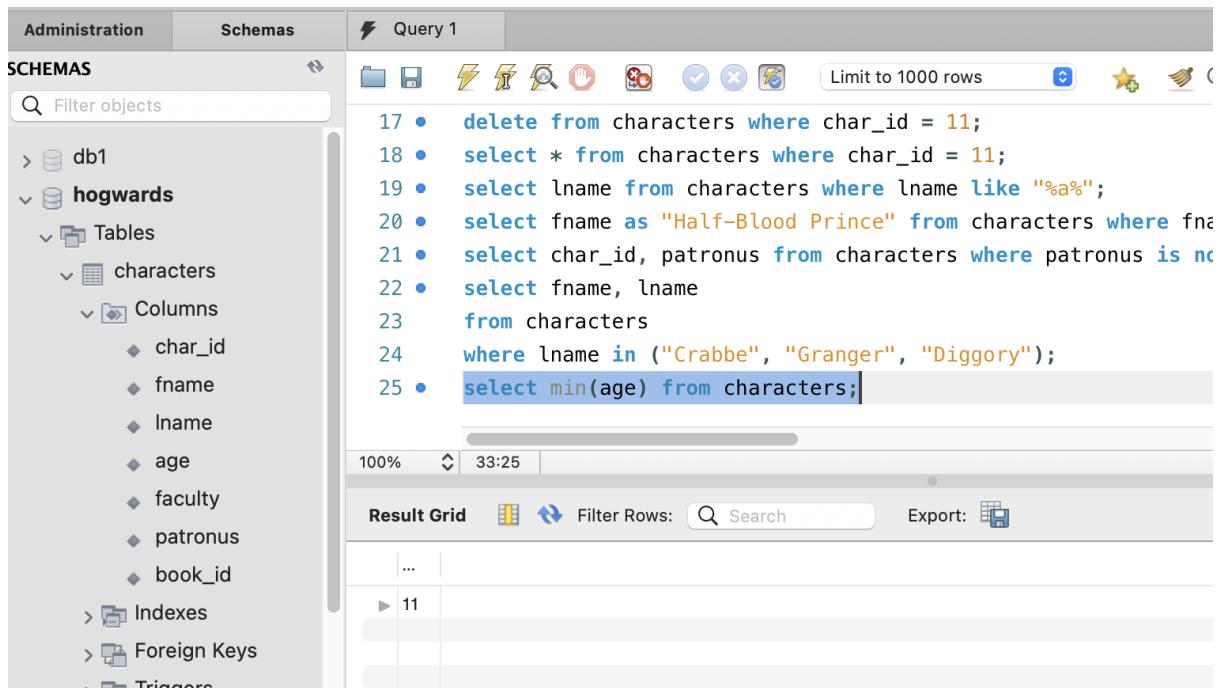
The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree shows 'db1' and 'hogwarts'. Under 'hogwarts', 'Tables' contains 'characters', which has 'Columns' including 'char\_id', 'fname', 'lname', 'age', 'faculty', 'patronus', and 'book\_id'. Other table categories like 'Indexes', 'Foreign Keys', and 'Triggers' are also listed. The main area displays a query editor with the following SQL code:

```
16 • select avg(age) from hogwarts.characters;
17 • delete from characters where char_id = 11;
18 • select * from characters where char_id = 11;
19 • select lname from characters where lname like "%a%";
20 • select fname as "Half-Blood Prince" from characters where fname = "Severus";
21 • select char_id, patronus from characters where patronus is not null order by patronus;
22 • select fname, lname
   from characters
  where lname in ("Crabbe", "Granger", "Diggory");
```

The result grid below shows the output of the last query:

fname	lname
Hermione	Granger
Vincent	Crabbe
Cedric	Diggory

16. Выведите минимальный возраст персонажа



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree shows 'db1' and 'hogwarts'. Under 'hogwarts', 'Tables' contains 'characters', which has 'Columns' including 'char\_id', 'fname', 'lname', 'age', 'faculty', 'patronus', and 'book\_id'. Other table categories like 'Indexes', 'Foreign Keys', and 'Triggers' are also listed. The main area displays a query editor with the following SQL code:

```
17 • delete from characters where char_id = 11;
18 • select * from characters where char_id = 11;
19 • select lname from characters where lname like "%a%";
20 • select fname as "Half-Blood Prince" from characters where fname = "Severus";
21 • select char_id, patronus from characters where patronus is not null order by patronus;
22 • select fname, lname
   from characters
  where lname in ("Crabbe", "Granger", "Diggory");
23 •
24 •
25 • select min(age) from characters;
```

The result grid below shows the output of the last query:

...
11

17. Используя оператор UNION выберите имена из таблицы characters и названия книг из таблицы library

```

20 • select fname as "Half-Blood Prince" from characters where fname = "Severus";
21 • select char_id, patronus from characters where patronus is not null order by patronus;
22 • select fname, lname
23   from characters
24   where lname in ("Crabbe", "Granger", "Diggory");
25 • select min(age) from characters;
26 • select fname from characters
27 union
28 select book_name from library;

```

Result Grid

fname
Gregory
Albus
Luna
Cedric
Severus
Hogwarts: A History
Quidditch Through The Ages
The Lockhart Collection
Moste Potente Potions
The Life And Lies Of Albus D...
Fantastic Beasts And Where...
The Tales Of Beedle The Bard

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

```

24   where lname in ("Crabbe", "Granger", "Diggory");
25 • select min(age) from characters;
26 • select fname from characters
27 union
28 select book_name from library;
29 • select count(char_id), faculty
30   from characters
31   group by faculty
32   having count(char_id) > 1;

```

Result Grid

count(char_i... faculty
4 Gryffindor
4 Slytherin

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

Если факультет Gryffindor, то в консоли должно вывестись 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

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view shows a database named 'hogwarts' with a single table 'characters'. The 'Tables' node under 'hogwarts' has 'characters' expanded, showing columns: char\_id, fname, lname, Iname, age, faculty, patronus, book\_id. Below this are 'Indexes', 'Foreign Keys', 'Triggers', and a 'library' node. At the bottom left are tabs for 'Object Info' and 'Session', with 'Column: book\_id' selected. The main area contains a SQL editor with the following query:

```
32     having count(char_id) > 1;
33 •   select fname, lname,
34     case
35       when faculty = "Gryffindor" then "Godric"
36       when faculty = "Slytherin" then "Salazar"
37       when faculty = "Ravenclaw" then "Rowena"
38       when faculty = "Hufflepuff" then "Helga"
39     else "Muggle"
40   end
41   from characters;
```

The status bar at the bottom indicates '100%' and '16:41'. Below the SQL editor is a 'Result Grid' window showing the results of the query:

fname	Iname	case
Harry	Potter	when faculty = "Gryffindor" then "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

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema structure under 'SCHEMAS'. The 'characters' table is selected, showing its columns: char\_id, fname, lname, age, faculty, patronus, and book\_id. The main area contains a query editor with the following code:

```
when faculty = "Slytherin" then "Salazar"
when faculty = "Ravenclaw" then "Rowena"
when faculty = "Hufflepuff" then "Helga"
else "Muggle"
end
from characters;

select * from characters where lname not regexp '^h' and '^l' and '^s';
```

The result grid below shows the first row of data:

char_id	fname	lname	age	faculty	patronus	book_id
NULL	NULL	NULL	NULL	NULL	NULL	NULL

A status bar at the bottom indicates '100%' completion and a duration of '72:43'.

## Задание 2

1. Выведите имя, фамилию персонажей и название книги, которая на них числится

The screenshot shows the MySQL Workbench interface. On the left, the schema browser displays the 'hogwarts' database with its tables: 'characters', 'Columns', 'Indexes', 'Foreign Keys', 'Triggers', and 'library'. The 'characters' table is currently selected. Below it, the 'Object Info' tab is active. The main area contains a query editor with the following SQL code:

```
when faculty = "Hufflepuff" then "Helga"
else "Muggle"
end
from characters;
select * from characters where lname not regexp "^\h" and "^\l" and "^\s";
use hogwarts;
show tables;
select * from library;
select characters.fname, characters.lname, library.book_name
from characters
inner join library on characters.char_id=library.char_id;
```

The results grid below shows the following data:

fname	lname	book_name
Herminione	Granger	A History Of Magic
Ron	Weasley	Advanced Potion-Making
Draco	Malfoy	Fantastic Beasts And Where To Find Them
Vincent	Crabbe	Fantastic Beasts And Where To Find Them
Gregory	Goyle	Hogwarts: A History
Albus	Dumbledore	Quidditch Through The Ages
Luna	Lovegood	Quidditch Through The Ages
Cedric	Diggory	The Lockhart Collection
Severus	Snape	Mnstr Potente Potions

On the right, there is a 'Result Grid' panel and a 'Form Editor' panel.

2. Выведите имя, фамилию персонажей и название книги, вне зависимости от того, есть ли у них книги или нет

```

SCHEMAS
Filter objects
> db1
hogwards
Tables
  > characters
    Columns
      char_id
      fname
      lname
      age
      faculty
      patronus
      book_id
  > Indexes
  > Foreign Keys
  > Triggers
  > library
Object Info Session
Columns:
char_id int AI PK
fname varchar(45)
lname varchar(45)

43 • select * from characters where lname not regexp "^(h|l|s)";
44
45 • use hogwarts;
46 • show tables;
47 • select * from library;
48 • select characters.fname, characters.lname, library.book_name
  from characters
  inner join library on characters.char_id=library.char_id;
51 • select characters.fname, characters.lname, library.book_name
  from characters
  left join library on characters.char_id=library.char_id;

Result Grid
Filter Rows: Search Export:
fname lname book_name
Harry Potter Magical Water Plants Of The Highland Rocks
Hermione Granger A History Of Magic
Ron Weasley Advanced Potion-Making
Draco Malfoy Fantastic Beasts And Where To Find Them
Vincent Crabbe Fantastic Beasts And Where To Find Them
Gregory Goyle Hogwarts: A History
Albus Dumbledore Quidditch Through The Ages
Luna Lovegood Quidditch Through The Ages
Cedric Diggory The Lockhart Collection
Severus Snape Moste Potente Potions

```

3. Выведите название книги и имя патронуса, вне зависимости от того, есть ли информация о держателе книги в таблице или нет

```

SCHEMAS
Filter objects
> db1
hogwards
Tables
  > characters
    Columns
      char_id
      fname
      lname
      age
      faculty
      patronus
      book_id
  > Indexes
  > Foreign Keys
  > Triggers
  > library
Object Info Session
Columns:
char_id int AI PK
fname varchar(45)
lname varchar(45)

46 • show tables;
47 • select * from library;
48 • select characters.fname, characters.lname, library.book_name
  from characters
  inner join library on characters.char_id=library.char_id;
51 • select characters.fname, characters.lname, library.book_name
  from characters
  left join library on characters.char_id=library.char_id;
54 • select characters.patronus, library.book_name
  from characters
  right join library on characters.char_id=library.char_id;

Result Grid
Filter Rows: Search Export:
patronus book_name
Dumbledore Advanced Potion-Making
Unknown The Lockhart Collection
Doe Moste Potente Potions
NULL The Life And Lies Of Albus Dumbledore
NULL Fantastic Beasts And Where To Find Them
NULL The Tales Of Beadle The Bard
Jack Russell terrier Advanced Potion-Making
Otter A History Of Magic
Stag Magical Water Plants Of The Highland Rocks
Hare Quidditch Through The Ages
NULL Magical Water Plants Of The Highland Rocks
NULL Fantastic Beasts And Where To Find Them

```

4. Выведите имя, фамилию, возраст персонажей и название книги, которая на них числится, при условии, что все владельцы книг должны быть старше 15 лет

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'Schemas' tab, the 'hogwarts' schema is selected. Within 'hogwarts', the 'characters' table is highlighted. The main area displays a query window with the following SQL code:

```

50     inner join library on characters.char_id=library.char_id;
51 •   select characters.fname, characters.lname, library.book_name
52     from characters
53     left join library on characters.char_id=library.char_id;
54 •   select characters.patronus, library.book_name
55     from characters
56     right join library on characters.char_id=library.char_id;
57 •   select characters.fname, characters.lname, characters.age, library.book_name
58     from characters
59     inner join library on characters.char_id=library.char_id
60     where age in(select age where age > 15);

```

The results grid below shows two rows of data:

fname	lname	age	book_name
Albus	Dumbledore	111	Quidditch Through The Ages
Severus	Snape	55	Moste Potente Potions

5. Выведите имя персонажа, название книги, дату выдачи и дату завершения, при условии, что он младше 15 лет и его патронус неизвестен

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'Schemas' tab, the 'hogwarts' schema is selected. Within 'hogwarts', the 'library' table is highlighted. The main area displays a query window with the following SQL code:

```

54 •   select characters.patronus, library.book_name
55     from characters
56     right join library on characters.char_id=library.char_id;
57 •   select characters.fname, characters.lname, characters.age, library.book_name
58     from characters
59     inner join library on characters.char_id=library.char_id
60     where age in(select age where age > 15);
61 •   select characters.fname, library.book_name, library.start_date, library.end_date
62     from characters
63     inner join library on characters.char_id=library.char_id
64     where age in(select age where age < 15) and patronus is null;

```

The results grid below shows three rows of data:

fname	book_name	start_date	end_date
Draco	Fantastic Beasts And Where To Find Them	2010-10-20	2020-10-20
Vincent	Fantastic Beasts And Where To Find Them	2006-06-20	2010-00-20
Gregory	Hogwarts: A History	2010-10-20	2020-10-20

6. Используя вложенный запрос количество книг, у которых end\_date больше, чем end\_date у Hermione

I completed this task in two steps. First, I got Hermione's end\_date.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree, which includes 'faculty', 'patronus', 'book\_id', 'library' (selected), 'Columns' (with 'lib\_id', 'char\_id', 'book\_name', 'start\_date', 'end\_date', 'book\_id'), and 'Indexes'. The main pane shows a query editor with the following SQL code:

```
59     inner join library on characters.char_id=library.char_id
60     where age in(select age where age > 15);
61 •  select characters.fname, library.book_name, library.start_date, library.end_date
62   from characters
63     inner join library on characters.char_id=library.char_id
64     where age in(select age where age < 15) and patronus is null;
65 •  select * from library;
66 •  select library.end_date
67   from characters
68     inner join library on characters.char_id=library.char_id
69     where fname = "Hermione";
70 •
```

The result grid shows one row with the value '2022-12-20' in the 'end\_date' column. The status bar at the bottom indicates 'Result 51'.

And second step:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree, which includes 'book\_id' (selected), 'Indexes', 'Foreign Keys', 'Triggers', 'library', 'Columns' (with 'lib\_id', 'char\_id', 'book\_name', 'start\_date', 'end\_date', 'book\_id'), and 'Indexes'. The main pane shows a query editor with the following SQL code:

```
64     inner join library on characters.char_id=library.char_id
65     where age in(select age where age < 15) and patronus is null;
66 •  select * from library;
67 •  select library.end_date
68   from characters
69     inner join library on characters.char_id=library.char_id
70     where fname = "Hermione";
71 •  select count(book_name) from library where end_date > "2022-12-20";
```

The result grid shows one row with the value '2' in the 'count(book\_name)' column. The status bar at the bottom indicates 'Result 51'.

7. С помощью вложенного запроса выведите имена всех патронусов, у которых владельцы старше возраста персонажа, у которого патронус Unknown

SCHEMAS

Filter objects

characters

- Columns
  - char\_id
  - fname
  - lname
  - age
  - faculty
  - patronus
  - book\_id
- Indexes
- Foreign Keys
- Triggers

library

- Columns
  - lib\_id
  - char\_id

Limit to 1000 rows

65 • select \* from library;  
66 • select library.end\_date  
from characters  
inner join library on characters.char\_id=library.char\_id  
where fname = "Hermione";  
67 • select count(book\_name) from library where end\_date > "2022-12-20";  
71 • use hogwarts;  
72 • select patronus from characters  
where age > any  
(select age from characters where patronus = "Unknown");  
75

100% 1:72

Result Grid Filter Rows: Search Export:

patronus
Phoenix
Doe

Result Grid

Form Editor