

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

```
SELECT fname, lname, patronus
FROM characters
WHERE NOT patronus = 'Unknown' AND patronus IS NOT NULL;
```

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

```
SELECT lname
FROM characters
WHERE lname LIKE '%e';
```

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

```
SELECT SUM(age)
FROM characters;
```

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

```
SELECT fname, lname, age
FROM characters
ORDER BY age DESC;
```

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

```
SELECT fname, age
FROM characters
WHERE age BETWEEN 50 AND 100;
```

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

```
SELECT DISTINCT age
FROM characters;
```

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

```
SELECT * FROM characters
WHERE faculty='Gryffindor' AND age >30;
```

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

```
SELECT DISTINCT faculty
FROM characters
LIMIT 3;
```

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

```
SELECT fname  
FROM characters  
WHERE fname LIKE 'H____' OR fname LIKE 'L%';
```

10. Посчитайте средний возраст всех персонажей

```
SELECT AVG (age)  
FROM characters;
```

11. Удалите персонажа с ID = 11

```
DELETE FROM characters  
WHERE char_id=11;
```

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

```
SELECT lname FROM characters  
WHERE lname LIKE '%a%';
```

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

```
SELECT fname as Half_Blood_Prince
FROM characters
WHERE fname = 'Severus';
```

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

```
SELECT char_id, patronus
FROM characters
WHERE patronus IS NOT NULL OR NOT patronus = 'Unknown'
ORDER BY patronus ASC;
```

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

```
SELECT fname, lname
FROM characters
WHERE lname IN ('Crabbe', 'Granger', 'Diggory');
```

16. Выведите минимальный возраст персонажа

```
SELECT MIN(age)
FROM characters;
```

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

```
SELECT fname FROM characters
UNION
SELECT book_name FROM library;
```

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

```
SELECT COUNT(char_id)
fname FROM characters
HAVING COUNT(char_id)>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
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

```
SELECT fname, lname,  
CASE  
  WHEN faculty = 'Gryffindor' THEN 'Godric'  
  WHEN faculty = 'Slytherin' THEN 'Salazar'  
  WHEN faculty = 'Ravenclaw' THEN 'Rowena'  
  WHEN faculty = 'Hufflepuff' THEN 'Helga'  
  ELSE 'Muggle'  
END AS Founders  
FROM characters;
```

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

```
SELECT lname  
FROM characters  
WHERE NOT lname REGEXP '^[HLS]';
```