

он-лайн курс на STEPİK

# Расширенные возможности SQL

Дальневосточный федеральный университет

Автор курса

**Озерова Галина Павловна**

## Оглавление

<b>1.1 Простые запросы на выборку, база данных склад</b>	<b>5</b>
1.1.1 Структура и наполнение базы данных «Склад»	5
1.1.2 Оператор CASE	8
Немного теории	8
1.1.3 Оператор CASE , пример	11
1.1.4 Вложенный оператор CASE	12
Немного теории	12
1.1.5 Оператор CASE и функция IF(), пример	15
1.1.6 Оператор CASE и функция IF(), пример	16
1.1.7 Вычисление выражений, FLOOR()	17
1.1.8 Функции работы с текстом, CHAR_LENGTH()	18
Немного теории	18
1.1.9 Функции работы с текстом, вывод константы	19
Немного теории	19
1.1.10 Функции работы с текстом CONCAT() и CONCAT_WS()	21
Немного теории	21
1.1.11 Функции работы с текстом LEFT() и RIGHT()	23
Немного теории	23
1.1.12 Функции работы с текстом, SUBSTRING_INDEX() и LOCATE()	25
Немного теории	25
<b>1.2 Простые запросы на выборку, база данных «Резюме»</b>	<b>28</b>
1.2.1 Структура и наполнение таблицы	28
1.2.2 Регулярные выражения, наличие текста в строке	31
Немного теории	31
1.2.3 Регулярные выражения, начало и конец строки	33
Немного теории	33
1.2.4 Регулярные выражения, наличие заданных символов в строке	35
Немного теории	35
1.2.5 Регулярные выражения, наличие интервала символов в строке	37
Немного теории	37
1.2.6 Регулярные выражения, отсутствие символов и интервала символов	40
Немного теории	40
1.2.7 Регулярные выражения, соответствие одному из шаблонов	43
Немного теории	43
1.2.8 Регулярные выражения, соответствие одному любому символу	46
Немного теории	46
1.2.9 Регулярные выражения, повторение символов 0 и более раз	48
Немного теории	48
1.2.10 Регулярные выражения, повторение символов один и более раз	51
Немного теории	51
1.2.11 Регулярные выражения, наличие или отсутствие символов	53
Немного теории	53
1.2.12 Регулярные выражения, специальный символ для выделения слова	56

Немного теории.....	56
<b>1.3 Простые запросы на выборку, база данных склад.....</b>	<b>59</b>
1.3.1 Структура и наполнение базы данных «Склад».....	59
1.3.2 Запросы с группировкой, MIN(), MAX() .....	61
1.3.3 Запросы с группировкой, COUNT(), SUM() .....	62
1.3.4 Запросы с группировкой, SUM() .....	62
1.3.5 Запросы с группировкой, AVG(), SUM() .....	63
1.3.6 Запросы с группировкой, GROUP_CONCAT() .....	63
Немного теории.....	63
1.3.7 Запросы с группировкой, GROUP_CONCAT() .....	65
1.3.8 Запросы с группировкой, алгоритм поиска первых/последних записей .....	65
1.3.9 Запросы с группировкой, алгоритм поиска первых/последних записей .....	66
<b>1.4 Запросы с группировкой, база данных «Резюме».....</b>	<b>67</b>
1.4.1 Структура и наполнение таблицы .....	67
1.4.2 Запросы с группировкой, функции DAY(), MONTH(), YEAR(), DAYOFYEAR(), MAKEDATE() .....	69
Немного теории.....	69
1.4.3 Запросы с группировкой, функции UNIX_TIMESTAMP(), FROM_UNIXTIME() .....	72
Немного теории.....	72
1.4.4 Запросы с группировкой, функции FROM_UNIXTIME() с форматированием .....	74
Немного теории.....	74
1.4.5 Запросы с группировкой, функции работы с датой, пример .....	77
1.4.6 Запросы с группировкой, алгоритм поиска лучшей/худшей записи .....	78
<b>1.5 Простые запросы на выборку, база данных склад.....</b>	<b>79</b>
1.5.1 Структура и наполнение базы данных «Склад».....	79
1.5.2 Операторы ALL и ANY .....	81
1.5.3 Вложенные запросы, оператор ALL .....	85
Немного теории.....	85
1.5.4 Вложенные запросы, оператор ANY .....	87
Немного теории.....	87
1.5.5 Вложенные запросы, оператор ANY .....	90
1.5.6 Табличные выражения, вложенные запросы .....	91
Немного теории.....	91
1.5.7. Вложенные запросы, табличные выражения, запросы без FROM .....	94
Немного теории.....	94
1.5.8 Несколько табличных выражений в одном запросе .....	96
Немного теории.....	96
1.5.9 Вложенные запросы и табличные выражения в одном запросе .....	98
<b>1.6 Вложенный запросы и табличные выражения, база данных «Резюме» .....</b>	<b>99</b>
1.6.1 Структура и наполнение таблицы .....	99
1.6.2 Вложенные запросы и табличные выражения, пример .....	101
1.6.3 Вложенные запросы и табличные выражения, пример .....	102
1.6.4 Вложенные запросы и табличные выражения, пример .....	103
1.6.5 Соединение таблиц, оператор UNION .....	104
Немного теории.....	104
1.6.6 Табличные выражения, оператор LIMIT .....	107

Немного теории.....	107
1.6.7 Табличные выражения, функции DATE_ADD(), DATE_SUB(), DAYOFWEEK().....	110
Немного теории.....	110
<b>1.7 Простые запросы на выборку, база данных склад.....</b>	<b>113</b>
1.7.1 Структура и наполнение базы данных «Склад».....	113
1.7.2 Запросы на обновление, табличные выражения .....	115
Немного теории.....	115
1.7.3 Запрос на обновление, табличные выражения, пример.....	118
1.7.4 Запрос на удаление, табличные выражения .....	119
Немного теории.....	119
1.7.5 Запрос на удаление, табличные выражения, пример.....	121
1.7.6 Запрос на удаление, табличные выражения, пример.....	122
1.7.7 Запрос на создание таблицы, табличные выражения .....	123
Немного теории.....	123
1.7.8 Создание таблицы .....	125
1.7.9 Запросы на добавление, табличные выражения.....	126
Немного теории.....	126
1.7.10 Запрос на создание таблицы, пример .....	129

## 1.1 Простые запросы на выборку, база данных склад

### 1.1.1 Структура и наполнение базы данных «Склад»

В этом уроке предлагается написать SQL запросы для таблицы `book`, в которой хранится информация о книгах на некотором складе. Все запросы будут формулироваться для таблицы `book`, которая создана следующим запросом:

```
CREATE TABLE book (  
  book_id INT PRIMARY KEY AUTO_INCREMENT,  
  title VARCHAR(50),  
  author VARCHAR(30),  
  price decimal(8,2),  
  amount INT );
```

Наполнение таблицы `book`:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
16	Загадочное происшествие	Агата Кристи	260.99	15

17	Капитанская дочка	Пушкин А.С.	199.99	5
18	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
21	Затерянный мир	Дойл Артур Конан	400.00	3
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
24	Герой нашего времени	Лермонтов М.Ю.	479.99	2
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	150.01	13

Дамп базы данных для этого урока:

```
CREATE TABLE book (
  book_id INT PRIMARY KEY AUTO_INCREMENT,
  title VARCHAR(50),
  author VARCHAR(30),
  price decimal(8,2),
  amount INT
);

INSERT INTO book (title, author, price, amount)
VALUES
('Мастер и Маргарита', 'Булгаков М.А.', 670.99, 3),
('Белая гвардия', 'Булгаков М.А.', 540.50, 5),
('Идиот', 'Достоевский Ф.М.', 460, 10),
('Братья Карамазовы', 'Достоевский Ф.М.', 799.01, 3),
('Игрок', 'Достоевский Ф.М.', 480.50, 10),
('Стихотворения и поэмы', 'Есенин С.А.', 650.00, 15),
('Таинственный остров', 'Жюль Верн', 169.99, 0),
('Пуаро ведет следствие', 'Агата Кристи', 272.50, 2),
('Евгений Онегин', 'Пушкин А.С.', 200.11, 8),
('Бородино', 'Лермонтов М.Ю.', 316.55, 10),
('Дубровский', 'Пушкин А.С.', 114.99, 7),
('Собачье сердце', 'Булгаков М.А.', 380.90, 6),
('Вокруг света за 80 дней', 'Жюль Верн', 201.01, 15),
('Смерть на Ниле', 'Агата Кристи', 250.12, 18),
('Убийства по алфавиту', 'Агата Кристи', 231.70, 9),
('Загадочное происшествие', 'Агата Кристи', 260.99, 15),
```

```
('Капитанская дочка', 'Пушкин А.С.', 199.99, 5),  
( 'Этюд в багровых тонах', 'Дойл Артур Конан', 215.00, 3),  
( 'Приключения Шерлока Холмса', 'Дойл Артур Конан', 590.50, 11),  
( 'Записки о Шерлоке Холмсе', 'Дойл Артур Конан', 470.20, 14),  
( 'Затерянный мир', 'Дойл Артур Конан', 400.00, 3),  
( 'Стихи', 'Лермонтов М.Ю.', 550.60, 6),  
( 'Поэмы', 'Лермонтов М.Ю.', 202.40, 8),  
( 'Герой нашего времени', 'Лермонтов М.Ю.', 479.99, 2),  
( 'Стихи', 'Пушкин А.С.', 600.50, 5),  
( 'Слово милой', 'Пушкин А.С.', 120.30, 12),  
( 'Поэмы', 'Пушкин А.С.', 630.50, 7),  
( 'Скрюченный домишко', 'Агата Кристи', 150.01, 13);
```

## 1.1.2 Оператор CASE

### Задание

автор - [Максим](#)

В конце месяца необходимо выполнить переоценку книг. Для тех книг, количество которых больше или равно 10, установить скидку 15%, округлить получившуюся цену до двух знаков после запятой, для остальных книг скидку не устанавливать (отобразить 0). Вывести автора, название книги, количество экземпляров книги, процент скидки (в виде числа) и новую цену книг с учетом скидки. Столбцы назвать **Автор**, **Книга**, **Количество**, **Скидка**, **Новая\_цена**. Информацию отсортировать сначала по автору в алфавитном порядке, а затем по названию книги также в алфавитном порядке.

### Структура таблицы

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

С помощью оператора **CASE** можно в зависимости от нескольких условий получить один из нескольких результатов.

Оператор **CASE** записывается в виде:

#### CASE

```
WHEN логическое_выражение_1 THEN выражение_1
WHEN логическое_выражение_2 THEN выражение_2
...
ELSE выражение_else
```

#### END

Раздел **ELSE** является необязательным.

Выполняется оператор **CASE** так:

- вычисляется **логическое\_выражение\_1**, если оно истинно, то результатом оператора является **выражение\_1**, если ложно - выполнение оператора продолжается;
- вычисляется **логическое\_выражение\_2**, если оно истинно, то результатом оператора является **выражение\_2**, если ложно - выполнение оператора продолжается;
- если все логические выражения оказались ложными, то результат оператора - **выражение\_else**

**CASE** можно использовать в **SELECT, UPDATE, DELETE, SET, WHERE, ORDER BY, HAVING** - всюду, где можно использовать выражения.

### Пример

Для каждой книги из таблицы **book** установим скидку следующим образом: если количество книг меньше 4, то скидка будет составлять 50% от цены, в противном случае 30%. Сумму скидки округлить до двух знаков после запятой.



Запрос (с использованием функции **IF()**):

```
SELECT title, amount, price,
       IF(amount < 4, ROUND(price * 0.5, 2),
          ROUND(price * 0.7, 2)) AS sale
FROM book;
```

Запрос (с использованием функции оператора **CASE**):

```
SELECT title, amount, price,
       CASE
         WHEN amount < 4 THEN ROUND(price * 0.5, 2)
         ELSE ROUND(price * 0.7, 2)
       END AS sale
FROM book;
```

### Результат

title	amount	price	sale
Мастер и Маргарита	3	670.99	335.50
Белая гвардия	5	540.50	378.35
Идиот	10	460.00	322.00
Братья Карамазовы	3	799.01	399.51
Игрок	10	480.50	336.35
Стихотворения и поэмы	15	650.00	455.00
Таинственный остров	0	169.99	85.00
Пуаро ведет следствие	2	272.50	136.25
Евгений Онегин	8	200.11	140.08
Бородино	10	316.55	221.59
Дубровский	7	114.99	80.49
Собачье сердце	6	380.90	266.63
Вокруг света за 80 дней	15	201.01	140.71
Смерть на Ниле	18	250.12	175.08
Убийства по алфавиту	9	231.70	162.19
Загадочное происшествие	15	260.99	182.69
Капитанская дочка	5	199.99	139.99
Этюд в багровых тонах	3	215.00	107.50
Приключения Шерлока Холмса	11	590.50	413.35
Записки о Шерлоке Холмсе	14	470.20	329.14
Затерянный мир	3	400.00	200.00
Стихи	6	550.60	385.42
Поэмы	8	202.40	141.68
Герой нашего времени	2	479.99	240.00
Стихи	5	600.50	420.35
Слово милой	12	120.30	84.21
Поэмы	7	630.50	441.35
Скрюченный домишко	13	150.01	105.01

**Текст задания** (чтобы не прокручивать страницу) :

В конце месяца необходимо выполнить переоценку книг. Для тех книг, количество которых больше или равно 10, установить скидку 15%, округлить получившуюся цену до двух знаков после запятой, для остальных книг скидку не устанавливать (отобразить 0). Вывести автора, название книги, количество экземпляров книги, процент скидки (в виде числа) и новую цену книг с учетом скидки. Столбцы назвать **Автор**, **Книга**, **Количество**, **Скидка**, **Новая\_цена**. Информацию отсортировать сначала по автору в алфавитном порядке, а затем по названию книги также в алфавитном порядке.

**ВАЖНО!** Числа 15 и 0 в запросе необходимо писать БЕЗ кавычек.

### Результат

Автор	Книга	Количество	Скидка	Новая_цена
Агата Кристи	Загадочное происшествие	15	15	221.84
Агата Кристи	Пуаро ведет следствие	2	0	272.50
Агата Кристи	Скрюченный домишко	13	15	127.51
Агата Кристи	Смерть на Ниле	18	15	212.60
Агата Кристи	Убийства по алфавиту	9	0	231.70
Булгаков М.А.	Белая гвардия	5	0	540.50
Булгаков М.А.	Мастер и Маргарита	3	0	670.99
Булгаков М.А.	Собачье сердце	6	0	380.90
Дойл Артур Конан	Записки о Шерлоке Холмсе	14	15	399.67
Дойл Артур Конан	Затерянный мир	3	0	400.00
Дойл Артур Конан	Приключения Шерлока Холмса	11	15	501.93
Дойл Артур Конан	Этюд в багровых тонах	3	0	215.00
Достоевский Ф.М.	Братья Карамазовы	3	0	799.01
Достоевский Ф.М.	Игрок	10	15	408.43
Достоевский Ф.М.	Идиот	10	15	391.00
Есенин С.А.	Стихотворения и поэмы	15	15	552.50
Жюль Верн	Вокруг света за 80 дней	15	15	170.86
Жюль Верн	Таинственный остров	0	0	169.99
Лермонтов М.Ю.	Бородино	10	15	269.07
Лермонтов М.Ю.	Герой нашего времени	2	0	479.99
Лермонтов М.Ю.	Поэмы	8	0	202.40
Лермонтов М.Ю.	Стихи	6	0	550.60
Пушкин А.С.	Дубровский	7	0	114.99
Пушкин А.С.	Евгений Онегин	8	0	200.11
Пушкин А.С.	Капитанская дочка	5	0	199.99
Пушкин А.С.	Поэмы	7	0	630.50
Пушкин А.С.	Слово милой	12	15	102.26
Пушкин А.С.	Стихи	5	0	600.50

Affected rows: 28

### 1.1.3 Оператор CASE , пример

#### Задание

автор - Елена Петухова

Снизить цены книг, которые стоят больше 400 рублей, на 20%. Вывести автора, название, цену, количество книг, рассчитанную скидку (столбец **sale\_20**) и цену книги со скидкой (**price\_sale**). Результаты округлить до двух знаков после запятой. Для тех книг, на которые скидка не действует, в последних двух столбцах вывести символ "-". Отсортировать информацию сначала по фамилии автора, а потом по названию книги.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Текст задания (чтобы не прокручивать страницу) :

Снизить цены книг, которые стоят больше 400 рублей, на 20%. Вывести автора, название, цену, количество книг, рассчитанную скидку (столбец **sale\_20**) и цену книги со скидкой (**price\_sale**). Результаты округлить до двух знаков после запятой. Для тех книг, на которые скидка не действует, в последних двух столбцах вывести символ "-". Отсортировать информацию сначала по фамилии автора, а потом по названию книги.

#### Результат

author	title	price	amount	sale_20	price_sale
Агата Кристи	Загадочное происшествие	260.99	15	-	-
Агата Кристи	Пуаро ведет следствие	272.50	2	-	-
Агата Кристи	Скрюченный домишко	150.01	13	-	-
Агата Кристи	Смерть на Ниле	250.12	18	-	-
Агата Кристи	Убийства по алфавиту	231.70	9	-	-
Булгаков М.А.	Белая гвардия	540.50	5	108.10	432.40
Булгаков М.А.	Мастер и Маргарита	670.99	3	134.20	536.79
Булгаков М.А.	Собачье сердце	380.90	6	-	-
Дойл Артур Конан	Записки о Шерлоке Холмсе	470.20	14	94.04	376.16
Дойл Артур Конан	Затерянный мир	400.00	3	-	-
Дойл Артур Конан	Приключения Шерлока Холмса	590.50	11	118.10	472.40
Дойл Артур Конан	Этюд в багровых тонах	215.00	3	-	-
Достоевский Ф.М.	Братья Карамазовы	799.01	3	159.80	639.21
Достоевский Ф.М.	Игрок	480.50	10	96.10	384.40
Достоевский Ф.М.	Идиот	460.00	10	92.00	368.00
Есенин С.А.	Стихотворения и поэмы	650.00	15	130.00	520.00
Жюль Верн	Вокруг света за 80 дней	201.01	15	-	-
Жюль Верн	Таинственный остров	169.99	0	-	-
Лермонтов М.Ю.	Бородино	316.55	10	-	-
Лермонтов М.Ю.	Герой нашего времени	479.99	2	96.00	383.99
Лермонтов М.Ю.	Поэмы	202.40	8	-	-
Лермонтов М.Ю.	Стихи	550.60	6	110.12	440.48
Пушкин А.С.	Дубровский	114.99	7	-	-
Пушкин А.С.	Евгений Онегин	200.11	8	-	-
Пушкин А.С.	Капитанская дочка	199.99	5	-	-
Пушкин А.С.	Поэмы	630.50	7	126.10	504.40
Пушкин А.С.	Слово милой	120.30	12	-	-
Пушкин А.С.	Стихи	600.50	5	120.10	480.40

## 1.1.4 Вложенный оператор CASE

### Задание

автор - [Сатурова Инна](#)

На распродаже размер скидки устанавливается в зависимости от количества экземпляров книги в магазине и от цены книги:

- для книг в остатке не менее 5 штук скидка 50%;
- для книг в остатке менее 5 штук скидка устанавливается в зависимости от цены: на книги не дешевле 700 руб скидка 20%, на остальные - 10%.

Вывести автора, название книги, количество, старая цена, скидка, цена со скидкой. Два последних столбца назвать **Скидка** и **Цена\_со\_скидкой**. Последний столбец округлить до двух знаков после запятой. Значение скидки вывести в виде строки текста (например, "15%"). Информацию отсортировать сначала по автору в алфавитном порядке, затем по убыванию цены книги со скидкой.

### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

В запросах можно использовать вложенные операторы **CASE**, которые можно располагать после THEN и ELSE:

**CASE**

```
WHEN логическое_выражение_1
  THEN
    CASE
      WHEN логическое_выражение_1_1 THEN выражение_1_1
      WHEN логическое_выражение_1_2 THEN выражение_1_2
      ...
      ELSE выражение_else_1
    END
  WHEN логическое_выражение_2 THEN выражение_2
  ...
  ELSE выражение_else
```

**END**

Раздел **ELSE** является необязательным.

### Пример

Для каждой книги из таблицы **book** установим скидку следующим образом:

- если книга написана Пушкиным или Лермонтовым, а количество книг их меньше 4, то скидка будет составлять 50% от цены, в противном случае 30%;
  - если книга написана Агатой Кристи, а количество книг меньше 5, то скидка будет составлять 40% от цены, в противном случае 20%;
  - для остальных книг скидку не устанавливать, оставить старую цену
- Сумму скидки округлить до двух знаков после запятой.

Запрос (с использованием функции `IF()`):

```
SELECT title, author, amount, price,
       IF(author = "Пушкин А.С." OR author = "Лермонтов М.Ю.",
          IF(amount < 4, ROUND(price * 0.5, 2),
             ROUND(price * 0.7, 2)),
          IF(author = "Агата Кристи",
             IF(amount < 5, ROUND(price * 0.6, 2),
                ROUND(price * 0.8, 2)), price
          )
       ) AS sale
FROM book;
```

Запрос (с использованием функции оператора `CASE`):

```
SELECT title, author, amount, price,
       CASE
         WHEN author = "Пушкин А.С." OR author = "Лермонтов М.Ю."
         THEN
           CASE
             WHEN amount < 4 THEN ROUND(price * 0.5, 2)
             ELSE ROUND(price * 0.7, 2)
           END
         WHEN author = "Агата Кристи"
         THEN
           CASE
             WHEN amount < 5 THEN ROUND(price * 0.6, 2)
             ELSE ROUND(price * 0.8, 2)
           END
         ELSE price
       END AS sale
FROM book;
```

## Результат

title	author	amount	price	sale
Мастер и Маргарита	Булгаков М.А.	3	670.99	670.99
Белая гвардия	Булгаков М.А.	5	540.50	540.50
Идиот	Достоевский Ф.М.	10	460.00	460.00
Братья Карамазовы	Достоевский Ф.М.	3	799.01	799.01
Игрок	Достоевский Ф.М.	10	480.50	480.50
Стихотворения и поэмы	Есенин С.А.	15	650.00	650.00
Таинственный остров	Жюль Верн	0	169.99	169.99
Пуаро ведет следствие	Агата Кристи	2	272.50	163.50
Евгений Онегин	Пушкин А.С.	8	200.11	140.08
Бородино	Лермонтов М.Ю.	10	316.55	221.59
Дубровский	Пушкин А.С.	7	114.99	80.49
Собачье сердце	Булгаков М.А.	6	380.90	380.90
Вокруг света за 80 дней	Жюль Верн	15	201.01	201.01
Смерть на Ниле	Агата Кристи	18	250.12	200.10
Убийства по алфавиту	Агата Кристи	9	231.70	185.36
Загадочное происшествие	Агата Кристи	15	260.99	208.79
Капитанская дочка	Пушкин А.С.	5	199.99	139.99
Этюд в багровых тонах	Дойл Артур Конан	3	215.00	215.00
Приключения Шерлока Холмса	Дойл Артур Конан	11	590.50	590.50
Записки о Шерлоке Холмсе	Дойл Артур Конан	14	470.20	470.20
Затерянный мир	Дойл Артур Конан	3	400.00	400.00
Стихи	Лермонтов М.Ю.	6	550.60	385.42
Поэмы	Лермонтов М.Ю.	8	202.40	141.68

Герой нашего времени	Лермонтов М.Ю.	2	479.99	240.00
Стихи	Пушкин А.С.	5	600.50	420.35
Слово милой	Пушкин А.С.	12	120.30	84.21
Поэмы	Пушкин А.С.	7	630.50	441.35
Скрюченный домишко	Агата Кристи	13	150.01	120.01

**Текст задания** (чтобы не прокручивать страницу) :

На распродаже размер скидки устанавливается в зависимости от количества экземпляров книги в магазине и от цены книги:

- для книг в остатке не менее 5 штук скидка 50%;
- для книг в остатке менее 5 штук скидка устанавливается в зависимости от цены: на книги не дешевле 700 руб скидка 20%, на остальные - 10%.

Вывести автора, название книги, количество, старая цена, скидка, цена со скидкой. Два последних столбца назвать **Скидка** и **Цена\_со\_скидкой**. Последний столбец округлить до двух знаков после запятой. Значение скидки вывести в виде строки текста (например, "15%"). Информацию отсортировать сначала по автору в алфавитном порядке, затем по убыванию цены книги со скидкой.

### Результат

author	title	amount	price	Скидка	Цена_со_скидкой
Агата Кристи	Пуаро ведет следствие	2	272.50	10%	245.25
Агата Кристи	Загадочное происшествие	15	260.99	50%	130.50
Агата Кристи	Смерть на Ниле	18	250.12	50%	125.06
Агата Кристи	Убийства по алфавиту	9	231.70	50%	115.85
Агата Кристи	Скрюченный домишко	13	150.01	50%	75.01
Булгаков М.А.	Мастер и Маргарита	3	670.99	10%	603.89
Булгаков М.А.	Белая гвардия	5	540.50	50%	270.25
Булгаков М.А.	Собачье сердце	6	380.90	50%	190.45
Дойл Артур Конан	Затерянный мир	3	400.00	10%	360.00
Дойл Артур Конан	Приключения Шерлока Холмса	11	590.50	50%	295.25
Дойл Артур Конан	Записки о Шерлоке Холмсе	14	470.20	50%	235.10
Дойл Артур Конан	Этюд в багровых тонах	3	215.00	10%	193.50
Достоевский Ф.М.	Братья Карамазовы	3	799.01	20%	639.21
Достоевский Ф.М.	Игрок	10	480.50	50%	240.25
Достоевский Ф.М.	Идиот	10	460.00	50%	230.00
Есенин С.А.	Стихотворения и поэмы	15	650.00	50%	325.00
Жюль Верн	Таинственный остров	0	169.99	10%	152.99
Жюль Верн	Вокруг света за 80 дней	15	201.01	50%	100.51
Лермонтов М.Ю.	Герой нашего времени	2	479.99	10%	431.99
Лермонтов М.Ю.	Стихи	6	550.60	50%	275.30
Лермонтов М.Ю.	Бородино	10	316.55	50%	158.28
Лермонтов М.Ю.	Поэмы	8	202.40	50%	101.20
Пушкин А.С.	Поэмы	7	630.50	50%	315.25
Пушкин А.С.	Стихи	5	600.50	50%	300.25
Пушкин А.С.	Евгений Онегин	8	200.11	50%	100.06
Пушкин А.С.	Капитанская дочка	5	199.99	50%	100.00
Пушкин А.С.	Слово милой	12	120.30	50%	60.15
Пушкин А.С.	Дубровский	7	114.99	50%	57.50

Affected rows: 28

### 1.1.5 Оператор CASE и функция IF(), пример

#### Задание

автор - Игорь Владимирович Лапшин

Акция "Купи книгу от 500 руб. и получи подарок".

Вывести автора, название книги и цену. Выбрать книги с ценой 500 рублей и выше, отсортировать информацию в алфавитном порядке сначала по автору, а затем по названию книги. Добавить столбец **Подарок**, в котором вывести, какой подарок получает покупатель: если куплена книга от 500 рублей до 600 рублей (включительно), то подарок - ручка, от 600.01 до 700 (включительно) - детская раскраска, выше 700 - гороскоп.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

Текст задания (чтобы не прокручивать страницу):

Акция "Купи книгу от 500 руб. и получи подарок".

Вывести автора, название книги и цену. Выбрать книги с ценой 500 рублей и выше, отсортировать информацию в алфавитном порядке сначала по автору, а затем названию книги. Добавить столбец **Подарок**, в котором вывести, какой подарок получает покупатель: если куплена книга от 500 рублей до 600 рублей (включительно), то подарок - ручка, от 600.01 до 700 (включительно) - детская раскраска, выше 700 - гороскоп.

#### Результат

author	title	price	Подарок
Булгаков М.А.	Белая гвардия	540.50	ручка
Булгаков М.А.	Мастер и Маргарита	670.99	детская раскраска
Дойл Артур Конан	Приключения Шерлока Холмса	590.50	ручка
Достоевский Ф.М.	Братья Карамазовы	799.01	гороскоп
Есенин С.А.	Стихотворения и поэмы	650.00	детская раскраска
Лермонтов М.Ю.	Стихи	550.60	ручка
Пушкин А.С.	Поэмы	630.50	детская раскраска
Пушкин А.С.	Стихи	600.50	детская раскраска

Affected rows: 8

### 1.1.6 Оператор CASE и функция IF(), пример

автор - Лариса Фернандес

К информации о книге добавить столбец "категория цены" :

- если цена меньше 250 - вывести "низкая";
- если цена в интервале от 250 до 600 включительно - вывести "средняя";
- более 600 - вывести "высокая".

Вывести автора, название, категорию, стоимость (цена \* количество). Последние два столбца назвать **price\_category** и **cost** . Из выборки исключить книги автора Агату Кристи, а также книги, имеющие названия "Поэмы" и "Стихи". Отсортировать сначала по убыванию стоимости, а затем по названию в алфавитном порядке.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

**Текст задания** (чтобы не прокручивать страницу) :

К информации о книге добавить столбец "категория цены" :

- если цена меньше 250 - вывести "низкая";
- если цена в интервале от 250 до 600 включительно - вывести "средняя";
- более 600 - вывести "высокая".

Вывести автора, название, категорию, стоимость (цена \* количество). Последние два столбца назвать **price\_category** и **cost** . Из выборки исключить книги автора Агату Кристи, а также книги, имеющие названия "Поэмы" и "Стихи". Отсортировать сначала по убыванию стоимости, а затем по названию в алфавитном порядке.

#### Результат

author	title	price_category	cost
Есенин С.А.	Стихотворения и поэмы	высокая	9750.00
Дойл Артур Конан	Записки о Шерлоке Холмсе	средняя	6582.80
Дойл Артур Конан	Приключения Шерлока Холмса	средняя	6495.50
Достоевский Ф.М.	Игрок	средняя	4805.00
Достоевский Ф.М.	Идиот	средняя	4600.00
Лермонтов М.Ю.	Бородино	средняя	3165.50
Жюль Верн	Вокруг света за 80 дней	низкая	3015.15
Булгаков М.А.	Белая гвардия	средняя	2702.50
Достоевский Ф.М.	Братья Карамазовы	высокая	2397.03
Булгаков М.А.	Собачье сердце	средняя	2285.40
Булгаков М.А.	Мастер и Маргарита	высокая	2012.97
Пушкин А.С.	Евгений Онегин	низкая	1600.88
Пушкин А.С.	Слово милой	низкая	1443.60
Дойл Артур Конан	Затерянный мир	средняя	1200.00
Пушкин А.С.	Капитанская дочка	низкая	999.95
Лермонтов М.Ю.	Герой нашего времени	средняя	959.98
Пушкин А.С.	Дубровский	низкая	804.93
Дойл Артур Конан	Этюд в багровых тонах	низкая	645.00
Жюль Верн	Таинственный остров	низкая	0.00

Affected rows: 19



### 1.1.7 Вычисление выражений, FLOOR()

#### Задание

автор - [Москвин Павел](#)

Вывести авторов, названия книг и их цену в двух столбцах – **Рубли** и **Копейки**, для тех книг, количество которых больше 11. Информацию отсортировать по сначала по убыванию рублей, затем убыванию копеек.

#### Пояснение

Получить ЧИСЛОВЫЕ значения рублей и копеек, используя функцию округления в меньшую сторону **FLOOR()**.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Текст задания (чтобы не прокручивать страницу) :

Вывести авторов, названия книг и их цену в двух столбцах – **Рубли** и **Копейки**, для тех книг, количество которых больше 11. Информацию отсортировать по сначала по убыванию рублей, затем убыванию копеек.

#### Результат

author	title	Рубли	Копейки
Есенин С.А.	Стихотворения и поэмы	650	0
Дойл Артур Конан	Записки о Шерлоке Холмсе	470	20
Агата Кристи	Загадочное происшествие	260	99
Агата Кристи	Смерть на Ниле	250	12
Жюль Верн	Вокруг света за 80 дней	201	1
Агата Кристи	Скрюченный домишко	150	1
Пушкин А.С.	Слово милой	120	30

Affected rows: 7

## 1.1.8 Функции работы с текстом, CHAR\_LENGTH()

### Задание

автор - Анатолий Алексеев

Вывести название книги и ее автора, упорядоченные сначала по возрастанию длины названия книги, затем по фамилии автора в алфавитном порядке. Также указать фактическую длину названия книги. Последний столбец назвать **Длина**.

### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

Для вычисления количества символов в строке используется функция

**CHAR\_LENGTH()**. Например:

```
CHAR_LENGTH("Запрос") = 6
```

**Текст задания** (чтобы не прокручивать страницу):

Вывести название книги и ее автора, упорядоченные сначала по возрастанию длины названия книги, затем по фамилии автора в алфавитном порядке. Также указать фактическую длину названия книги. Последний столбец назвать **Длина**.

### Результат

title	author	Длина
Идиот	Достоевский Ф.М.	5
Игрок	Достоевский Ф.М.	5
Стихи	Лермонтов М.Ю.	5
Поэмы	Лермонтов М.Ю.	5
Поэмы	Пушкин А.С.	5
Стихи	Пушкин А.С.	5
Бородино	Лермонтов М.Ю.	8
Дубровский	Пушкин А.С.	10
Слово милой	Пушкин А.С.	11
Белая гвардия	Булгаков М.А.	13
Смерть на Ниле	Агата Кристи	14
Собачье сердце	Булгаков М.А.	14
Затерянный мир	Дойл Артур Конан	14
Евгений Онегин	Пушкин А.С.	14
Братья Карамазовы	Достоевский Ф.М.	17
Капитанская дочка	Пушкин А.С.	17
Скрюченный домишко	Агата Кристи	18
Мастер и Маргарита	Булгаков М.А.	18
Таинственный остров	Жюль Верн	19
Убийства по алфавиту	Агата Кристи	20
Герой нашего времени	Лермонтов М.Ю.	20
Пуаро ведет следствие	Агата Кристи	21
Этюд в багровых тонах	Дойл Артур Конан	21
Стихотворения и поэмы	Есенин С.А.	21
Загадочное происшествие	Агата Кристи	23
Вокруг света за 80 дней	Жюль Верн	23
Записки о Шерлоке Холмсе	Дойл Артур Конан	24
Приключения Шерлока Холмса	Дойл Артур Конан	26

Affected rows: 28

### 1.1.9 Функции работы с текстом, вывод константы

#### Задание

Подготовить информацию для распродажи книг в конце года. Для этого вывести книги, которые есть в наличии на складе в количестве меньше 4. Для них уменьшить цену в два раза. Отобрать название книги, ее автора, количество и новую цену. Первым столбцом для каждой книги вывести слово **Распродажа**. Столбцы назвать **Категория**, **Книга**, **Автор**, **Количество**, **Цена**. Последний столбец округлить до двух знаков после запятой. Информацию отсортировать сначала по убыванию цены, затем по названию книги в алфавитном порядке.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Немного теории

В результат запроса можно включить столбец с одинаковым значением для всех строк таблицы. Для этого нужно указать это значение в **SELECT**.

#### Пример

Вывести информацию о всех книгах на складе, указав для каждой книги название этого склада ОПТ-КНИГА. Столбец назвать **Склад**.

Запрос:

```
SELECT
  'ОПТ_КНИГА' AS Склад, title, author, price, amount
FROM book;
```

#### Результат

Склад	title	author	price	amount
ОПТ_КНИГА	Мастер и Маргарита	Булгаков М.А.	670.99	3
ОПТ_КНИГА	Белая гвардия	Булгаков М.А.	540.50	5
ОПТ_КНИГА	Идиот	Достоевский Ф.М.	460.00	10
ОПТ_КНИГА	Братья Карамазовы	Достоевский Ф.М.	799.01	3
ОПТ_КНИГА	Игрок	Достоевский Ф.М.	480.50	10
ОПТ_КНИГА	Стихотворения и поэмы	Есенин С.А.	650.00	15
ОПТ_КНИГА	Таинственный остров	Жюль Верн	169.99	0
ОПТ_КНИГА	Пуаро ведет следствие	Агата Кристи	272.50	2
ОПТ_КНИГА	Евгений Онегин	Пушкин А.С.	200.11	8
ОПТ_КНИГА	Бородино	Лермонтов М.Ю.	316.55	10
ОПТ_КНИГА	Дубровский	Пушкин А.С.	114.99	7
ОПТ_КНИГА	Собачье сердце	Булгаков М.А.	380.90	6
ОПТ_КНИГА	Вокруг света за 80 дней	Жюль Верн	201.01	15
ОПТ_КНИГА	Смерть на Ниле	Агата Кристи	250.12	18
ОПТ_КНИГА	Убийства по алфавиту	Агата Кристи	231.70	9
ОПТ_КНИГА	Загадочное происшествие	Агата Кристи	260.99	15
ОПТ_КНИГА	Капитанская дочка	Пушкин А.С.	199.99	5
ОПТ_КНИГА	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
ОПТ_КНИГА	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
ОПТ_КНИГА	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
ОПТ_КНИГА	Затерянный мир	Дойл Артур Конан	400.00	3
ОПТ_КНИГА	Стихи	Лермонтов М.Ю.	550.60	6

ОПТ_КНИГА	Поэмы	Лермонтов М.Ю.	202.40	8
ОПТ_КНИГА	Герой нашего времени	Лермонтов М.Ю.	479.99	2
ОПТ_КНИГА	Стихи	Пушкин А.С.	600.50	5
ОПТ_КНИГА	Слово милой	Пушкин А.С.	120.30	12
ОПТ_КНИГА	Поэмы	Пушкин А.С.	630.50	7
ОПТ_КНИГА	Скрюченный домишко	Агата Кристи	150.01	13

**Текст задания** (чтобы не прокручивать страницу) :

Подготовить информацию для распродажи книг в конце года. Для этого вывести книги, которые есть в наличии на складе в количестве меньше 4. Для них уменьшить цену в два раза. Отобрать название книги, ее автора, количество и новую цену. Первым столбцом для каждой книги вывести слово **Распродажа**. Столбцы назвать **Категория**, **Книга**, **Автор**, **Количество**, **Цена**. Последний столбец округлить до двух знаков после запятой. Информацию отсортировать сначала по убыванию цены, затем по названию книги в алфавитном порядке.

### Результат

Категория	Книга	Автор	Количество	Цена
Распродажа	Братья Карамазовы	Достоевский Ф.М.	3	399.51
Распродажа	Мастер и Маргарита	Булгаков М.А.	3	335.50
Распродажа	Герой нашего времени	Лермонтов М.Ю.	2	240.00
Распродажа	Затерянный мир	Дойл Артур Конан	3	200.00
Распродажа	Пуаро ведет следствие	Агата Кристи	2	136.25
Распродажа	Этюд в багровых тонах	Дойл Артур Конан	3	107.50

Affected rows: 6

### 1.1.10 Функции работы с текстом CONCAT() и CONCAT\_WS()

#### Задание

автор - [Dmitry Kravtsov](#)

Магазин счёл, что классика уже не пользуется популярностью, поэтому необходимо в выборке:

1. Сменить всех авторов на "Донцова Дарья".
2. К названию каждой книги в начале дописать "Евлампия Романова и " ( *пробел в конце*).
3. Цену поднять на 42% (округлить её до двух знаков после запятой).
4. Отсортировать по убыванию цены.

Столбцы назвать `author`, `title`, `price`.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Немного теории

Для соединения значений нескольких столбцов или констант используются функции

- `CONCAT()`, в скобках, через запятую перечисляются названия столбцов или значения констант:

- `CONCAT_WS()`, в скобках первым параметром указывается разделитель, который будет разделять соединяемые значения, а затем через запятую перечисляются названия столбцов или значения констант.

#### Пример

Вывести информацию о книгах в одном столбце. Между значениями вставить запятую и пробел. Столбец назвать `Книга`.

*Запрос (оба запроса выдают одинаковый результат):*

```
SELECT
    CONCAT(title, ', ', author, ', ', price, ', ', amount) AS Книга
FROM book;
```

```
SELECT
    CONCAT_WS( ', ', title, author, price, amount) AS Книга
FROM book;
```

#### Результат

Книга
Мастер и Маргарита, Булгаков М.А., 670.99, 3
Белая гвардия, Булгаков М.А., 540.50, 5
Идиот, Достоевский Ф.М., 460.00, 10
Братья Карамазовы, Достоевский Ф.М., 799.01, 3
Игрок, Достоевский Ф.М., 480.50, 10
Стихотворения и поэмы, Есенин С.А., 650.00, 15
Таинственный остров, Жюль Верн, 169.99, 0
Пуаро ведет следствие, Агата Кристи, 272.50, 2

Евгений Онегин, Пушкин А.С., 200.11, 8	
Бородино, Лермонтов М.Ю., 316.55, 10	
Дубровский, Пушкин А.С., 114.99, 7	
Собачье сердце, Булгаков М.А., 380.90, 6	
Вокруг света за 80 дней, Жюль Верн, 201.01, 15	
Смерть на Ниле, Агата Кристи, 250.12, 18	
Убийства по алфавиту, Агата Кристи, 231.70, 9	
Загадочное происшествие, Агата Кристи, 260.99, 15	
Капитанская дочка, Пушкин А.С., 199.99, 5	
Этьюд в багровых тонах, Дойл Артур Конан, 215.00, 3	
Приключения Шерлока Холмса, Дойл Артур Конан, 590.50, 11	
Записки о Шерлоке Холмсе, Дойл Артур Конан, 470.20, 14	
Затерянный мир, Дойл Артур Конан, 400.00, 3	
Стихи, Лермонтов М.Ю., 550.60, 6	
Поэмы, Лермонтов М.Ю., 202.40, 8	
Герой нашего времени, Лермонтов М.Ю., 479.99, 2	
Стихи, Пушкин А.С., 600.50, 5	
Слово милой, Пушкин А.С., 120.30, 12	
Поэмы, Пушкин А.С., 630.50, 7	
Скрюченный домишко, Агата Кристи, 150.01, 13	
+-----+	+

**Текст задания** (чтобы не прокручивать страницу) :

Магазин счёл, что классика уже не пользуется популярностью, поэтому необходимо в выборке:

1. Сменить всех авторов на "Донцова Дарья".
2. К названию каждой книги в начале дописать "Евлампия Романова и " ( пробел в конце).
3. Цену поднять на 42% (округлить её до двух знаков после запятой).
4. Отсортировать по убыванию цены.

Столбцы назвать **author**, **title**, **price**.

## Результат

author	title	price
Донцова Дарья   Евлампия Романова и Братья Карамазовы		1134.59
Донцова Дарья   Евлампия Романова и Мастер и Маргарита		952.81
Донцова Дарья   Евлампия Романова и Стихотворения и поэмы		923.00
Донцова Дарья   Евлампия Романова и Поэмы		895.31
Донцова Дарья   Евлампия Романова и Стихи		852.71
Донцова Дарья   Евлампия Романова и Приключения Шерлока Холмса		838.51
Донцова Дарья   Евлампия Романова и Стихи		781.85
Донцова Дарья   Евлампия Романова и Белая гвардия		767.51
Донцова Дарья   Евлампия Романова и Игрок		682.31
Донцова Дарья   Евлампия Романова и Герой нашего времени		681.59
Донцова Дарья   Евлампия Романова и Записки о Шерлоке Холмсе		667.68
Донцова Дарья   Евлампия Романова и Идиот		653.20
Донцова Дарья   Евлампия Романова и Затерянный мир		568.00
Донцова Дарья   Евлампия Романова и Собачье сердце		540.88
Донцова Дарья   Евлампия Романова и Бородино		449.50
Донцова Дарья   Евлампия Романова и Пуаро ведет следствие		386.95
Донцова Дарья   Евлампия Романова и Загадочное происшествие		370.61
Донцова Дарья   Евлампия Романова и Смерть на Ниле		355.17
Донцова Дарья   Евлампия Романова и Убийства по алфавиту		329.01
Донцова Дарья   Евлампия Романова и Этьюд в багровых тонах		305.30
Донцова Дарья   Евлампия Романова и Поэмы		287.41
Донцова Дарья   Евлампия Романова и Вокруг света за 80 дней		285.43
Донцова Дарья   Евлампия Романова и Евгений Онегин		284.16
Донцова Дарья   Евлампия Романова и Капитанская дочка		283.99
Донцова Дарья   Евлампия Романова и Таинственный остров		241.39
Донцова Дарья   Евлампия Романова и Скрюченный домишко		213.01
Донцова Дарья   Евлампия Романова и Слово милой		170.83
Донцова Дарья   Евлампия Романова и Дубровский		163.29
+-----+	+	+

### 1.1.11 Функции работы с текстом LEFT() и RIGHT()

#### Задание

Магазин печатает этикетки небольшого размера. Длинные названия книг невозможно разместить на них. Поэтому было решено "обрезать" названия книг до 15 символов. Названия книг, длиной меньше или равных 15 оставить без изменения, а остальные обрезать до 12 символов и добавить многоточие ("...") в конце. Например:

- Приключения Шерлока Холмса преобразовать в Приключения ...
- Игрок - оставить без изменения

Вывести автора, сокращенное название книг и их цену. Столбцы назвать **Автор**, **Книга** и **Цена**. Информацию отсортировать сначала по названию книг в алфавитном порядке, затем по фамилии автора также в алфавитном порядке.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Немного теории

"Обрезать" текстовые значения можно с помощью двух функций:

LEFT(строка, длина) – обрезает строку с начала до длины, указанной во втором параметре

RIGHT(строка, длина) – обрезает строку с конца до длины, указанной во втором параметре

Например:

- LEFT("Язык структурированных запросов", 15) возвращает строку "Язык структурир"
- RIGHT("Язык структурированных запросов", 15) возвращает строку "ванных запросов"

#### Пример

Вывести названия книг, "обрезав" их до 20 знаков, в выборку включить только книги, названия которых больше 20.

Запрос:

```
SELECT
    left(title, 20) AS Книга
FROM book
WHERE CHAR_LENGTH(title) > 20;
```

#### Результат

Книга
Стихотворения и поэм
Пуаро ведет следствие

Вокруг света за 80 д
Загадочное происшеств
Этюд в багровых тона
Приключения Шерлока
Записки о Шерлоке Хо
+-----+

**Текст задания** (чтобы не прокручивать страницу) :

Магазин печатает этикетки небольшого размера. Длинные названия книг невозможно разместить на них. Поэтому было решено "обрезать" названия книг до 15 символов. Названия книг, длиной меньше или равных 15 оставить без изменения, а остальные обрезать до 12 символов и добавить многоточие ("...") в конце. Например:

- **Приключения Шерлока Холмса** преобразовать в **Приключения ...**
- **Игрок** - оставить без изменения

Вывести автора, сокращенное название книг и их цену. Столбцы назвать **Автор**, **Книга** и **Цена**. Информацию отсортировать сначала по названию книг в алфавитном порядке, затем по фамилии автора также в алфавитном порядке.

### Результат

Автор	Книга	Цена
Булгаков М.А.	Белая гвардия	540.50
Лермонтов М.Ю.	Бородино	316.55
Достоевский Ф.М.	Братья Карам...	799.01
Жюль Верн	Вокруг света...	201.01
Лермонтов М.Ю.	Герой нашего...	479.99
Пушкин А.С.	Дубровский	114.99
Пушкин А.С.	Евгений Онегин	200.11
Агата Кристи	Загадочное п...	260.99
Дойл Артур Конан	Записки о Ше...	470.20
Дойл Артур Конан	Затерянный мир	400.00
Достоевский Ф.М.	Игрок	480.50
Достоевский Ф.М.	Идиот	460.00
Пушкин А.С.	Капитанская ...	199.99
Булгаков М.А.	Мастер и Мар...	670.99
Лермонтов М.Ю.	Поэмы	202.40
Пушкин А.С.	Поэмы	630.50
Дойл Артур Конан	Приключения ...	590.50
Агата Кристи	Пуаро ведет ...	272.50
Агата Кристи	Скрюченный д...	150.01
Пушкин А.С.	Слово милой	120.30
Агата Кристи	Смерть на Ниле	250.12
Булгаков М.А.	Собачье сердце	380.90
Лермонтов М.Ю.	Стихи	550.60
Пушкин А.С.	Стихи	600.50
Есенин С.А.	Стихотворени...	650.00
Жюль Верн	Таинственный...	169.99
Агата Кристи	Убийства по ...	231.70
Дойл Артур Конан	Этюд в багро...	215.00

Affected rows: 28



### 1.1.12 Функции работы с текстом, SUBSTRING\_INDEX() и LOCATE()

Магазин печатает этикетки небольшого размера. Длинные названия книг невозможно разместить на них. Поэтому было решено "обрезать" названия книг до 15 символов. Названия книг, длиной меньше или равных 15 оставить без изменения, а остальные обрезать до 12 символов и добавить многоточие ("...") в конце. Также решили сократить фамилии и инициалы авторов по следующему правилу:

- если в таблице указаны инициалы автора, то удалить инициалы, например, **Пушкин А.С.** --> **Пушкин**;
- если же фамилия и имена авторов написаны без сокращений, то оставить последнее слово, например, **Дойл Артур Конан** --> **Конан**.

Вывести автора, сокращенное название книг и их цену. Столбцы назвать **Автор**, **Книга** и **Цена**. Информацию отсортировать сначала по названию книг в алфавитном порядке, затем по фамилии автора также в алфавитном порядке.

**Структура таблицы:**

book	
PK	book_id
	title
	author
	price
	amount

#### Немного теории

1. Функция:

**SUBSTRING\_INDEX(строка, разделитель, N)**

возвращает подстроку из **строки** перед появлением **N** вхождений разделителя.

Если **N** положителен, то возвращается все, что находится слева от последнего разделителя (считая слева). Если **N** отрицателен, то возвращается все, что находится справа от последнего разделителя (считая справа).

Например:

- `SUBSTRING_INDEX("Дойл Артур Конан", " ", 1)` - возвращает подстроку перед первым вхождением пробела, то есть Дойл;
- `SUBSTRING_INDEX("Дойл Артур Конан", " ", -1)` - возвращает подстроку перед первым вхождением пробела, но этот пробел ищется с конца, то есть результатом будет Конан.

#### Пример

Выделить первое слово из названия книг, количество которых больше 10.

*Запрос:*

```
SELECT
    SUBSTRING_INDEX(title, " ", 1) AS Книга
FROM book
WHERE amount > 10;
```

#### Результат

Книга
Стихотворения
Вокруг
Слово

Загадочное	
Приключения	
Записки	
Смерть	
Скрюченный	
+-----+	
Affected rows: 8	

2. Определить, встречается ли заданная подстрока в строке, а также получить ее позицию, можно с помощью функции

- `LOCATE(подстрока, строка)` - находит первое вхождение подстроки в строку;
- `LOCATE(подстрока, строка, N)` - находит первое вхождение подстроки в строку, начиная с позиции N.

Если подстроки в строке нет, то результат функции 0. Например:

```
LOCATE("SQL", "MySQL и SQL" ) = 3
```

```
LOCATE("SQL", "MySQL и SQL", 5) = 9
```

```
LOCATE("query", "MySQL и SQL") = 0
```

### Пример

Вывести книги и их авторов, название которых состоит из одного слова, то есть не содержат пробел.

*Запрос:*

```
SELECT title, author
FROM book
WHERE LOCATE(" ", title) = 0;
```

### Результат

+-----+	
title	author
+-----+	
Идиот	Достоевский Ф.М.
Игрок	Достоевский Ф.М.
Бородино	Лермонтов М.Ю.
Дубровский	Пушкин А.С.
Стихи	Лермонтов М.Ю.
Поэмы	Лермонтов М.Ю.
Стихи	Пушкин А.С.
Поэмы	Пушкин А.С.
+-----+	

**Текст задания** (чтобы не прокручивать страницу) :

Магазин печатает этикетки небольшого размера. Длинные названия книг невозможно разместить на них. Поэтому было решено "обрезать" названия книг до 15 символов. Названия книг, длиной меньше или равных 15 оставить без изменения, а остальные обрезать до 12 символов и добавить многоточие ("...") в конце. Также решили сократить фамилии и инициалы авторов по следующему правилу:

- если в таблице указаны инициалы автора, например, **Пушкин А.С.** --> **Пушкин**;
- если же фамилия и имена авторов написаны без сокращений, то оставить последнее слово, например, **Дойл Артур Конан** --> **Конан**.

Вывести автора, сокращенное название книг и их цену. Столбцы назвать **Автор**, **Книга** и **Цена**. Информацию отсортировать сначала по названию книг в алфавитном порядке, затем по фамилии автора также в алфавитном порядке.

## Результат

Автор	Книга	Цена
Булгаков	Белая гвардия	540.50
Лермонтов	Бородино	316.55
Достоевский	Братья Карам...	799.01
Верн	Вокруг света...	201.01
Лермонтов	Герой нашего...	479.99
Пушкин	Дубровский	114.99
Пушкин	Евгений Онегин	200.11
Кристи	Загадочное п...	260.99
Конан	Записки о Ше...	470.20
Конан	Затерянный мир	400.00
Достоевский	Игрок	480.50
Достоевский	Идиот	460.00
Пушкин	Капитанская ...	199.99
Булгаков	Мастер и Мар...	670.99
Лермонтов	Поэмы	202.40
Пушкин	Поэмы	630.50
Конан	Приключения ...	590.50
Кристи	Пуаро ведет ...	272.50
Кристи	Скрюченный д...	150.01
Пушкин	Слово милой	120.30
Кристи	Смерть на Ниле	250.12
Булгаков	Собачье сердце	380.90
Лермонтов	Стихи	550.60
Пушкин	Стихи	600.50
Есенин	Стихотворени...	650.00
Верн	Таинственный...	169.99
Кристи	Убийства по ...	231.70
Конан	Этюд в багро...	215.00

Affected rows: 28

## 1.2 Простые запросы на выборку, база данных «Резюме»

В этом уроке предлагается написать SQL запросы для таблицы **resume**, в которой хранится информация о резюме для трудоустройства. Каждое резюме содержит имя соискателя, профессиональную область, в которой он ищет работу, должность, минимальную зарплату, стаж работы в этой должности, а также дата рождения соискателя. Дату рождения, как правило, в резюме не указывают, но здесь она включена, чтобы повторить и изучить новые функции работы с датами.

### 1.2.1 Структура и наполнение таблицы

Все запросы будут формулироваться для таблицы **resume**, которая создана следующим запросом:

```
CREATE TABLE resume (  
    resume_id INT PRIMARY KEY AUTO INCREMENT,  
    applicant VARCHAR(60), # фамилия и инициалы соискателя  
    specialisation VARCHAR(80), # область профессиональной  
    деятельности  
    position VARCHAR(50), # должность  
    min_salary INT, # минимальная заработная плата  
    experience INT, # опыт работы  
    date_birth DATE # дата рождения  
);
```

Наполнение таблицы **resume**:

resume_id	applicant	specialisation	position	min_salary	experience	date_birth
1	Иванов М.С.	IT	web-программист	50000	2	1998-12-15
2	Петров Ф.С.	Строительство	инженер-строитель	45000	5	1995-1-7
3	Федоров П.Р.	Продажи	менеджер по продажам	35000	7	1993-8-26
4	Абрамова А.А.	Продажи	региональный менеджер	90000	4	1992-9-23
5	Самарин С.С.	Транспорт	логист-диспетчер	30000	0	2001-7-18
6	Туполев И.Д.	Строительство	инженер-строитель	40000	10	1985-3-16
7	Баранов П.В.	Юристы	адвокат	60000	5	1982-2-24
8	Шевцова А.Р.	Продажи	менеджер по продажам	30000	2	1995-11-12
9	Белых К.Д.	IT	Python программист	100000	5	1991-12-9
10	Садиев С.И.	IT	программист PHP и Питон	90000	6	1982-11-7
11	Астахов И.И.	Строительство	прораб	100000	15	1976-1-30
12	Борисов В.В.	Продажи	менеджер по продажам	50000	12	1980-3-1

resume_id	applicant	specialisation	position	min_salary	experience	date_birth
13	Тощакон П.С.	Строительство	технолог	50000	15	1979-4-25
14	Солиева К.П.	Продажи	директор по продажам	130000	7	1973-4-30
15	Левченко А.Г.	IT	PHP разработчик	110000	9	1967-12-27
16	Хованец Г.П.	Строительство	прораб	90000	16	1970-10-1
17	Тихомиров Т.М.	Транспорт	экспедитор	45000	20	1975-8-8
18	Дворников М.А.	Транспорт	логист-диспетчер	35000	2	1998-11-10
19	Жидкова Р.Л.	Продажи	менеджер по продажам	30000	5	1987-10-20
20	Бондарев С.С.	IT	web-программист	130000	9	1987-02-16
21	Иванов П.А.	Продажи	директор по продажам	140000	11	1979-04-24
22	Прохоров А.Д.	IT	web-программист	200000	8	1985-03-23
23	Соболев А.И.	Транспорт	водитель	35000	15	1984-03-30
24	Ромашко Ф.Ю.	Продажи	продавец-консультант	25000	0	2002-05-07
25	Краско С.С.	IT	программист 1С	85000	3	1997-09-02
26	Паршина Г.И.	IT	программист С и Питон	110000	9	1981-08-11
27	Светлова С.Т.	Транспорт	водитель	40000	10	1991-09-17
28	Буюнов П.Д.	Юристы	адвокат	55000	15	1980-06-01
29	Ульянов М.А.	Продажи	региональный менеджер	130000	11	1979-06-19
30	Тупикин К.Л.	IT	программист 1.7С	140000	12	1981-07-03
31	Ермоленко Г.С.	IT	программист С#	90000	6	1993-12-09
32	Сомов Я.Г.	IT	программист С++	115000	8	1990-11-16
33	Орлова Е.Р.	Строительство	head	170000	15	1972-01-16
34	Ларин В.П.	Строительство	инженер-строитель	70000	12	1970-05-15

Дамп базы данных этого урока:

```
CREATE TABLE resume (
  resume_id INT PRIMARY KEY AUTO_INCREMENT,
  applicant VARCHAR(60),
  specialisation VARCHAR(80),
  position VARCHAR(50),
  min_salary INT,
  experience INT,
  date_birth DATE);
```

```

INSERT INTO resume(applicant, specialisation, position,min_salary, experience,
date_birth) VALUES
('Иванов М.С.', 'IT', 'web-программист', 50000, 2, '1998-12-15'),
('Петров Ф.С.', 'Строительство', 'инженер-строитель', 45000, 5, '1995-01-07'),
('Федоров П.Р.', 'Продажи', 'менеджер по продажам', 35000, 7, '1993-08-26'),
('Абрамова А.А.', 'Продажи', 'региональный менеджер', 90000, 4, '1992-09-23'),
('Самарин С.С.', 'Транспорт', 'логист-диспетчер', 30000, 0, '2001-07-18'),
('Туполев И.Д.', 'Строительство', 'инженер-строитель', 40000, 10, '1985-03-16'),
('Баранов П.В.', 'Юристы', 'адвокат', 60000, 5, '1982-02-24'),
('Шевцова А.Р.', 'Продажи', 'менеджер по продажам', 30000, 2, '1995-11-12'),
('Белых К.Д.', 'IT', 'Python программист', 100000, 5, '1991-12-09'),
('Садиев С.И.', 'IT', 'программист PHP и Питон', 90000, 6, '1982-11-07'),
('Астахов И.И.', 'Строительство', 'прораб', 100000, 15, '1976-01-30'),
('Борисов В.В.', 'Продажи', 'менеджер по продажам', 50000, 12, '1980-03-01'),
('Тошаков П.С.', 'Строительство', 'технолог', 50000, 15, '1979-04-25'),
('Солиева К.П.', 'Продажи', 'директор по продажам', 130000, 7, '1973-04-30'),
('Левченко А.Г.', 'IT', 'PHP разработчик', 110000, 9, '1967-12-27'),
('Хованец Г.П.', 'Строительство', 'прораб', 90000, 16, '1970-10-01'),
('Тихомиров Т.М.', 'Транспорт', 'экспедитор', 45000, 20, '1975-08-08'),
('Дворников М.А.', 'Транспорт', 'логист-диспетчер', 35000, 2, '1998-11-10'),
('Жидкова Р.Л.', 'Продажи', 'менеджер по продажам', 30000, 5, '1987-10-20'),
('Бондарев С.С.', 'IT', 'web-программист', 130000, 9, '1987-02-16'),
('Иванов П.А.', 'Продажи', 'директор по продажам', 140000, 11, '1979-04-24'),
('Прохоров А.Д.', 'IT', 'web-программист', 200000, 8, '1985-03-23'),
('Соболев А.И.', 'Транспорт', 'водитель', 35000, 15, '1984-03-30'),
('Ромашко Ф.Ю.', 'Продажи', 'продавец-консультант', 25000, 0, '2002-05-07'),
('Краско С.С.', 'IT', 'программист 1С', 85000, 3, '1997-09-02'),
('Паршина Г.И.', 'IT', 'программист С и Питон', 110000, 9, '1981-08-11'),
('Светлова С.Т.', 'Транспорт', 'водитель', 40000, 10, '1991-09-17'),
('Буянов П.Д.', 'Юристы', 'адвокат', 55000, 15, '1980-06-01'),
('Ульянов М.А.', 'Продажи', 'региональный менеджер', 130000, 11, '1979-06-19'),
('Тупикин К.Л.', 'IT', 'программист 1.7C', 140000, 12, '1981-07-03'),
('Ермоленко Г.С.', 'IT', 'программист C#', 90000, 6, '1993-12-09'),
('Сомов Я.Г.', 'IT', 'программист C++', 115000, 8, '1990-11-16'),
('Орлова Е.Р.', 'Строительство', 'head', 170000, 15, '1972-01-16'),
('Ларин В.П.', 'Строительство', 'инженер-строитель', 70000, 12, '1970-05-15');

```

## 1.2.2 Регулярные выражения, наличие текста в строке

### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой содержит слово "менеджер". Для каждого соискателя указать их должность. Информацию отсортировать по фамилиям соискателей в алфавитном порядке.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary,
        experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP**:

```
строка [NOT] REGEXP регулярное_выражение
```

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результатом сравнения будет истина, если этот текст встречается в любом месте строки.

*Например:*

```
"я изучаю SQL" REGEXP "SQL" - истина
```

```
"я изучаю sql" REGEXP "SQL" - истина
```

```
"я изучаю Structure Query Language" REGEXP "SQL" - ложь
```

На данной платформе по умолчанию установлено, что строка шаблона - регистронезависимая, то есть SQL можно записать как большими, так и маленькими буквами.

Оператор **REGEXP** по выполняемой функции похож на оператор **LIKE**, но позволяет реализовать более сложный поиск по строке. Так, например, следующие выражения выдают совершенно одинаковый результат:

```
"я изучаю SQL" REGEXP "SQL" - истина
```

```
"я изучаю SQL" LIKE "%SQL%" - истина
```

### Пример

Вывести все должности, которые содержат слово "программист".

*Запрос:*

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP 'программист';
```

Вариант с **LIKE**:

```
SELECT DISTINCT position
FROM resume
WHERE position LIKE '%программист%';
```

### Результат

```
+-----+
| position |
+-----+
| web-программист |
| Python программист |
| программист 1C |
+-----+
Affected rows: 3
```

**Текст задания (чтобы не прокручивать страницу) :**

Вывести фамилию и инициалы тех соискателей, претендующих на должность, название которой содержит слово "менеджер". Для каждого соискателя указать их должность. Информацию отсортировать по фамилиям соискателей в алфавитном порядке.

### Результат

```
+-----+-----+
| applicant | position |
+-----+-----+
| Абрамова А.А. | региональный менеджер |
| Борисов В.В. | менеджер по продажам |
| Жидкова Р.Л. | менеджер по продажам |
| Ульянов М.А. | региональный менеджер |
| Федоров П.Р. | менеджер по продажам |
| Шевцова А.Р. | менеджер по продажам |
+-----+-----+
Affected rows: 6
```



### 1.2.3 Регулярные выражения, начало и конец строки

#### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой заканчивается на русскую букву "р". Для каждого соискателя указать их должность и дату рождения. Информацию отсортировать сначала по должности в алфавитном порядке, затем по возрастанию даты рождения.

#### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position,min_salary,
```

#### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца.

Синтаксис **REGEXP** :

**строка [NOT] REGEXP регулярное\_выражение**

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результатом сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки.

#### Например:

```
"я изучаю SQL" REGEXP "SQL$" - истина  
"я изучаю SQL" REGEXP "^SQL" - ложь
```

#### Пример

Вывести все должности, первые буквы которого - "программист".

Запрос:

```
SELECT DISTINCT position  
FROM resume  
WHERE position REGEXP '^программист';
```

#### Пояснение

Шаблон состоит из двух частей:

- "программист" - это строка должна встречаться в названии (отберутся названия с *программист*, *программисты*, *web-программист*);
- "^" - указывает, что строка "программист" должна располагаться в начале названия (отберутся названия с *программист*, *программисты*, но не *web-программист*, так как это название начинается с "web", а не "программист").

Вариант с **LIKE**:

```
SELECT DISTINCT position
FROM resume
WHERE position LIKE 'программист%';
```

### Результат

position
программист PHP и Питон
программист 1C
программист C и Питон
программист 1.7C
программист C#
программист C++

Affected rows: 6

**Текст задания** (чтобы не прокручивать страницу) :

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой заканчивается на русскую букву "р". Для каждого соискателя указать их должность и дату рождения. Информацию отсортировать сначала по должности в алфавитном порядке, затем по возрастанию даты рождения.

### Результат

applicant	position	date_birth
Дворников М.А.	логист-диспетчер	1998-11-10
Самарин С.С.	логист-диспетчер	2001-07-18
Ульянов М.А.	региональный менеджер	1979-06-19
Абрамова А.А.	региональный менеджер	1992-09-23
Тихомиров Т.М.	экспедитор	1975-08-08

## 1.2.4 Регулярные выражения, наличие заданных символов в строке

### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой начинается либо с "а", либо "п", либо "и". Для каждого соискателя указать их должность и минимальную зарплату. Информацию отсортировать по фамилиям соискателей в алфавитном порядке.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary, experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP**:

**строка [NOT] REGEXP регулярное\_выражение**

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результатом сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания.

### Например:

"я изучаю SQL" REGEXP "[SP]" - истина, в строке содержится символ S

"я изучаю Python" REGEXP "[SP]" - истина, в строке содержится символ

P

"я изучаю C++" REGEXP "[SP]" - ложь, в строке не содержится ни S, ни P

### Пример

Вывести все должности, которые содержат либо пробел, либо "-".

Запрос:

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP '[ -]';
```

### Пояснение

Шаблон указывает на то, что в названии обязательно должен быть один из символов: либо пробел, либо "-", - так как эти символы заключены в квадратные скобки.

Вариант с **LIKE**:

```
SELECT DISTINCT position
FROM resume
WHERE position LIKE '% %' OR position LIKE '%-%';
```

### Результат

position
web-программист
инженер-строитель
менеджер по продажам
региональный менеджер
логист-диспетчер
Python программист
программист PHP и Питон
директор по продажам
PHP разработчик
продавец-консультант
программист 1C
программист C и Питон
программист 1.7C
программист C#
программист C++

Affected rows: 15

### Текст задания (чтобы не прокручивать страницу) :

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой начинается либо с "а", либо "п", либо "и". Для каждого соискателя указать их должность и минимальную зарплату. Информацию отсортировать по фамилиям соискателей в алфавитном порядке.

### Результат

applicant	position	min_salary
Астахов И.И.	прораб	100000
Баранов П.В.	адвокат	60000
Буянов П.Д.	адвокат	55000
Ермоленко Г.С.	программист C#	90000
Краско С.С.	программист 1C	85000
Ларин В.П.	инженер-строитель	70000
Паршина Г.И.	программист C и Питон	110000
Петров Ф.С.	инженер-строитель	45000
Ромашко Ф.Ю.	продавец-консультант	25000
Садиев С.И.	программист PHP и Питон	90000
Сомов Я.Г.	программист C++	115000
Тупикин К.Л.	программист 1.7C	140000
Туполев И.Д.	инженер-строитель	40000
Хованец Г.П.	прораб	90000

Affected rows: 14

## 1.2.5 Регулярные выражения, наличие интервала символов в строке

### Задание

Вывести фамилию и инициалы тех соискателей, имя (первая буква инициалов) которых начинается с букв от "А" до "К" включительно. Указать должность, на которую они претендуют, и минимальную заработную плату. Информацию отсортировать сначала по инициалам в алфавитном порядке, а затем по убыванию минимальной заработной платы.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary,
        experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP** :

**строка [NOT] REGEXP регулярное\_выражение**

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результатом сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания;
- **[начальный\_символ-конечный\_символ]** - соответствует любому одиночному символу из диапазона символов.

### Например:

```
"я изучаю 1С" REGEXP "[1-6]" - истина, в строке содержится символ 1,
                               который принадлежит интервалу от 1 до 6 включительно
"я изучаю 2С" REGEXP "[1-6]" - истина, в строке содержится символ 2,
                               который принадлежит интервалу от 1 до 6 включительно
"я изучаю 7С" REGEXP "[1-6]" - ложь, в строке содержится символ 7,
                               а в соответствии с шаблоном там должно быть число, принадлежащее
                               интервалу от 1 до 6 включительно
```

### Пример

Вывести все должности, которые начинаются с русских букв от "а" до "м" включительно.

Запрос:

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP '^[a-м]';
```

### Пояснение

Шаблон состоит из двух частей:

- "[a-м]" - указывает на то, что название должно содержать русскую буквы от "а" до "м" включительно;
- "^" - указывает, что эта буква должна располагаться в начале названия .

Вариант с **LIKE**:

```
SELECT DISTINCT position
FROM resume
WHERE position LIKE 'а%' OR position LIKE 'б%' OR LIKE 'в%'
OR LIKE 'г%' OR LIKE 'д%' OR LIKE 'е%' OR LIKE 'ж%'
OR LIKE 'з%' OR LIKE 'и%' OR LIKE 'й%' OR LIKE 'к%'
OR LIKE 'л%' OR LIKE 'м%';
```

### Результат

```
+-----+
| position |
+-----+
| инженер-строитель |
| менеджер по продажам |
| логист-диспетчер |
| адвокат |
| директор по продажам |
| водитель |
+-----+
Affected rows: 6
```

### Текст задания (чтобы не прокручивать страницу) :

Вывести фамилию и инициалы тех соискателей, имя (первая буква инициалов) которых начинается с букв от "А" до "К" включительно. Указать должность, на которую они претендуют, и минимальную заработную плату. Информацию отсортировать сначала по инициалам в алфавитном порядке, а затем по убыванию минимальной заработной платы.

### Пояснение

Сортировку можно осуществлять не только по столбцу целиком, но и по его части. Для выделения части текстового значения столбца используйте функцию **SUBSTRING\_INDEX()**.

### Результат

applicant	position	min_salary
Абрамова А.А.	региональный менеджер	90000
Левченко А.Г.	PHP разработчик	110000
Прохоров А.Д.	web-программист	200000
Соболев А.И.	водитель	35000
Шевцова А.Р.	менеджер по продажам	30000

	Борисов В.В.		менеджер по продажам		50000	
	Ларин В.П.		инженер-строитель		70000	
	Паршина Г.И.		программист С и Питон		110000	
	Хованец Г.П.		прораб		90000	
	Ермоленко Г.С.		программист С#		90000	
	Орлова Е.Р.		head		170000	
	Туполев И.Д.		инженер-строитель		40000	
	Астахов И.И.		прораб		100000	
	Белых К.Д.		Python программист		100000	
	Тупикин К.Л.		программист 1.7C		140000	
	Солиева К.П.		директор по продажам		130000	
+-----+						
Affected rows: 16						

## 1.2.6 Регулярные выражения, отсутствие символов и интервала символов

### Задание

Вывести фамилию и инициалы тех соискателей, имя (первая буква инициалов) которых не содержит букв "А" и "М", а отчество (вторая буква инициалов) не содержит букв из интервала от "К" до "Ф". Указать должность, на которую они претендуют, и минимальную заработную плату. Информацию отсортировать сначала по второй букве инициалов в обратном алфавитном порядке, а затем по возрастанию минимальной заработной платы.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary,
        experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP** :

строка [NOT] REGEXP регулярное\_выражение

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результатом сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания;
- **[начальный\_символ-конечный\_символ]** - соответствует любому одиночному символу из диапазона символов;
- **[^символы]** строка содержит НЕ символы, перечисленные после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^;
- **[^начальный\_символ-конечный\_символ]** - строка содержит НЕ символы, из интервала после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^.

### Дополнение к теории от [Белый Евгений](#)

Регулярное выражения - это тот шаблон, который необходимо найти в тексте, и ответ ИСТИНА или ЛОЖЬ будет зависеть от того, найдено ли полное совпадение.



### Например

`"я изучаю 1С" REGEXP "[^12]"` - выдаст часть строки за исключением цифр 1 или 2, т.е. результатом будет "я изучаю С" (результат ИСТИНА, т.к. в данном случае истина любое значение, отличное от 0);

`"я изучаю 1" REGEXP "[^12]$"` - выдаст часть строки за исключением цифр 1 или 2, расположенных в конце строки (результат ЛОЖЬ, так как будет выделена пустая строка).

### Пример

Вывести все должности, которые не начинаются с русских букв "а" до "м" включительно.

Запрос:

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP '^[^а-м]';
```

### Пояснение

Шаблон состоит из двух частей:

- `"[а-м]"` - указывает на то, что название должно содержать русскую буквы от "а" до "м" включительно;
- `"^"` - указывает, что эта буква должна располагаться в начале названия .

Вариант с **LIKE**:

```
SELECT DISTINCT position
FROM resume
WHERE position NOT LIKE 'а%' OR position NOT LIKE 'б%' OR NOT LIKE 'в%'
OR NOT LIKE 'г%' OR NOT LIKE 'д%' OR NOT LIKE 'е%' OR NOT LIKE 'ж%'
OR NOT LIKE 'з%' OR NOT LIKE 'и%' OR NOT LIKE 'й%' OR NOT LIKE 'к%'
OR NOT LIKE 'л%' OR NOT LIKE 'м%';
```

### Результат

position
web-программист
региональный менеджер
Python программист
программист PHP и Питон
прораб
технолог
PHP разработчик
экспедитор
продавец-консультант
программист 1С
программист С и Питон
программист 1.7С
программист С#
программист С++
head

Affected rows: 15

### Текст задания (чтобы не прокручивать страницу) :

Вывести фамилию и инициалы тех соискателей, имя (первая буква инициалов) которых не содержит букв "А" и "М", а отчество ( вторая буква инициалов) не содержит букв из интервала от "К" до "Ф". Указать должность, на которую они претендуют, и минимальную

заработную плату. Информацию отсортировать сначала по второй букве инициалов в обратном алфавитном порядке, а затем по возрастанию минимальной заработной платы.

### Пояснение

Сортировку можно осуществлять не только по столбцу целиком, но и по его части. Для выделения части текстового значения столбца используйте функцию **SUBSTRING\_INDEX()**.

### Результат

applicant	position	min_salary
Ромашко Ф.Ю.	продавец-консультант	25000
Садиев С.И.	программист PHP и Питон	90000
Астахов И.И.	прораб	100000
Паршина Г.И.	программист С и Питон	110000
Туполев И.Д.	инженер-строитель	40000
Буянов П.Д.	адвокат	55000
Белых К.Д.	Python программист	100000
Сомов Я.Г.	программист C++	115000
Борисов В.В.	менеджер по продажам	50000
Баранов П.В.	адвокат	60000
Иванов П.А.	директор по продажам	140000

Affected rows: 11

## 1.2.7 Регулярные выражения, соответствие одному из шаблонов

### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой либо начинается с английской буквы, либо заканчивается символами "менеджер". Для каждого соискателя указать их должность и дату рождения. Информацию отсортировать по убыванию даты рождения.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary,
        experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP**:

```
строка [NOT] REGEXP регулярное_выражение
```

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результат сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания;
- **[начальный\_символ-конечный\_символ]** - соответствует любому одиночному символу из диапазона символов;
- **[^символы]** строка содержит НЕ символы, перечисленные после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^;
- **[^начальный\_символ-конечный\_символ]** - строка содержит НЕ символы, из интервала после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^.
- **|** - отделяет два шаблона строки, и значение должно соответствовать одному из этих шаблонов.

*Например:*

"я изучаю SQL" REGEXP "SQL|Python" - истина, так как в строке есть SQL

"я изучаю Python" REGEXP "SQL|Python" - истина, так как в строке есть Python

"я изучаю PHP" REGEXP "SQL|Python" - ложь, так как в строке нет ни SQL, ни Python

### Пример

Вывести все должности, первые символы которого либо "программист", либо "менеджер."

*Запрос:*

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP '^программист|^менеджер';
```

### Пояснение

Шаблон включает:

- "|" - указывает на то, что шаблон состоит из двух частей, название должно соответствовать либо первой части шаблона, либо второй;
- первая часть шаблона "^программист" - название должно начинаться с "программист";
- вторая часть шаблона "^менеджер" - название должно начинаться с "менеджер".

Вариант с **LIKE**:

```
SELECT DISTINCT position
FROM resume
WHERE position LIKE 'программист%' OR position LIKE 'менеджер%';
```

### Результат

position
менеджер по продажам
программист PHP и Питон
программист 1C
программист C и Питон
программист 1.7C
программист C#
программист C++

Affected rows: 7

### Текст задания (чтобы не прокручивать страницу):

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой либо начинается с английской буквы, либо заканчивается символами "менеджер". Для каждого соискателя указать их должность и дату рождения. Информацию отсортировать по убыванию даты рождения.

### Результат

applicant	position	date birth
Иванов М.С.	web-программист	1998-12-15
Абрамова А.А.	региональный менеджер	1992-09-23
Белых К.Д.	Python программист	1991-12-09
Бондарев С.С.	web-программист	1987-02-16
Прохоров А.Д.	web-программист	1985-03-23
Ульянов М.А.	региональный менеджер	1979-06-19

Орлова Е.Р.	head	1972-01-16	
Левченко А.Г.	РНР разработчик	1967-12-27	
+-----+-----+-----+			

## 1.2.8 Регулярные выражения, соответствие одному любому символу

### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой состоит из 6 или 7 символов. Для каждого соискателя указать их должность и минимальную зарплату. Информацию отсортировать по убыванию минимальной заработной платы.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary,
        experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP** :

**строка [NOT] REGEXP регулярное\_выражение**

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результат сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания;
- **[начальный\_символ-конечный\_символ]** - соответствует любому одиночному символу из диапазона символов;
- **[^символы]** строка содержит НЕ символы, перечисленные после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^;
- **[^начальный\_символ-конечный\_символ]** - строка содержит НЕ символы, из интервала после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^.
- **|** - отделяет два шаблона строки, и значение должно соответствовать одному из этих шаблонов;
- **.** - соответствует любому непустому символу, если же в шаблоне символ точка "." как элемент текстовой строки, то этот символ необходимо "экранировать", то есть вставить перед символом строки \.

*Например:*

"я изучаю SQL" REGEXP " ...\$" - истина, так как строка заканчивается на слово SQL, которое состоит из 3 символов

"я изучаю Python" REGEXP " ...\$" - ложь, в конце строки нет слова из 3-х символов

"PHP я изучаю" REGEXP " ...\$" - ложь, так как в конце строки нет слова из 3-х символов

### Пример

Вывести все профессиональные области, длина названия которых больше 2 символов.

Запрос:

```
SELECT DISTINCT specialisation
FROM resume
WHERE specialisation NOT REGEXP '^.$|^..$';
```

### Пояснение

Шаблон включает :

- "|" - указывает на то, что шаблон состоит из двух частей, название должно соответствовать либо первой части шаблона, либо второй;
- первая часть шаблона "^.\$":
  - "." - название должно состоять из одного любого символа,
  - "^" - с этого символа должно начинаться название,
  - "\$" - этим символом название должно заканчиваться,
 то есть шаблон "^.\$" означает, что название должно состоять из одного символа;
- вторая часть шаблона "^..\$" описывает вариант, что название может состоять из двух символов.

**NOT REGEXP** используется для того, чтобы отобрать все названия, кроме тех, которые соответствуют шаблону.

Вариант с **LIKE**:

```
SELECT DISTINCT specialisation
FROM resume
WHERE specialisation NOT LIKE '_' AND specialisation NOT LIKE '__';
```

### Результат

```
+-----+
| specialisation |
+-----+
| Строительство |
| Продажи       |
| Транспорт     |
| Юристы        |
+-----+
Affected rows: 4
```

### Текст задания (чтобы не прокручивать страницу):

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой состоит из 6 или 7 символов. Для каждого соискателя указать их должность и минимальную зарплату. Информацию отсортировать по убыванию минимальной заработной платы.

### Результат

```
+-----+-----+-----+
| applicant | position | min_salary |
+-----+-----+-----+
| Астахов И.И. | прораб   | 100000     |
| Хованец Г.П. | прораб   | 90000      |
| Баранов П.В. | адвокат | 60000      |
| Буянов П.Д.  | адвокат | 55000      |
+-----+-----+-----+
```

## 1.2.9 Регулярные выражения, повторение символов 0 и более раз

### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой состоит из двух слов, начинается со слова "программист", а второе слово включает буквы "С" (русскую или английскую) перед которой могут (но не обязательно) стоять только цифры, а после только символы "+" и "#" в любом количестве, в том числе и эти символы могут отсутствовать. Вывести должность, на которую претендует соискатель. Информацию отсортировать по фамилии соискателя в алфавитном порядке.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary,
        experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP**:

**строка [NOT] REGEXP регулярное\_выражение**

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результат сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания;
- **[начальный\_символ-конечный\_символ]** - соответствует любому одиночному символу из диапазона символов;
- **[^символы]** строка содержит НЕ символы, перечисленные после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^;
- **[^начальный\_символ-конечный\_символ]** - строка содержит НЕ символы, из интервала после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^.
- **|** - отделяет два шаблона строки, и значение должно соответствовать одному из этих шаблонов;



- `.` - соответствует любому непустому символу, если же в шаблоне символ точка "." как элемент текстовой строки, то этот символ необходимо "экранировать", то есть вставить перед символом строки `\`;
- **СИМВОЛ\*** - определяет любое количество повторений указанного перед звездочкой символа ( в том числе символа вообще может и не быть).

### Например:

"я изучаю SQL" REGEXP "[a-zA-Z]\*\$" - истина, так как в строке есть английские буквы  
 "я изучаю" REGEXP "[a-zA-Z]\*\$" - истина, так как при таком шаблоне английских букв в строке может и не быть

### Пример

Вывести все должности, первое слово названия которых начинается с английской буквы "p", за которой следуют английские буквы (слово может состоять из одной буквы "p"). Слова в названии разделяются пробелами.

**Пояснение.** Если должность состоит из нескольких слов, то признаком конца первого слова является пробел, если же должность - одно слово, то признак его конца - конец строки.

### Запрос:

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP "^p[a-zA-Z]* |^p[a-zA-Z]*$";
```

### Пояснение

Шаблон включает :

- `"|"` - указывает на то, что шаблон состоит из двух частей, название должно соответствовать либо первой части шаблона, либо второй;
  - первая часть шаблона `"^p[a-zA-Z]* "`:  
`"^p"` - название должно начинаться с "p",  
`"[a-zA-Z]*"` - далее может идти любое количество английских букв, в том числе и нулевое,  
`" "` (пробел) - пробелом должны заканчиваться английские буквы, то есть шаблон `"^p[a-zA-Z]* "` означает, что название должно начинаться с английского слова, первая буква которого "p", признаком конца слова является пробел;
  - вторая часть шаблона `"^p[a-zA-Z]*$"` учитывает вариант, что название состоит из одного английского слова ("`^`" символ начала строки, затем "p", затем любое количество повторений английских букв `"[a-zA-Z]*"` и признак конца строки `"$"`).

### Результат

```
+-----+
| position |
+-----+
| Python программист |
| PHP разработчик |
+-----+
Affected rows: 2
```

### Текст задания (чтобы не прокручивать страницу):

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой состоит из двух слов, начинается со слова "программист", а второе слово включает буквы "С" (русскую или английскую) перед которой могут (но не обязательно) стоять только

цифры, а после только символы "+" и "#" в любом количестве, в том числе и эти символы могут отсутствовать. Вывести должность, на которую претендует соискатель. Информацию отсортировать по фамилии соискателя в алфавитном порядке.

### Результат

applicant	position
Ермоленко Г.С.	программист С#
Краско С.С.	программист 1С
Сомов Я.Г.	программист С++

Affected rows: 3

## 1.2.10 Регулярные выражения, повторение символов один и более раз

### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой заканчивается на слово, состоящее из английских букв, цифр или символов "+", "#", ".", расположенных в любом порядке. Слова в названии разделяются пробелом. Вывести должность, на которую претендует соискатель. Информацию отсортировать по фамилии соискателя в алфавитном порядке.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position,min_salary,
        experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP** :

**строка** [NOT] **REGEXP** регулярное\_выражение

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результат сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания;
- **[начальный\_символ-конечный\_символ]** - соответствует любому одиночному символу из диапазона символов;
- **[^символы]** строка содержит НЕ символы, перечисленные после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^;
- **[^начальный\_символ-конечный\_символ]** - строка содержит НЕ символы, из интервала после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^.
- **|** - отделяет два шаблона строки, и значение должно соответствовать одному из этих шаблонов;
- **.** - соответствует любому непустому символу, если же в шаблоне символ точка "." как элемент текстовой строки, то этот символ необходимо "экранировать", то есть вставить перед символом строки \;

- **СИМВОЛ\*** - определяет любое количество повторений указанного перед звездочкой символа ( в том числе символа вообще может и не быть);
- **СИМВОЛ+** - определяет одно и более количество повторений указанного перед плюсом символа.

### Например:

"я изучаю SQL" REGEXP "[a-zA-Z]+\$" - истина, так как в строке есть английские буквы  
 "я изучаю" REGEXP "[a-zA-Z]+\$" - ложь, так как в строке нет английских букв

### Пример

Вывести все уникальные должности, название которых состоит из двух русских слов, разделенных символом "-".

### Запрос:

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP "^[a-яА-Я]+-[a-яА-Я]+$";
```

### Пояснение

Шаблон включает:

- **"^[a-яА-Я]+"** - означает, что название должно начинаться с русского слова, состоящего из одной и более букв ;
- **"-"** - за словом должен следовать символ "-";
- **"[a-яА-Я]+\$"** - название должно заканчиваться русским словом, состоящим из одной и более букв.

### Результат

position
инженер-строитель
логист-диспетчер
продавец-консультант

Affected rows: 3

### Текст задания (чтобы не прокручивать страницу):

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой заканчивается на слово, состоящее из английских букв, цифр или символов "+", "#", ".", расположенных в любом порядке. Слова в названии разделяются пробелом. Вывести должность, на которую претендует соискатель. Информацию отсортировать по фамилии соискателя в алфавитном порядке.

### Результат

applicant	position
Ермоленко Г.С.	программист C#
Краско С.С.	программист 1C
Орлова Е.Р.	head
Сомов Я.Г.	программист C++
Тупикин К.Л.	программист 1.7C

Affected rows: 5

## 1.2.11 Регулярные выражения, наличие или отсутствие символов

### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой включает слова из одного, двух или трех символов. Эти слова состоят из английских букв, цифр и символов "+", "#", ".", расположенных в любом порядке. Слова в названии разделяются пробелом. Вывести профессиональную область и должность, на которую претендует соискатель. Информацию отсортировать по фамилии соискателя в алфавитном порядке.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation,  
position,min_salary, experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца. Синтаксис **REGEXP**:

**строка [NOT] REGEXP регулярное\_выражение**

Регулярное выражение может содержать текст и специальные символы. Если произвольный текст встречается в регулярном выражении, то результат сравнения будет истина, если этот текст встречается в любом месте строки.

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания;
- **[начальный\_символ-конечный\_символ]** - соответствует любому одиночному символу из диапазона символов;
- **[^символы]** строка содержит НЕ символы, перечисленные после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^;
- **[^начальный\_символ-конечный\_символ]** - строка содержит НЕ символы, из интервала после ^, то есть в строке есть какие-то символы, помимо тех, что указаны после ^.
- **|** - отделяет два шаблона строки, и значение должно соответствовать одному из этих шаблонов;

• `.` - соответствует любому непустому символу, если же в шаблоне символ точка "." как элемент текстовой строки, то этот символ необходимо "экранировать", то есть вставить перед символом строки `\\`;

• **СИМВОЛ\*** - определяет любое количество повторений указанного перед звездочкой символа ( в том числе символа вообще может и не быть);

• **СИМВОЛ+** - определяет одно и более количество повторений указанного перед плюсом символа;

• **СИМВОЛ?** - определяет любой символ, в том числе и отсутствие символа.

### Например:

"я изучаю SQL" REGEXP "[a-zA-z]? \$" - истина, так как последняя буква строки - английская

"я изучаю" REGEXP "[a-zA-z]? \$" - истина, так как в соответствии с шаблоном английская буква может отсутствовать на последнем месте в строке

### Пример

Вывести все должности, в которых есть русские слова, длиной не более двух символов. Слова в названии разделяются пробелом.

*Запрос:*

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP "^[a-яА-Я][a-яА-Я]? $|^ [a-яА-Я][a-яА-Я]? | [a-яА-Я][a-яА-Я]? $| [a-яА-Я][a-яА-Я]? " ;
```

### Пояснение

Перед тем, как приступить к составлению шаблона, необходимо определить, где в названии может располагаться слово и чем оно ограничивается.

**1 вариант.** Название состоит из одного слова:

- тогда оно ограничивается началом и концом строки.

**2 вариант.** Название состоит из нескольких слов:

- слово может быть первым в названии, тогда оно ограничивается началом строки перед ним и пробелом после;
- слово может быть последним в названии, тогда оно ограничивается пробелом перед ним и концом строки после;
- слово может находиться в середине строки, тогда оно с двух сторон ограничивается пробелами.

Таким образом получилось 4 возможных положения слов в названии. Для каждого из положений необходимо составить шаблон.

Шаблон включает описание каждого из вариантов:

- "[a-яА-Я][a-яА-Я]?" - означает, что слово состоит обязательно из одной русской буквы, за которой может следовать еще одна, при этом второй буквы может и не быть ;
- "^[a-яА-Я][a-яА-Я]?\$" - одно слово в названии;
- "^[a-яА-Я][a-яА-Я]? " - несколько слов, слово в начале;
- " [a-яА-Я][a-яА-Я]?\$" - несколько слов, слово в конце;
- " [a-яА-Я][a-яА-Я]? " - несколько слов, слово в середине

## Результат

position
менеджер по продажам
программист PHP и Питон
директор по продажам
программист C и Питон

Affected rows: 4

## Текст задания (чтобы не прокручивать страницу):

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой включает слова из одного, двух или трех символов. Эти слова состоят из английских букв, цифр и символов "+", "#", ".", расположенных в любом порядке. Слова в названии разделяются пробелом. Вывести профессиональную область и должность, на которую претендует соискатель. Информацию отсортировать по фамилии соискателя в алфавитном порядке.

## Результат

applicant	specialisation	position
Ермоленко Г.С.	IT	программист C#
Краско С.С.	IT	программист 1C
Левченко А.Г.	IT	PHP разработчик
Паршина Г.И.	IT	программист C и Питон
Садиев С.И.	IT	программист PHP и Питон
Сомов Я.Г.	IT	программист C++

Affected rows: 6

## 1.2.12 Регулярные выражения, специальный символ для выделения слова

### Задание

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой включает слова из одного, двух или трех символов. Эти слова состоят из английских букв и цифр, расположенных в любом порядке. Слова в названии разделяются пробелом, символом "-", знаками препинания и другими символами ("+", "#" и пр.). Вывести профессиональную область и должность, на которую претендует соискатель. Информацию отсортировать по фамилии соискателя в алфавитном порядке.

Это задание необходимо решить, используя в шаблоне `\\b`.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume id, applicant, specialisation,  
position,min_salary, experience, date_birth)
```

### Немного теории

Оператор **REGEXP** позволяет задать регулярное выражение, которому должно соответствовать значение столбца.

Синтаксис **REGEXP** :

**WHERE выражение [NOT] REGEXP регулярное выражение**

В регулярном выражении можно использовать следующие специальные символы:

- **^** - указывает на начало строки;
- **\$** - указывает на конец строки;
- **[символы]** - соответствует любому одиночному символу из квадратных скобок, символы записываются без пробелов и других знаков препинания;
- **[начальный\_символ-конечный\_символ]** - соответствует любому одиночному символу из диапазона символов;
- **[^символы]** строка содержит НЕ символы, перечисленные после **^**, то есть в строке есть какие-то символы, помимо тех, что указаны после **^**;
- **[^начальный\_символ-конечный\_символ]** - строка содержит НЕ символы, из интервала после **^**, то есть в строке есть какие-то символы, помимо тех, что указаны после **^**.
- **|** - отделяет два шаблона строки, и значение должно соответствовать одну из этих шаблонов;
- **.** - соответствует любому символу (не пустому) ;



- **СИМВОЛ\*** - определяет любое количество повторений указанного перед звездочкой символа ( в том числе символа вообще может и не быть);
- **СИМВОЛ+** - определяет одно и более количество повторений указанного перед плюсом символа;
- **СИМВОЛ?** - определяет любой символ, в том числе и отсутствие символа;
- **\\b** - специальный символ, который означает начало строки, конец строки, перенос строки, пробел, запятую, знаки операций и другие символы, которые могут стоять между отдельными словами.

### Например

Запрос:

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP '[a-zA-Z]+\b'
```

Отбирает, все должности, в названии которых есть английские буквы, справа ограниченные либо концом строки, либо пробелом, либо знаками "+", "#" и т.д. А перед этими буквами может идти что угодно: и цифры, и русские буквы и другие символы. Поэтому в нашу выборку попали названия, перед которыми идут цифры:

```
+-----+
| position |
+-----+
| web-программист |
| Python программист |
| программист PHP и Питон |
| PHP разработчик |
| программист 1C |
| программист C и Питон |
| программист 1.7C |
| программист C# |
| программист C++ |
| head |
+-----+
Affected rows: 10
```

Запрос:

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP '\\b[a-zA-Z]+'
```

Отбирает, все должности, в названии которых есть английские буквы, слева ограниченные либо концом началом строки, либо пробелом, либо знаками "+", "#" и т.д. А после этих букв может идти что угодно: и цифры, и русские буквы и другие символы (в нашей таблице нет вариантов, в которых бы после английских букв шли другие символы):

```
+-----+
| position |
+-----+
| web-программист |
| Python программист |
| программист PHP и Питон |
| PHP разработчик |
| программист C и Питон |
| программист C# |
| программист C++ |
| head |
+-----+
Affected rows: 8
```

Запрос:

```
SELECT DISTINCT position
FROM resume
WHERE position REGEXP '\\b[a-zA-Z]+\\b'
```

Отбирает, все должности, в названии которых есть английские буквы, ограниченные справа и слева либо концом строки, либо пробелом, либо знаками "+", "#" и т.д. С помощью этого шаблона можно отбирать слова в строке.

position
web-программист
Python программист
программист PHP и Питон
PHP разработчик
программист C и Питон
программист C#
программист C++
head

Affected rows: 8

### Текст задания (чтобы не прокручивать страницу):

Вывести фамилию и инициалы соискателей, претендующих на должность, название которой включает слова из одного, двух или трех символов. Эти слова состоят из английских букв и цифр, расположенных в любом порядке. Слова в названии разделяются пробелом, символом "-", знаками препинания и другими символами ("+", "#" и пр.). Вывести профессиональную область и должность, на которую претендует соискатель. Информацию отсортировать по фамилии соискателя в алфавитном порядке.

Это задание необходимо решить, используя в шаблоне `\\b`.

### Результат

applicant	specialisation	position
Бондарев С.С.	IT	web-программист
Ермоленко Г.С.	IT	программист C#
Иванов М.С.	IT	web-программист
Краско С.С.	IT	программист 1C
Левченко А.Г.	IT	PHP разработчик
Паршина Г.И.	IT	программист C и Питон
Прохоров А.Д.	IT	web-программист
Садиев С.И.	IT	программист PHP и Питон
Сомов Я.Г.	IT	программист C++
Тупикин К.Л.	IT	программист 1.7C

Affected rows: 10

!!!Обратите внимание, что "программист 1.7C" тоже отбирается, так как здесь словом считается 7C, а точка и конец строки считается ограничителями слова

А в слове C++ словом считается C, а пробел и + являются ограничителями слова!!!

## 1.3 Простые запросы на выборку, база данных склад

### 1.3.1 Структура и наполнение базы данных «Склад»

В этом уроке предлагается написать SQL запросы для таблицы `book`, в которой хранится информация о книгах на некотором складе. Все запросы будут формулироваться для таблицы `book`, которая создана следующим запросом:

```
CREATE TABLE book (  
  book_id INT PRIMARY KEY AUTO_INCREMENT,  
  title VARCHAR(50),  
  author VARCHAR(30),  
  price decimal(8,2),  
  amount INT );
```

Наполнение таблицы `book`:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
...				

Дамп базы данных для этого урока:

```
CREATE TABLE book (  
  book_id INT PRIMARY KEY AUTO_INCREMENT,  
  title VARCHAR(50),  
  author VARCHAR(30),  
  price decimal(8,2),  
  amount INT  
);  
  
INSERT INTO book (title, author, price, amount)  
VALUES  
( 'Мастер и Маргарита', 'Булгаков М.А.', 670.99, 3),  
( 'Белая гвардия', 'Булгаков М.А.', 540.50, 5),  
( 'Идиот', 'Достоевский Ф.М.', 460, 10),  
( 'Братья Карамазовы', 'Достоевский Ф.М.', 799.01, 3),  
( 'Игрок', 'Достоевский Ф.М.', 480.50, 10),  
( 'Стихотворения и поэмы', 'Есенин С.А.', 650.00, 15),  
( 'Таинственный остров', 'Жюль Верн', 169.99, 0),  
( 'Пуаро ведет следствие', 'Агата Кристи', 272.50, 2),  
( 'Евгений Онегин', 'Пушкин А.С.', 200.11, 8),  
( 'Бородино', 'Лермонтов М.Ю.', 316.55, 10),  
( 'Дубровский', 'Пушкин А.С.', 114.99, 7),  
( 'Собачье сердце', 'Булгаков М.А.', 380.90, 6),  
( 'Вокруг света за 80 дней', 'Жюль Верн', 201.01, 15),  
( 'Смерть на Ниле', 'Агата Кристи', 250.12, 18),  
( 'Убийства по алфавиту', 'Агата Кристи', 231.70, 9),  
( 'Загадочное происшествие', 'Агата Кристи', 260.99, 15),  
( 'Капитанская дочка', 'Пушкин А.С.', 199.99, 5),  
( 'Этюд в багровых тонах', 'Дойл Артур Конан', 215.00, 3),  
( 'Приключения Шерлока Холмса', 'Дойл Артур Конан', 590.50, 11),  
( 'Записки о Шерлоке Холмсе', 'Дойл Артур Конан', 470.20, 14),  
( 'Затерянный мир', 'Дойл Артур Конан', 400.00, 3),  
( 'Стихи', 'Лермонтов М.Ю.', 550.60, 6),  
( 'Поэмы', 'Лермонтов М.Ю.', 202.40, 8),  
( 'Герой нашего времени', 'Лермонтов М.Ю.', 479.99, 2),  
( 'Стихи', 'Пушкин А.С.', 600.50, 5),  
( 'Слово милой', 'Пушкин А.С.', 120.30, 12),  
( 'Поэмы', 'Пушкин А.С.', 630.50, 7),  
( 'Скрюченный домишко', 'Агата Кристи', 150.01, 13);
```

### 1.3.2 Запросы с группировкой, MIN(), MAX()

**Задание** (автор - [Игорь Владимирович Лапшин](#))

При анализе остатков книг на складе было решено дополнительно заказать книги авторов, у которых суммарное число экземпляров книг меньше 20. В таблице должны быть отображены авторы, наименьшее и наибольшее количество их книг. Столбцы назвать **Автор**, **Наименьшее\_количество**, **Наибольшее\_количество**. Информацию отсортировать по фамилии автора в алфавитном порядке.

**Структура таблицы:**

book	
PK	book_id
	title
	author
	price
	amount

**Результат**

Автор	Наименьшее_количество	Наибольшее_количество
Булгаков М.А.	3	6
Есенин С.А.	15	15
Жюль Верн	0	15

**Наполнение таблицы** **book**

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460.00	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
16	Загадочное происшествие	Агата Кристи	260.99	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
18	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
21	Затерянный мир	Дойл Артур Конан	400.00	3
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
24	Герой нашего времени	Лермонтов М.Ю.	479.99	2
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	150.01	13

### 1.3.3 Запросы с группировкой, COUNT(), SUM()

Запрос (автор - [Артём Нестеров](#))

Найти авторов, у которых есть не менее двух книг с ценой более 300 и количеством не менее 3 штук на складе. Учитывая только эти книги, вывести автора, количество различных произведений автора, минимальную цену и количество экземпляров книг на складе. Столбцы назвать **Автор**, **Различных\_книг**, **Минимальная\_цена**, **Количество**. Информацию отсортировать по фамилии автора в алфавитном порядке, а затем по убыванию количества.

Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

Результат

Автор	Различных_книг	Минимальная_цена	Количество
Булгаков М.А.	3	380.90	14
Дойл Артур Конан	3	400.00	28
Достоевский Ф.М.	3	460.00	23
Лермонтов М.Ю.	2	316.55	16
Пушкин А.С.	2	600.50	12

### 1.3.4 Запросы с группировкой, SUM()

Задание (автор - [Pavel Dubov](#))

Сколько денег понадобится покупателю, чтобы приобрести все имеющиеся на складе книги, по одному экземпляру каждой? Вывести стоимость покупки и количество купленных книг. Столбцы назвать **Стоимость** и **Количество**.

Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

Результат

Стоимость	Количество
10439.86	27

### 1.3.5 Запросы с группировкой, AVG(), SUM()

Задание (автор - [Артём Нестеров](#))

Выбрать авторов, у которых на складе в наличии более трех наименований произведений. Для таких авторов вывести общее число экземпляров их книг и среднюю цену этих книг (суммарную стоимость разделить на количество экземпляров книг). Столбцы назвать **Автор**, **Количество** и **Средняя\_цена**. Последний столбец округлить до двух знаков после запятой. Информацию отсортировать по фамилии автора в алфавитном порядке.

Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

Результат

Автор	Количество	Средняя_цена
Агата Кристи	57	228.03
Лермонтов М.Ю.	26	348.01
Пушкин А.С.	44	278.76

### 1.3.6 Запросы с группировкой, GROUP\_CONCAT()

Задание

Вывести авторов и перечислить их произведения через точку с запятой в алфавитном порядке. Столбцы назвать **Автор** и **Книги**.

Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Немного теории

С помощью группировки можно соединить значения, относящиеся к одной группе, в одну строку. Для этого используется **групповая** функция **GROUP\_CONCAT()**. Ее синтаксис:

```
GROUP_CONCAT ([DISTINCT] имя_столбца
```

```
[ORDER BY {число | имя столбца} [ASC | DESC]]
[SEPARATOR разделитель]
)
```

В скобках указывается имя столбца, значения которого, если они относятся к одной группе, нужно соединить. Также можно задать последовательность соединения, указав сортировку, и разделитель между значениями. По умолчанию в качестве разделителя используется запятая. Указав ключевое слово **DISTINCT**, можно получить только различные значения.

**Пояснение.** При описании функции в квадратных скобках указаны необязательные части, символ **|** означает "или".

Например, для того чтобы вывести всех различных авторов, у которых указаны инициалы, из таблицы **book** в алфавитном порядке, разделив их пробелом, используется запрос:

```
SELECT GROUP_CONCAT(DISTINCT author ORDER BY author SEPARATOR " ") AS
Авторы
FROM book
WHERE author LIKE "% _._.";
```

### Результат

Авторы
Булгаков М.А. Достоевский Ф.М. Есенин С.А. Лермонтов М.Ю. Пушкин А.С.

### Результат

Автор	Книги
Агата Кристи	Загадочное происшествие; Пуаро ведет следствие; Скрюченный домишко; Смерть ...
Булгаков М.А.	Белая гвардия; Мастер и Маргарита; Собачье сердце
Дойл Артур Конан	Записки о Шерлоке Холмсе; Затерянный мир; Приключения Шерлока Холмса; Этюды ...
Достоевский Ф.М.	Братья Карамазовы; Игрок; Идиот
Есенин С.А.	Стихотворения и поэмы
Жюль Верн	Вокруг света за 80 дней; Таинственный остров
Лермонтов М.Ю.	Бородино; Герой нашего времени; Поэмы; Стихи
Пушкин А.С.	Дубровский; Евгений Онегин; Капитанская дочка; Поэмы; Слово милой; Стихи



### 1.3.7 Запросы с группировкой, GROUP\_CONCAT()

#### Задание

Если на складе есть книги с одинаковыми названиями, вывести это название и авторов, которые написали эти книги. Авторы перечислить в алфавитном порядке через запятую. Столбцы назвать **Книга** и **Авторы**. Информацию отсортировать по названию книги в алфавитном порядке.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Результат

Книга	Авторы
Поэмы	Лермонтов М.Ю., Пушкин А.С.
Стихи	Лермонтов М.Ю., Пушкин А.С.

### 1.3.8 Запросы с группировкой, алгоритм поиска первых/последних записей

#### Задание

Для каждого автора вывести книгу с наименьшей ценой. Столбцы назвать **Автор** и **Книга**. Информацию отсортировать по фамилии автора в алфавитном порядке.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Алгоритм решения

1. С помощью **GROUP\_CONCAT()** объедините в одну строку книги каждого автора, отсортировав их по возрастанию цены,
2. Первая книга в этой строке и будут самой дешёвой. Выделите ее с помощью функции **SUBSTRING\_INDEX()**.

#### Результат

Автор	Книга
Агата Кристи	Скрюченный домишко
Булгаков М.А.	Собачье сердце
Дойл Артур Конан	Этюд в багровых тонах
Достоевский Ф.М.	Идиот
Есенин С.А.	Стихотворения и поэмы
Жюль Верн	Таинственный остров
Лермонтов М.Ю.	Поэмы
Пушкин А.С.	Дубровский

### 1.3.9 Запросы с группировкой, алгоритм поиска первых/последних записей

#### Задание

Для каждого автора вывести книгу с наибольшей ценой. Добавить цену книги к названию через символ "-". Столбцы назвать **Автор** и **Книга**. Информацию отсортировать по фамилии автора в алфавитном порядке.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Результат

Автор	Книга
Агата Кристи	Пуаро ведет следствие-272.50
Булгаков М.А.	Мастер и Маргарита-670.99
Дойл Артур Конан	Приключения Шерлока Холмса-590.50
Достоевский Ф.М.	Братья Карамазовы-799.01
Есенин С.А.	Стихотворения и поэмы-650.00
Жюль Верн	Вокруг света за 80 дней-201.01
Лермонтов М.Ю.	Стихи-550.60
Пушкин А.С.	Поэмы-630.50

## 1.4 Запросы с группировкой, база данных «Резюме»

В этом уроке предлагается написать SQL запросы для таблицы **resume**, в которой хранится информация о резюме для трудоустройства. Каждое резюме содержит имя соискателя, профессиональную область, в которой он ищет работу, должность, минимальную зарплату, стаж работы в этой должности, а также дата рождения соискателя. Дату рождения, как правило, в резюме не указывают, но здесь она включена, чтобы повторить и изучить новые функции работы с датами.

### 1.4.1 Структура и наполнение таблицы

Все запросы будут формулироваться для таблицы **resume**, которая создана следующим запросом:

```
CREATE TABLE resume (  
    resume_id INT PRIMARY KEY AUTO_INCREMENT,  
    applicant VARCHAR(60), # фамилия и инициалы соискателя  
    specialisation VARCHAR(80), # область профессиональной деятельности  
    position VARCHAR(50), # должность  
    min_salary INT, # минимальная заработная плата  
    experience INT, # опыт работы  
    date_birth DATE # дата рождения  
);
```

Наполнение таблицы **resume**:

resume_id	applicant	specialisation	position	min_salary	experience	date_birth
1	Иванов М.С.	IT	web-программист	50000	2	1998-12-15
2	Петров Ф.С.	Строительство	инженер-строитель	45000	5	1995-1-7
3	Федоров П.Р.	Продажи	менеджер по продажам	35000	7	1993-8-26
4	Абрамова А.А.	Продажи	региональный менеджер	90000	4	1992-9-23
5	Самарин С.С.	Транспорт	логист-диспетчер	30000	0	2001-7-18
6	Туполев И.Д.	Строительство	инженер-строитель	40000	10	1985-3-16
7	Баранов П.В.	Юристы	адвокат	60000	5	1982-2-24
8	Шевцова А.Р.	Продажи	менеджер по продажам	30000	2	1995-11-12
9	Белых К.Д.	IT	Python программист	100000	5	1991-12-9
10	Садиев С.И.	IT	программист PHP и Питон	90000	6	1982-11-7
11	Астахов И.И.	Строительство	прораб	100000	15	1976-1-30
12	Борисов В.В.	Продажи	менеджер по продажам	50000	12	1980-3-1
...						

## Дамп базы данных этого урока:

```
CREATE TABLE resume (
    resume_id INT PRIMARY KEY AUTO_INCREMENT,
    applicant VARCHAR(60),
    specialisation VARCHAR(80),
    position VARCHAR(50),
    min_salary INT,
    experience INT,
    date_birth DATE);

INSERT INTO resume(applicant, specialisation, position,min_salary, experience,
date_birth) VALUES
('Иванов М.С.', 'IT', 'web-программист', 50000, 2, '1998-12-15'),
('Петров Ф.С.', 'Строительство', 'инженер-строитель', 45000, 5, '1995-01-07'),
('Федоров П.Р.', 'Продажи', 'менеджер по продажам', 35000, 7, '1993-08-26'),
('Абрамова А.А.', 'Продажи', 'региональный менеджер', 90000, 4, '1992-09-23'),
('Самарин С.С.', 'Транспорт', 'логист-диспетчер', 30000, 0, '2001-07-18'),
('Туполев И.Д.', 'Строительство', 'инженер-строитель', 40000, 10, '1985-03-16'),
('Баранов П.В.', 'Юристы', 'адвокат', 60000, 5, '1982-02-24'),
('Шевцова А.Р.', 'Продажи', 'менеджер по продажам', 30000, 2, '1995-11-12'),
('Белых К.Д.', 'IT', 'Python программист', 100000, 5, '1991-12-09'),
('Садиев С.И.', 'IT', 'программист PHP и Питон', 90000, 6, '1982-11-07'),
('Астахов И.И.', 'Строительство', 'прораб', 100000, 15, '1976-01-30'),
('Борисов В.В.', 'Продажи', 'менеджер по продажам', 50000, 12, '1980-03-01'),
('Тошаков П.С.', 'Строительство', 'технолог', 50000, 15, '1979-04-25'),
('Солиева К.П.', 'Продажи', 'директор по продажам', 130000, 7, '1973-04-30'),
('Левченко А.Г.', 'IT', 'PHP разработчик', 110000, 9, '1967-12-27'),
('Хованец Г.П.', 'Строительство', 'прораб', 90000, 16, '1970-10-01'),
('Тихомиров Т.М.', 'Транспорт', 'экспедитор', 45000, 20, '1975-08-08'),
('Дворников М.А.', 'Транспорт', 'логист-диспетчер', 35000, 2, '1998-11-10'),
('Жидкова Р.Л.', 'Продажи', 'менеджер по продажам', 30000, 5, '1987-10-20'),
('Бондарев С.С.', 'IT', 'web-программист', 130000, 9, '1987-02-16'),
('Иванов П.А.', 'Продажи', 'директор по продажам', 140000, 11, '1979-04-24'),
('Прохоров А.Д.', 'IT', 'web-программист', 200000, 8, '1985-03-23'),
('Соболев А.И.', 'Транспорт', 'водитель', 35000, 15, '1984-03-30'),
('Ромашко Ф.Ю.', 'Продажи', 'продавец-консультант', 25000, 0, '2002-05-07'),
('Краско С.С.', 'IT', 'программист 1С', 85000, 3, '1997-09-02'),
('Паршина Г.И.', 'IT', 'программист C и Питон', 110000, 9, '1981-08-11'),
('Светлова С.Т.', 'Транспорт', 'водитель', 40000, 10, '1991-09-17'),
('Буянов П.Д.', 'Юристы', 'адвокат', 55000, 15, '1980-06-01'),
('Ульянов М.А.', 'Продажи', 'региональный менеджер', 130000, 11, '1979-06-19'),
('Тупикин К.Л.', 'IT', 'программист 1.7C', 140000, 12, '1981-07-03'),
('Ермоленко Г.С.', 'IT', 'программист C#', 90000, 6, '1993-12-09'),
('Сомов Я.Г.', 'IT', 'программист C++', 115000, 8, '1990-11-16'),
('Орлова Е.Р.', 'Строительство', 'head', 170000, 15, '1972-01-16'),
('Ларин В.П.', 'Строительство', 'инженер-строитель', 70000, 12, '1970-05-15');
```

## 1.4.2 Запросы с группировкой, функции DAY(), MONTH(), YEAR(), DAYOFYEAR(), MAKEDATE()

### Задание

Для каждой профессиональной области создать "усредненный" портрет соискателя. Найти количество соискателей, среднюю минимальную заработную плату и "среднюю" дату рождения по каждой профессиональной области. Среднюю минимальную заработную плату округлить до целого. Столбцы назвать **Специализация**, **Количество**, **Мин\_зарплата**, **Дата\_рождения**. Информацию отсортировать по убыванию минимальной заработной платы.

Алгоритм вычисления "средней даты рождения":

- из каждой даты рождения выделить год и найти среднее арифметическое года для каждой профессиональной области;
- из каждой даты рождения выделить количество дней, прошедших с начала года до дня рождения, и найти среднее арифметическое количества дней для каждой профессиональной области;
- построить новую дату по полученным средним значениям.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary, experience, date_birth)
```

### Немного теории

В SQL есть функции, которые выделяют различные части даты (полный список функций для работы с датой можно найти [здесь](#)):

- **DAY (дата)** - выделяет день месяца из даты;
- **MONTH (дата)** - определяет номер месяца, на который приходится дата;
- **YEAR (дата)** - определяет год, на который приходится дата;
- **DAYOFYEAR (дата)** - определяет количество дней от начала года до даты;
- **NOW ()** - возвращает текущую дату и время.

Например

```
DAY('2021-08-02') = 2
MONTH('2021-08-02') = 8
YEAR('2021-08-02') = 2021
DAYOFYEAR('2021-08-02') = 214
```

NOW() = "2021-09-21 13:20:04", запрос с функцией запускался в этот момент времени

Для создания новой даты из отдельных частей используется функция:

- **MAKEDATE (год, количество\_дней\_с\_начала\_года)** - создает дату, год которой указан в первом параметре, а день и месяц вычисляются автоматически по количеству дней, прошедших с начала года. Например:

`MAKEDATE (2021, 364) = "2021-12-30"`

## Пример

Для всех соискателей вывести день их рождения в текущем году и посчитать сколько лет им исполнится.

*Запрос:*

```
SELECT
    applicant,
    YEAR(NOW()) - YEAR(date_birth) AS Лет,
    MAKEDATE(YEAR(NOW()), DAYOFYEAR(date_birth)) AS День_рождения
FROM resume
```

**Результат**(выведен на момент составления задания, у Вас может получиться другое количество лет)

applicant	Лет	День_рождения
Иванов М.С.,	23	2021-12-15
Петров Ф.С.	26	2021-01-07
Федоров П.Р.	28	2021-08-26
Абрамова А.А.	29	2021-09-24
Самарин С.С.	20	2021-07-18
Туполев И.Д.	36	2021-03-16
Баранов П.В.	39	2021-02-24
Шевцова А.Р.	26	2021-11-12
Белых К.Д.	30	2021-12-09
Садиев С.И.	39	2021-11-07
Астахов И.И.	45	2021-01-30
Борисов В.В.	41	2021-03-02
Тошаков П.С.	42	2021-04-25
Солиева К.П.	48	2021-04-30
Левченко А.Г.	54	2021-12-27
Хованец Г.П.	51	2021-10-01
Тихомиров Т.М.	46	2021-08-08
Дворников М.А.	23	2021-11-10
Жидкова Р.Л.	34	2021-10-20
Бондарев С.С.	34	2021-02-16
Иванов П.А.	42	2021-04-24
Прохоров А.Д.	36	2021-03-23
Соболев А.И.	37	2021-03-31
Ромашко Ф.Ю.	19	2021-05-07
Краско С.С.	24	2021-09-02
Паршина Г.И.	40	2021-08-11
Светлова С.Т.	30	2021-09-17
Буянов П.Д.	41	2021-06-02
Ульянов М.А.	42	2021-06-19
Тупикин К.Л.	40	2021-07-03
Ермоленко Г.С.	28	2021-12-09
Сомов Я.Г.	31	2021-11-16
Орлова Е.Р.	49	2021-01-16
Ларин В.П.	51	2021-05-15

**Текст задания** (чтобы не прокручивать страницу):

Для каждой профессиональной области создать "усредненный" портрет соискателя. Найти количество соискателей, среднюю минимальную заработную плату и "среднюю" дату рождения по каждой профессиональной области. Среднюю минимальную заработную плату округлить до целого. Столбцы назвать **Специализация**, **Количество**, **Мин\_зарплата**, **Дата\_рождения**. Информацию отсортировать по убыванию минимальной заработной платы.

**Алгоритм** вычисления "средней даты рождения":

- из каждой даты рождения выделить год и найти среднее арифметическое года для каждой профессиональной области;
- из каждой даты рождения выделить количество дней, прошедших с начала года до дня рождения, и найти среднее арифметическое количества дней для каждой профессиональной области;
- построить новую дату по полученным средним значениям.

### Результат

Специализация	Количество	Мин_зарплата	Дата_рождения
ИТ	11	110909	1987-09-12
Строительство	7	80714	1978-04-03
Продажи	9	73333	1987-07-09
Юристы	2	57500	1981-04-14
Транспорт	5	37000	1990-08-05

Affected rows: 5

### 1.4.3 Запросы с группировкой, функции UNIX\_TIMESTAMP(), FROM\_UNIXTIME()

Для каждой профессиональной области создать "усредненный" портрет соискателя. Найти среднюю минимальную заработную плату и "среднюю" дату рождения по каждой профессиональной области. Среднюю минимальную заработную плату округлить до целого. Столбцы назвать **Специализация**, **Мин\_зарплата**, **Дата\_рождения**. Информацию отсортировать по убыванию минимальной заработной платы.

**Алгоритм** вычисления "средней" даты рождения:

- перевести каждую дату в unix-время;
- найти среднее арифметическое для найденных значений по каждой профессиональной области;
- перевести среднее арифметическое дат в обычный формат.

К сожалению, при таком алгоритме для всех людей, рожденных ранее 1 января 1970 года, будет считаться, что они родились 1 января 1970 года...

#### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary, experience, date_birth)
```

#### Немного теории

Значения типа **DATE** можно представить в форме unix-времени.

**Unix-время** определяется как количество секунд, прошедших с полуночи (00:00:00 UTC) 1 января 1970 года (четверг); этот момент называют «эпохой Unix». Например, 2 августа 2021 года в формате unix-времени - это число 1627862400.

В SQL для перевода из обычного формата времени используется функция

**UNIX\_TIMESTAMP(дата)**. Для обратного перевода используется функция

**FROM\_UNIXTIME(число)**. Например

```
UNIX_TIMESTAMP('2021-08-02') = 1627862400
```

```
UNIX_TIMESTAMP('1964-08-02') = 0
```

```
FROM_UNIXTIME(1527869999) = '2018-06-01 16:19:59'
```

Обратите внимание, что для всех дат ранее 1 января 1970 года и позже 19 января 2038 равно 0.

#### Пример



Посчитать, сколько секунд прошло с дня рождения до текущей даты всех соискателей, рожденных в 21 веке. Для этого перевести в unix-время текущую дату и дату рождения соискателей и найти разницу.

*Запрос:*

```
SELECT
    applicant,
    date_birth,
    UNIX_TIMESTAMP(NOW()) - UNIX_TIMESTAMP(date_birth) AS Количество_секунд
FROM resume
WHERE YEAR(date_birth) >= 2000;
```

### Пояснение

**NOW()** - выдает системное дату и время.

**Результат** (результат выведен на момент составления задания, у Вас получится другое количество секунд)

applicant	date_birth	Количество_секунд
Самарин С.С.	2001-07-18	632466391
Ромашко Ф.Ю.	2002-05-07	607151191

Affected rows: 2

**Текст задания** (чтобы не прокручивать страницу):

Для каждой профессиональной области создать "усредненный" портрет соискателя. Найти среднюю минимальную заработную плату и "среднюю" дату рождения по каждой профессиональной области. Среднюю минимальную заработную плату округлить до целого. Столбцы назвать **Специализация**, **Мин\_зарплата**, **Дата\_рождения**. Информацию отсортировать по убыванию минимальной заработной платы.

**Алгоритм** вычисления "средней" даты рождения:

- перевести каждую дату в unix-время;
- найти среднее арифметическое для найденных значений по каждой профессиональной области;
- перевести среднее арифметическое дат в обычный формат.

К сожалению, при таком алгоритме для всех людей, рожденных ранее 1 января 1970 года, будет считаться, что они родились 1 января 1970 года...

### Результат

Специализация	Мин_зарплата	Дата_рождения
IT	110909	1987-06-05 10:54:32.727200
Строительство	80714	1978-05-25 03:25:42.857100
Продажи	73333	1987-03-08 21:20:00
Юристы	57500	1981-04-13 12:00:00
Транспорт	37000	1990-05-23 09:36:00

Affected rows: 5

## 1.4.4 Запросы с группировкой, функции FROM\_UNIXTIME() с форматированием

### Задание

Для каждой профессиональной области создать "усредненный" портрет соискателя. Найти среднюю минимальную заработную плату и "среднюю" дату рождения по каждой профессиональной области. Среднюю минимальную заработную плату округлить до целого. Дату рождения вывести в формате число, название месяца и год через пробел (например, **3 June 2001**). Столбцы назвать **Специализация**, **Мин\_зарплата**, **Дата\_рождения**. Информацию отсортировать по убыванию минимальной заработной платы.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary, experience, date_birth)
```

### Немного теории

Для перевода unix-времени к обычной дате используется функция **FROM\_UNIXTIME(число)**. При этом дата выдается в формате, при котором указывается дата (год, месяц, день через "-") и время через (час, минуты, секунды через ":"). Например:

```
FROM_UNIXTIME(1527869999) = '2018-06-01 16:19:59'
```

Для того, чтобы вывести результат функции в другом виде, используется формат:

**FROM\_UNIXTIME(число, формат)**. Формат включает:

- **%символ** - специальные символы для обозначения частей даты;
- символы типа " ", "-", ".", "/" - служат для соединения частей даты.

В таблице приведены форматные символы для обозначения дня, месяца и года. [Здесь](#) можно найти описания всех специальных символов.

Specifier	Description
%D	день месяца с английским суффиксом (0th, 1st, 2nd, 3rd, ...)
%d	день месяца в виде двух цифр (00..31)
%e	день месяца (0..31)

Specifier	Description
%M	английское название месяца (January..December)
%m	числовое значение месяца в виде двух цифр (01..12)
%c	числовое значение месяца (1..12)
%Y	значение года, состоящее из 4 цифр
%y	значение года, состоящее из двух последних цифр

*Например:*

```
FROM_UNIXTIME(1527869999, "%d.%m.%Y") = "01.06.2018"
FROM_UNIXTIME(1527869999, "%D %M %y") = "1st June 18"
FROM_UNIXTIME(1527869999, "%e-%c-%Y") = "1-6-2018"
```

**Текст задания** (чтобы не прокручивать страницу):

Для каждой профессиональной области создать "усредненный" портрет соискателя. Найти среднюю минимальную заработную плату и "среднюю" дату рождения по каждой профессиональной области. Среднюю минимальную заработную плату округлить до целого. Дату рождения вывести в формате число, название месяца и год через пробел (например, **3 June 2001**). Столбцы назвать **Специализация**, **Мин\_зарплата**, **Дата\_рождения**. Информацию отсортировать по убыванию минимальной заработной платы.

Алгоритм вычисления "средней" даты рождения:

- перевести каждую дату в unix-время;
- найти среднее арифметическое для найденных значений по каждой профессиональной области;
- перевести среднее арифметическое дат в обычный формат.

При таком алгоритме для всех людей, рожденных ранее 1 января 1970 года, будет считаться, что они родились 1 января 1970 года...

## Результат

Специализация	Мин_зарплата	Дата_рождения
IT	110909	5 June 1987
Строительство	80714	25 May 1978
Продажи	73333	8 March 1987
Юристы	57500	13 April 1981
Транспорт	37000	23 May 1990

Affected rows: 5

## 1.4.4 Запросы с группировкой, функции работы с датой, пример

### Задание

Найти разницу в возрасте (в годах) между самым молодым и самым возрастным соискателем в каждой профессиональной области. Вывести название профессиональной области и разницу в возрасте. Последний столбец назвать **Количество\_лет**. Информацию отсортировать по возрастанию разницы.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_bith

```
resume(resume_id, applicant, specialisation, position,min_salary,
        experience, date_birth)
```

### Пояснение

Для вычисления разницы в возрасте необходимо от максимального года рождения соискателей каждой профессиональной области отнять минимальный.

### Результат

specialisation	Количество лет
Юристы	2
Строительство	25
Транспорт	26
Продажи	29
IT	31

Affected rows: 5

## 1.4.5 Запросы с группировкой, функции работы с датой, пример

### Задание

Вывести профессиональные области, в которых есть соискатели, не меняющие специализацию. Посчитать, сколько соискателей не меняли специализацию. Если человек работал в указанной профессиональной области с **23** лет и ранее - будем считать, что он специализацию не менял (в поле **experience** хранится количество лет, которое человек работает по указанной специализации). Текущим считать **2021** год. Вывести профессиональную область и вычисленное количество таких соискателей, последний столбец назвать **Количество**. Информацию отсортировать по убыванию количества соискателей.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position,min_salary,
        experience, date_birth)
```

### Результат

```
+-----+
| specialisation | Количество |
+-----+
| IT             | 4         |
| Транспорт     | 4         |
| Продажи       | 2         |
| Строительство | 1         |
+-----+
```

Affected rows: 4

## 1.4.6 Запросы с группировкой, алгоритм поиска лучшей/худшей записи

### Задание

Для каждой должности найти "лучшего" кандидата. Кандидат считается "лучшим", если у него самый большой опыт работы. Если опыт работы одинаков "лучшим" считается тот, у кого минимальная заработная плата ниже. Если совпадает и эта характеристика, предпочтение отдать более молодому соискателю. Вывести профессиональную область, должность и фамилию и имя отчество соискателя. Последний столбец назвать **Кандидат**. Информацию отсортировать сначала по профессиональной области в алфавитном порядке, затем по названию должности в алфавитном порядке.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position,min_salary, experience, date_birth)
```

### Пояснение

Возможный алгоритм решения.

1. Для каждой области получить упорядоченный по заданным правилам список соискателей в одной строке, используя групповую функцию **GROUP\_CONCAT()**.
2. Выделить первую фамилию соискателя с помощью **SUBSTRING\_INDEX()**.

### Результат

specialisation	position	Кандидат
IT	PHP разработчик	Левченко А.Г.
IT	Python программист	Белых К.Д.
IT	web-программист	Бондарев С.С.
IT	программист 1.7C	Тупикин К.Л.
IT	программист 1C	Краско С.С.
IT	программист C и Питон	Паршина Г.И.
IT	программист C#	Ермоленко Г.С.
IT	программист C++	Сомов Я.Г.
IT	программист PHP и Питон	Садиев С.И.
Продажи	директор по продажам	Иванов П.А.
Продажи	менеджер по продажам	Борисов В.В.
Продажи	продавец-консультант	Ромашко Ф.Ю.
Продажи	региональный менеджер	Ульянов М.А.
Строительство	head	Орлова Е.Р.
Строительство	инженер-строитель	Ларин В.П.
Строительство	прораб	Хованец Г.П.
Строительство	технолог	Тошаков П.С.
Транспорт	водитель	Соболев А.И.
Транспорт	логист-диспетчер	Дворников М.А.
Транспорт	экспедитор	Тихомиров Т.М.
Юристы	адвокат	Буянов П.Д.

Affected rows: 21

## 1.5 Простые запросы на выборку, база данных склад

### 1.5.1 Структура и наполнение базы данных «Склад»

В этом уроке предлагается написать SQL запросы для таблицы `book`, в которой хранится информация о книгах на некотором складе. Все запросы будут формулироваться для таблицы `book`, которая создана следующим запросом:

```
CREATE TABLE book (  
  book_id INT PRIMARY KEY AUTO_INCREMENT,  
  title VARCHAR(50),  
  author VARCHAR(30),  
  price decimal(8,2),  
  amount INT );
```

Наполнение таблицы `book`:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
...				

Дамп базы данных для этого урока:

```
CREATE TABLE book (
  book_id INT PRIMARY KEY AUTO_INCREMENT,
  title VARCHAR(50),
  author VARCHAR(30),
  price decimal(8,2),
  amount INT
);

INSERT INTO book (title, author, price, amount)
VALUES
('Мастер и Маргарита', 'Булгаков М.А.', 670.99, 3),
('Белая гвардия', 'Булгаков М.А.', 540.50, 5),
('Идиот', 'Достоевский Ф.М.', 460, 10),
('Братья Карамазовы', 'Достоевский Ф.М.', 799.01, 3),
('Игрок', 'Достоевский Ф.М.', 480.50, 10),
('Стихотворения и поэмы', 'Есенин С.А.', 650.00, 15),
('Таинственный остров', 'Жюль Верн', 169.99, 0),
('Пуаро ведет следствие', 'Агата Кристи', 272.50, 2),
('Евгений Онегин', 'Пушкин А.С.', 200.11, 8),
('Бородино', 'Лермонтов М.Ю.', 316.55, 10),
('Дубровский', 'Пушкин А.С.', 114.99, 7),
('Собачье сердце', 'Булгаков М.А.', 380.90, 6),
('Вокруг света за 80 дней', 'Жюль Верн', 201.01, 15),
('Смерть на Ниле', 'Агата Кристи', 250.12, 18),
('Убийства по алфавиту', 'Агата Кристи', 231.70, 9),
('Загадочное происшествие', 'Агата Кристи', 260.99, 15),
('Капитанская дочка', 'Пушкин А.С.', 199.99, 5),
('Этюд в багровых тонах', 'Дойл Артур Конан', 215.00, 3),
('Приключения Шерлока Холмса', 'Дойл Артур Конан', 590.50, 11),
('Записки о Шерлоке Холмсе', 'Дойл Артур Конан', 470.20, 14),
('Затерянный мир', 'Дойл Артур Конан', 400.00, 3),
('Стихи', 'Лермонтов М.Ю.', 550.60, 6),
('Поэмы', 'Лермонтов М.Ю.', 202.40, 8),
('Герой нашего времени', 'Лермонтов М.Ю.', 479.99, 2),
('Стихи', 'Пушкин А.С.', 600.50, 5),
('Слово милой', 'Пушкин А.С.', 120.30, 12),
('Поэмы', 'Пушкин А.С.', 630.50, 7),
('Скрюченный домишко', 'Агата Кристи', 150.01, 13);
```

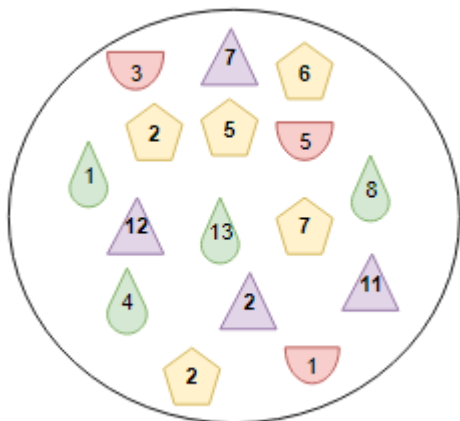


## 1.5.2 Операторы ALL и ANY

Вложенный запрос, возвращающий несколько значений одного столбца, можно использовать для отбора записей с помощью операторов **ANY** и **ALL** совместно с операциями отношения (**=**, **!=**, **<=**, **>=**, **<**, **>**).

Рассмотрим, как работают эти операторы.

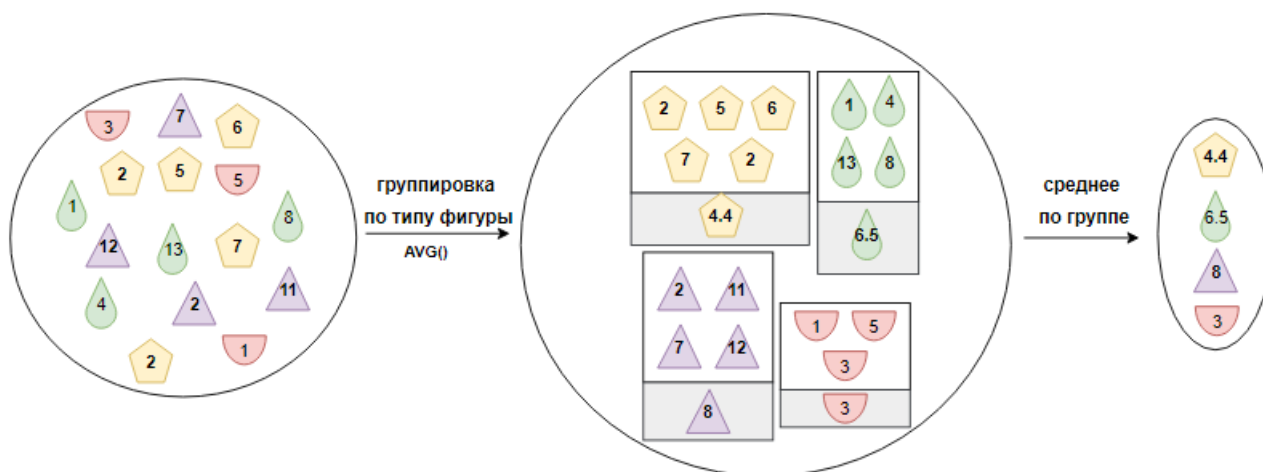
Пусть у нас есть некоторый набор данных, который включает фигуры разной формы, каждая фигура имеет свой вес:



Для этого набора данных найдем средний вес фигур каждого типа. Для этого в SQL используется запрос:

```
SELECT AVG(вес)
FROM набор_фигур
GROUP BY тип_фигур
```

Сначала данные группируются по типу фигуры, а затем вычисляется среднее в каждой группе. Результатом будет новый набор данных, который включает средний вес фигур каждого типа.



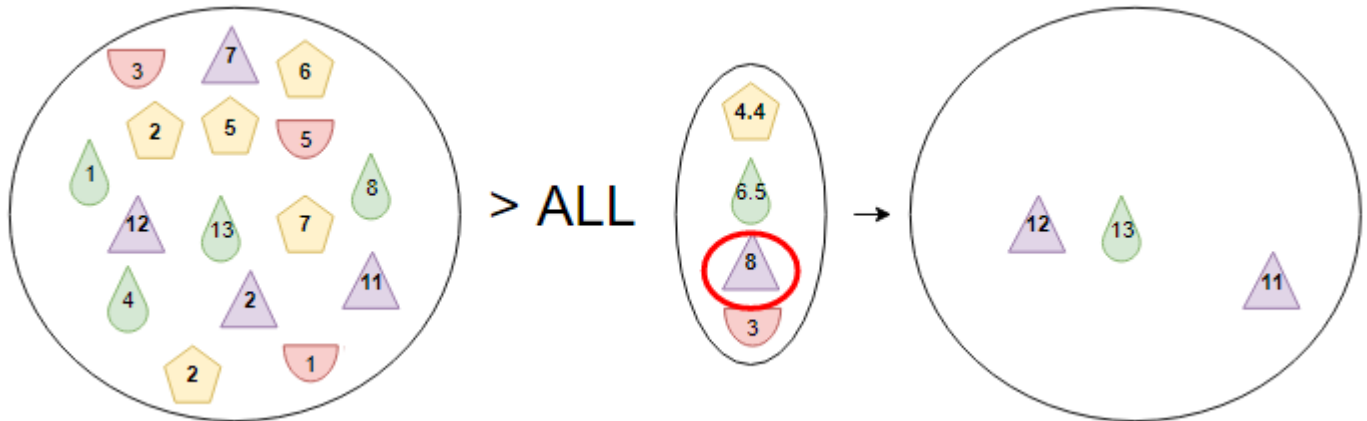
Далее нам необходимо отобрать фигуры из исходного набора данных по некоторому условию, связанному с набором данных со средним весом. Для этого используется запрос:

```
SELECT фигуры
FROM набор_фигур
WHERE вес знак_отношения ANY|ALL (
```

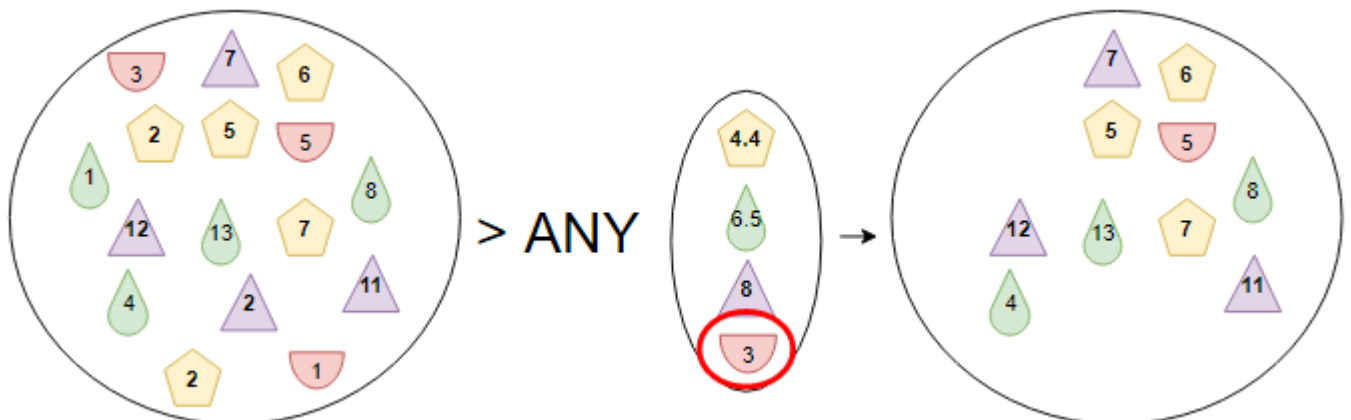
```
SELECT AVG (вес)
FROM набор фигур
GROUP BY тип_фигур
);
```

С помощью этого запроса, подставляя различные знаки отношения совместно с **ANY** или **ALL** можно выполнить следующие действия:

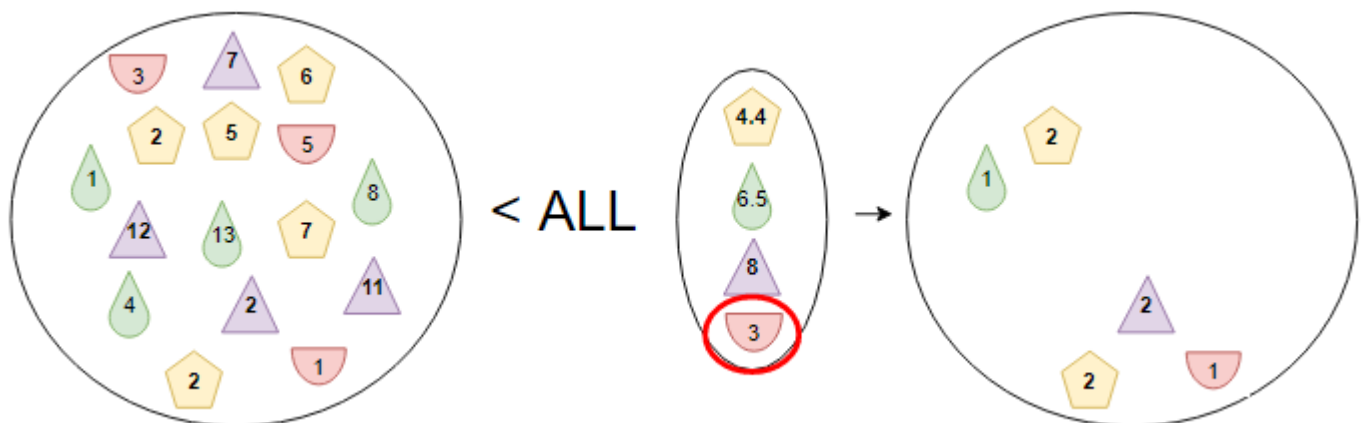
- отобрать фигуры, вес которых **больше самого большого** среднего веса среди всех средних весов каждого типа фигур:



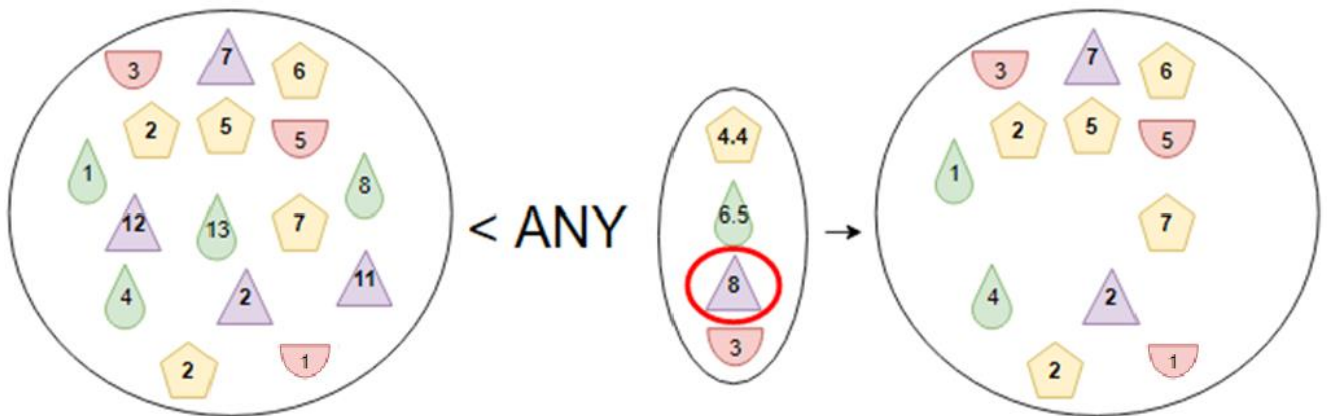
- отобрать фигуры, вес которых **больше самого маленького** среднего веса среди всех средних весов каждого типа фигур:



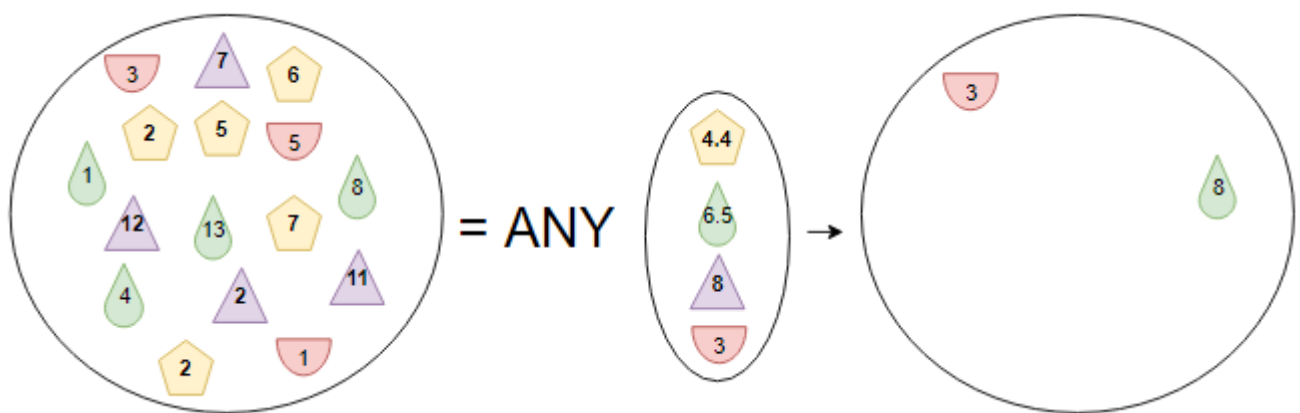
- отобрать фигуры, вес которых **меньше самого маленького** среднего веса среди всех средних весов каждого типа фигур:



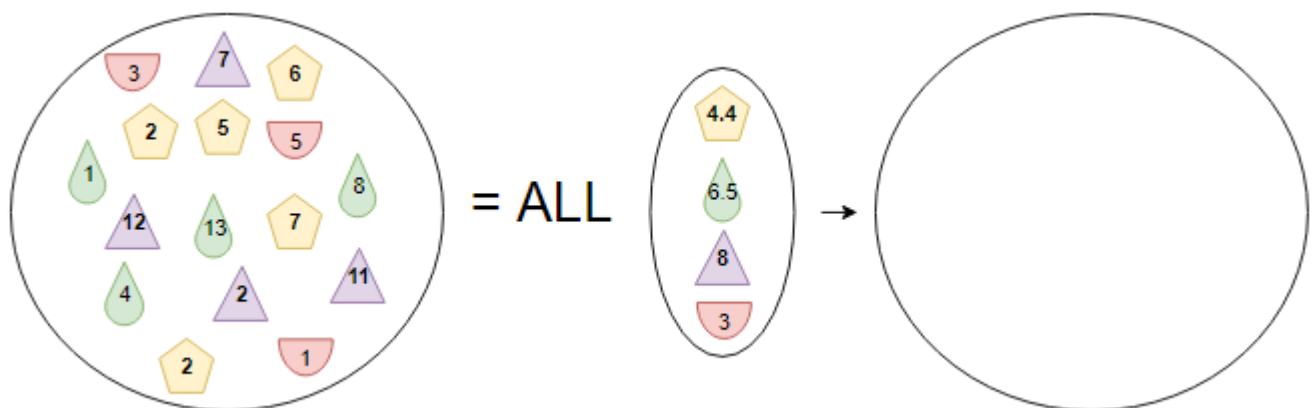
- отобрать фигуры, вес которых **меньше самого большого** среднего веса среди всех средних весов каждого типа фигур:



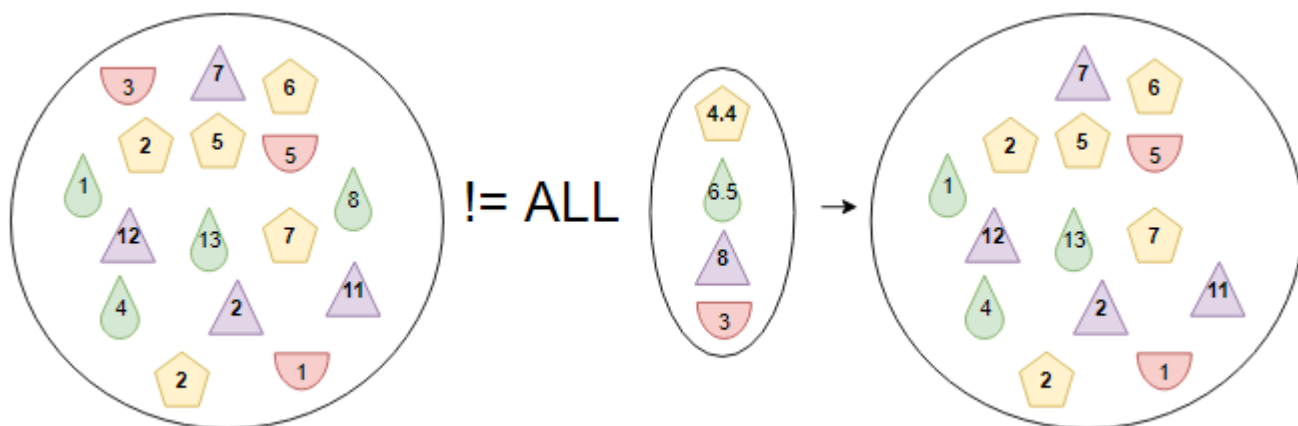
- отобрать фигуры, веса которых **равны любому** среднему весу каждого типа фигур:



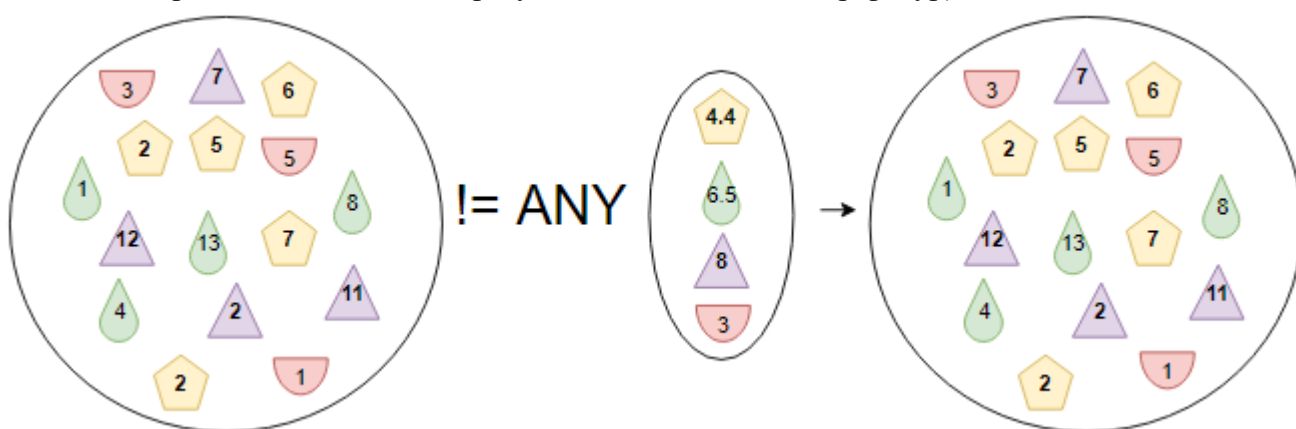
- отобрать фигуры, веса которых **одновременно равны каждому** среднему весу каждого типа фигур (никакое значение не может быть равно одновременно нескольким значениям):



- отобрать фигуры, веса которых **не равны любому** среднему весу каждого типа фигур (результат - исходный набор данных без тех элементов, которые равны любому из средних весов фигур):



- отобрать фигуры, веса которых **не равны каждому одновременно** среднему весу каждого типа фигур (поскольку значение может быть равно только одному значению, а остальным - не равно, следовательно результат - исходный набор фигур):



### 1.5.3 Вложенные запросы, оператор ALL

#### Задание

Вывести информацию о тех книгах, цена которых меньше самой маленькой средней цены книги каждого автора. Среднее вычислить как общая стоимость экземпляров книг каждого автора, деленная на общее количество экземпляров книг того же автора. Столбцы назвать **Книга**, **Автор**, **Цена**. Информацию отсортировать сначала по названию книг в алфавитном порядке, а затем по убыванию цены.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Немного теории

##### Пример

Вывести информацию о тех книгах, количество которых меньше самого маленького среднего количества книг каждого автора.

Это задание можно решить двумя способами:

**1 вариант**, с использованием двух вложенных запросов по следующему алгоритму:

- посчитать среднее количество книг каждого автора;
- найти среди них минимальное;
- отобрать все книги, количество которых меньше минимального.

*Запрос:*

```
SELECT title, author, amount
FROM book
WHERE amount < (
    # минимальное среднее количество книг
    SELECT MIN(avg_amount)
    FROM (
        # среднее количество книг каждого автора
        SELECT AVG(amount) AS avg_amount
        FROM book
        GROUP BY author
    ) query_in
);
```

**2 вариант**, с использованием одного вложенного запроса и оператора **ALL** по следующему алгоритму:

- посчитать среднее количество книг каждого автора;
- отобрать все книги, количество которых меньше самого маленького среднего

значения с помощью оператора **ALL**.

```

SELECT title, author, amount
FROM book
WHERE amount < ALL(
    # среднее количество книг каждого автора
    SELECT AVG(amount) AS avg_amount
    FROM book
    GROUP BY author
);

```

Несмотря на то, что оба решения верные, предпочтительнее использовать вариант с оператором **ALL**.

### Результат

Чтобы легче было оценить результат, сначала приведем результат вложенного запроса, вычисляющего среднее количество книг каждого автора:

```

+-----+
| avg_amount |
+-----+
| 4.6667     |
| 7.6667     |
| 15.0000    |
| 7.5000     |
| 11.4000    |
| 7.3333     |
| 6.5000     |
| 7.7500     |
+-----+

```

Результат запроса присера:

```

+-----+-----+-----+
| title          | author          | amount |
+-----+-----+-----+
| Мастер и Маргарита | Булгаков М.А.   | 3      |
| Братья Карамазовы  | Достоевский Ф.М. | 3      |
| Таинственный остров | Жюль Верн      | 0      |
| Пуаро ведет следствие | Агата Кристи   | 2      |
| Этьюд в багровых тонах | Дойл Артур Конан | 3      |
| Затерянный мир     | Дойл Артур Конан | 3      |
| Герой нашего времени | Лермонтов М.Ю.  | 2      |
+-----+-----+-----+

```

Affected rows: 7

**Текст задания** (чтобы не прокручивать страницу) :

Вывести информацию о тех книгах, цена которых меньше самой маленькой средней цены книги каждого автора. Среднее вычислить как общая стоимость экземпляров книг каждого автора, деленная на общее количество экземпляров книг того же автора. Столбцы назвать **Книга**, **Автор**, **Цена**. Информацию отсортировать сначала по названию книг в алфавитном порядке, а затем по убыванию цены.

Используйте вложенный запрос с оператором **ALL**.

### Результат

```

+-----+-----+-----+
| Книга          | Автор          | Цена   |
+-----+-----+-----+
| Дубровский     | Пушкин А.С.   | 114.99 |
| Евгений Онегин  | Пушкин А.С.   | 200.11 |
| Капитанская дочка | Пушкин А.С.   | 199.99 |
| Скрюченный домишко | Агата Кристи  | 150.01 |
| Слово милой     | Пушкин А.С.   | 120.30 |
| Таинственный остров | Жюль Верн    | 169.99 |
+-----+-----+-----+

```

Affected rows: 6

## 1.5.4 Вложенные запросы, оператор ANY

### Задание

Вывести информацию о тех книгах, цена которых больше самой маленькой средней цены книги каждого автора. Среднее вычислить как общая стоимость книг каждого автора, деленная на общее количество книг того же автора. Столбцы назвать **Книга**, **Автор**, **Цена**. Информацию отсортировать сначала по названию книг в алфавитном порядке, а затем по убыванию цены.

### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

### Пример

Вывести информацию о тех книгах, количество которых меньше самого большого среднего количества книг каждого автора.

Это задание можно решить двумя способами:

**1 вариант**, с использованием двух вложенных запросов по следующему алгоритму:

- посчитать среднее количество книг каждого автора;
- найти среди них максимальное;
- отобрать все книги, количество которых меньше максимального.

*Запрос:*

```
SELECT title, author, amount
FROM book
WHERE amount < (
    # максимальное среднее количество книг
    SELECT MAX(avg amount)
    FROM (
        # среднее количество книг каждого автора
        SELECT AVG(amount) AS avg_amount
        FROM book
        GROUP BY author
    ) query_in
);
```

**2 вариант**, с использованием одного вложенного запроса и оператора **ANY** по следующему алгоритму:

- посчитать среднее количество книг каждого автора;
- отобрать все книги, количество которых меньше самого большого среднего значения с помощью оператора **ANY**.

```

SELECT title, author, amount
FROM book
WHERE amount < ANY(
    # среднее количество книг каждого автора
    SELECT AVG(amount) AS avg_amount
    FROM book
    GROUP BY author
);

```

Несмотря на то, что оба решения верные, предпочтительнее использовать вариант с оператором **ANY**.

## Результат

Чтобы легче было оценить результат, сначала приведем результат вложенного запроса, вычисляющего среднее количество книг каждого автора:

```

+-----+
| avg_amount |
+-----+
| 4.6667     |
| 7.6667     |
| 15.0000    |
| 7.5000     |
| 11.4000    |
| 7.3333     |
| 6.5000     |
| 7.7500     |
+-----+

```

Результат запроса примера:

```

+-----+-----+-----+
| title                | author          | amount |
+-----+-----+-----+
| Мастер и Маргарита   | Булгаков М.А.  | 3      |
| Белая гвардия        | Булгаков М.А.  | 5      |
| Идиот                | Достоевский Ф.М. | 10     |
| Братья Карамазовы    | Достоевский Ф.М. | 3      |
| Игрок                | Достоевский Ф.М. | 10     |
| Таинственный остров | Жюль Верн      | 0      |
| Пуаро ведет следствие | Агата Кристи   | 2      |
| Евгений Онегин       | Пушкин А.С.    | 8      |
| Бородино             | Лермонтов М.Ю. | 10     |
| Дубровский           | Пушкин А.С.    | 7      |
| Собачье сердце       | Булгаков М.А.  | 6      |
| Убийства по алфавиту | Агата Кристи   | 9      |
| Капитанская дочка    | Пушкин А.С.    | 5      |
| Эюд в багровых тонах | Дойл Артур Конан | 3      |
| Приключения Шерлока Холмса | Дойл Артур Конан | 11     |
| Записки о Шерлоке Холмсе | Дойл Артур Конан | 14     |
| Затерянный мир       | Дойл Артур Конан | 3      |
| Стихи                | Лермонтов М.Ю. | 6      |
| Поэмы                | Лермонтов М.Ю. | 8      |
| Герой нашего времени | Лермонтов М.Ю. | 2      |
| Стихи                | Пушкин А.С.    | 5      |
| Слово милой          | Пушкин А.С.    | 12     |
| Поэмы                | Пушкин А.С.    | 7      |
| Скрюченный домишко   | Агата Кристи   | 13     |
+-----+-----+-----+

```

Affected rows: 24

## Текст задания (чтобы не прокручивать страницу):

Вывести информацию о тех книгах, цена которых больше самой маленькой средней цены книги каждого автора. Среднее вычислить как общая стоимость книг каждого автора, деленная на общее количество книг того же автора. Столбцы назвать **Книга**, **Автор**, **Цена**. Информацию отсортировать сначала по названию книг в алфавитном порядке, а затем по убыванию цены.



Используйте вложенный запрос с оператором **ANY**.

## Результат

Книга	Автор	Цена
Белая гвардия	Булгаков М.А.	540.50
Бородино	Лермонтов М.Ю.	316.55
Братья Карамазовы	Достоевский Ф.М.	799.01
Герой нашего времени	Лермонтов М.Ю.	479.99
Загадочное происшествие	Агата Кристи	260.99
Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20
Затерянный мир	Дойл Артур Конан	400.00
Игрок	Достоевский Ф.М.	480.50
Идиот	Достоевский Ф.М.	460.00
Мастер и Маргарита	Булгаков М.А.	670.99
Поэмы	Пушкин А.С.	630.50
Поэмы	Лермонтов М.Ю.	202.40
Приключения Шерлока Холмса	Дойл Артур Конан	590.50
Пуаро ведет следствие	Агата Кристи	272.50
Смерть на Ниле	Агата Кристи	250.12
Собачье сердце	Булгаков М.А.	380.90
Стихи	Пушкин А.С.	600.50
Стихи	Лермонтов М.Ю.	550.60
Стихотворения и поэмы	Есенин С.А.	650.00
Убийства по алфавиту	Агата Кристи	231.70
Этюд в багровых тонах	Дойл Артур Конан	215.00

Affected rows: 21

### 1.5.5 Вложенные запросы, оператор ANY

#### Задание (автор - [Adilbek Kaliyev](#))

Составить рейтинг книг в зависимости от того, какая книга принесет больше всего выручки (в процентах), при условии продажи всех книг. Выручка в процентах вычисляется как стоимость всех экземпляров книги деленная на суммарную стоимость всех экземпляров книг на складе и умноженное на 100. Полученное значение округлить до двух знаков после запятой. Вывести автора, название книги, цена, количество и выручка в процентах. Последний столбец назвать `income_percent`. Информацию отсортировать по убыванию выручки.

При решении задания используйте вложенный запрос.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

#### Результат

author	title	price	amount	income_percent
Есенин С.А.	Стихотворения и поэмы	650.00	15	12.07
Дойл Артур Конан	Записки о Шерлоке Холмсе	470.20	14	8.15
Дойл Артур Конан	Приключения Шерлока Холмса	590.50	11	8.04
Достоевский Ф.М.	Игрок	480.50	10	5.95
Достоевский Ф.М.	Идиот	460.00	10	5.69
Агата Кристи	Смерть на Ниле	250.12	18	5.57
Пушкин А.С.	Поэмы	630.50	7	5.46
Агата Кристи	Загадочное происшествие	260.99	15	4.84
Лермонтов М.Ю.	Стихи	550.60	6	4.09
Лермонтов М.Ю.	Бородино	316.55	10	3.92
Жюль Верн	Вокруг света за 80 дней	201.01	15	3.73
Пушкин А.С.	Стихи	600.50	5	3.72
Булгаков М.А.	Белая гвардия	540.50	5	3.34
Достоевский Ф.М.	Братья Карамазовы	799.01	3	2.97
Булгаков М.А.	Собачье сердце	380.90	6	2.83
Агата Кристи	Убийства по алфавиту	231.70	9	2.58
Булгаков М.А.	Мастер и Маргарита	670.99	3	2.49
Агата Кристи	Скрюченный домишко	150.01	13	2.41
Лермонтов М.Ю.	Поэмы	202.40	8	2.00
Пушкин А.С.	Евгений Онегин	200.11	8	1.98
Пушкин А.С.	Слово милой	120.30	12	1.79
Дойл Артур Конан	Затерянный мир	400.00	3	1.49
Пушкин А.С.	Капитанская дочка	199.99	5	1.24
Лермонтов М.Ю.	Герой нашего времени	479.99	2	1.19
Пушкин А.С.	Дубровский	114.99	7	1.00
Дойл Артур Конан	Этюд в багровых тонах	215.00	3	0.80
Агата Кристи	Пуаро ведет следствие	272.50	2	0.67
Жюль Верн	Таинственный остров	169.99	0	0.00

Affected rows: 28

## 1.5.6 Табличные выражения, вложенные запросы

### Задание(автор - [Adilbek Kaliyev](#))

Составить рейтинг книг в зависимости от того, какая книга принесет больше всего выручки (в процентах), при условии продажи всех книг. Выручка в процентах вычисляется как стоимость всех экземпляров книги, деленная на суммарную стоимость всех экземпляров книг на складе и умноженное на 100. Полученное значение округлить до двух знаков после запятой. Вывести автора, название книги, цена, количество и выручка в процентах. Последний столбец назвать `income_percent`. Информацию отсортировать по убыванию выручки.

При решении задания используйте табличное выражение.

#### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

Табличное выражение определяется с помощью оператора `WITH` и является частью запроса. Его синтаксис:

```
WITH имя_выражения (имя_1, имя_2, ...)
AS
(
    SELECT столбец_1, столбец_2,
    FROM
        ...
)
SELECT ...
FROM ... имя_выражения
```

В табличном выражении определяется запрос, результат которого нужно использовать в основной части запроса после `SELECT`. При этом основной запрос может обратиться к столбцам результата табличного выражения через имена, заданные в заголовке `WITH`. При этом количество имен должно совпадать с количеством результирующих столбцов табличного выражения.

В табличном выражении необязательно давать имена столбцам результата. В этом случае в основном запросе можно использовать имена столбцов, указанных после `SELECT` в табличном выражении. При наличии одинаковых имен в нескольких табличных выражениях необходимо использовать полное имя столбца (имя табличного выражения, точка, имя столбца).

С помощью табличного выражения можно решать все задачи, которые используют вложенный запрос.

### Пример

Для каждой книги вывести количество экземпляров книг, которое нужно заказать, чтобы суммарное количество экземпляров каждой книги стало одинаковым и равным максимальному количеству экземпляров книг на складе.

*Запрос (с вложенным запросом):*

```
SELECT
    author, title,
    (SELECT MAX(amount) FROM book) - amount AS Заказ
FROM
    book
WHERE amount != (SELECT MAX(amount) FROM book)
```

В данном решении один и тот же вложенный запрос включается в основной запрос дважды. Табличное выражение позволяет один раз написать запрос и затем использовать его результаты любое количество раз. Если в запросе используется несколько таблиц или табличных выражений, то они обязательно перечисляются во **FROM**.

*Запрос (с табличным выражением):*

```
WITH get_max_amount(max_amount)
AS (
    SELECT MAX(amount)
    FROM book
)
SELECT
    author, title, max_amount - amount AS Заказ
FROM
    book, get_max_amount
WHERE amount != max_amount
```

### Результат

author	title	Заказ
Булгаков М.А.	Мастер и Маргарита	15
Булгаков М.А.	Белая гвардия	13
Достоевский Ф.М.	Идиот	8
Достоевский Ф.М.	Братья Карамазовы	15
Достоевский Ф.М.	Игрок	8
Есенин С.А.	Стихотворения и поэмы	3
Жюль Верн	Таинственный остров	18
Агата Кристи	Пуаро ведет следствие	16
Пушкин А.С.	Евгений Онегин	10
Лермонтов М.Ю.	Бородино	8
Пушкин А.С.	Дубровский	11
Булгаков М.А.	Собачье сердце	12
Жюль Верн	Вокруг света за 80 дней	3
Агата Кристи	Убийства по алфавиту	9
Агата Кристи	Загадочное происшествие	3
Пушкин А.С.	Капитанская дочка	13
Дойл Артур Конан	Этюд в багровых тонах	15
Дойл Артур Конан	Приключения Шерлока Холмса	7
Дойл Артур Конан	Записки о Шерлоке Холмсе	4
Дойл Артур Конан	Затерянный мир	15
Лермонтов М.Ю.	Стихи	12
Лермонтов М.Ю.	Поэмы	10
Лермонтов М.Ю.	Герой нашего времени	16
Пушкин А.С.	Стихи	13
Пушкин А.С.	Слово милой	6
Пушкин А.С.	Поэмы	11
Агата Кристи	Скрюченный домишко	5

Affected rows: 27

**Текст задания** (чтобы не прокручивать страницу):

Составить рейтинг книг в зависимости от того, какая книга принесет больше всего выручки (в процентах), при условии продажи всех книг. Выручка в процентах вычисляется как стоимость всех экземпляров книги, деленная на суммарную стоимость всех экземпляров книг на складе и умноженное на 100. Полученное значение округлить до двух знаков после запятой. Вывести автора, название книги, цена, количество и выручка в процентах. Последний столбец назвать `income_percent`. Информацию отсортировать по убыванию выручки.

При решении задания используйте табличное выражение.

### Результат

author	title	price	amount	income_percent
Есенин С.А.	Стихотворения и поэмы	650.00	15	12.07
Дойл Артур Конан	Записки о Шерлоке Холмсе	470.20	14	8.15
Дойл Артур Конан	Приключения Шерлока Холмса	590.50	11	8.04
Достоевский Ф.М.	Игрок	480.50	10	5.95
Достоевский Ф.М.	Идиот	460.00	10	5.69
Агата Кристи	Смерть на Ниле	250.12	18	5.57
Пушкин А.С.	Поэмы	630.50	7	5.46
Агата Кристи	Загадочное происшествие	260.99	15	4.84
Лермонтов М.Ю.	Стихи	550.60	6	4.09
Лермонтов М.Ю.	Вородино	316.55	10	3.92
Жюль Верн	Вокруг света за 80 дней	201.01	15	3.73
Пушкин А.С.	Стихи	600.50	5	3.72
Булгаков М.А.	Белая гвардия	540.50	5	3.34
Достоевский Ф.М.	Братья Карамазовы	799.01	3	2.97
Булгаков М.А.	Собачье сердце	380.90	6	2.83
Агата Кристи	Убийства по алфавиту	231.70	9	2.58
Булгаков М.А.	Мастер и Маргарита	670.99	3	2.49
Агата Кристи	Скрюченный домишко	150.01	13	2.41
Лермонтов М.Ю.	Поэмы	202.40	8	2.00
Пушкин А.С.	Евгений Онегин	200.11	8	1.98
Пушкин А.С.	Слово милой	120.30	12	1.79
Дойл Артур Конан	Затерянный мир	400.00	3	1.49
Пушкин А.С.	Капитанская дочка	199.99	5	1.24
Лермонтов М.Ю.	Герой нашего времени	479.99	2	1.19
Пушкин А.С.	Дубровский	114.99	7	1.00
Дойл Артур Конан	Этюд в багровых тонах	215.00	3	0.80
Агата Кристи	Пуаро ведет следствие	272.50	2	0.67
Жюль Верн	Таинственный остров	169.99	0	0.00

Affected rows: 28

## 1.5.7. Вложенные запросы, табличные выражения, запросы без FROM

### Задание

Посчитать, сколько экземпляров книг на складе имеют цену меньше 500 рублей, а сколько больше или равно. Столбцы назвать `less_500`, `more_500`.

При решении используйте вложенные запросы.

### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

В запросе может отсутствовать раздел **FROM** в том случае,

- если выводятся значения, вычисленные не на основе данных таблицы;
- выводится значение, вычисленные вложенным запросом, возвращающим одно значение.

### Пример

Вывести текущее время и версию MySQL на платформе Stepik.

*Запрос*

```
SELECT NOW(), VERSION()
```

### Результат

NOW()	VERSION()
2021-07-07 13:34:52	8.0.22

Affected rows: 1

### Пример

Посчитать количество книг, написанных Пушкиным и Лермонтовым, а также количество книг, написанных Агатой Кристи и Дойл Артур Конан.

*Запрос*

```
SELECT
  (SELECT COUNT(title)
   FROM book
   WHERE author IN ("Пушкин А.С.", "Лермонтов М.Ю.")
  ) AS count_1,
  (SELECT COUNT(title)
   FROM book
   WHERE author IN ("Агата Кристи", "Дойл Артур Конан")
  ) AS count_2
```

### Результат

count_1	count_2
10	9

Affected rows: 1

**Текст задания** (чтобы не прокручивать страницу):

Посчитать, сколько экземпляров книг на складе имеют цену меньше 500 рублей, а сколько больше или равно. Столбцы назвать `less_500`, `more_500`.

При решении задания используйте вложенные запросы.

### Результат

less_500	more_500
170	55

Affected rows: 1

## 1.5.8 Несколько табличных выражений в одном запросе

### Задание

Посчитать, сколько экземпляров книг на складе имеют цену меньше 500 рублей, а сколько больше или равно. Столбцы назвать `less_500`, `more_500`.

При решении используйте табличные выражения.

### Структура таблицы:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

В одном запросе может быть несколько табличных выражений. Тогда они записываются через запятую перед основным запросом. Оператор **WITH** указывается только перед первым табличным выражением.

```
WITH табличное_выражение_1 (...)
AS (
    ...
),
табличное_выражение_2 (...)
AS (
    ...
)
SELECT ...
FROM табличное_выражение_1, табличное_выражение_2;
```

### Пример

Посчитать количество книг, написанных Пушкиным и Лермонтовым, а также количество книг, написанных Агатой Кристи и Дойл Артур Конан.

*Запрос:*

```
WITH get_count_1(count_author_1)
AS (
    SELECT COUNT(title)
    FROM book
    WHERE author IN ("Пушкин А.С.", "Лермонтов М.Ю.")
),
get_count_2(count_author_2)
AS (
    SELECT COUNT(title)
    FROM book
    WHERE author IN ("Агата Кристи", "Дойл Артур Конан")
)
```



```
SELECT count_author_1 AS count_1,  
       count_author_2 AS count_2  
FROM get_count_1, get_count_2;
```

### Результат

count_1	count_2
10	9

Affected rows: 1

**Текст задания** (чтобы не прокручивать страницу):

Посчитать, сколько экземпляров книг на складе имеют цену меньше 500 рублей, а сколько больше или равно. Столбцы назвать `less_500`, `more_500`.

При решении задания используйте табличные выражения.

### Результат

less_500	more_500
170	55

Affected rows: 1

## 1.5.9 Вложенные запросы и табличные выражения в одном запросе

**Задание (автор - Елена Голяева)**

Вывести авторов и суммарную стоимость их книг, если хотя бы одна их книга имеет цену выше средней по складу. Средняя цена рассчитывается как простое среднее, с помощью `avg()`. Последний столбец назвать `Стоимость`. Информацию отсортировать по убыванию суммарной стоимости.

В этом задании используется и табличное выражение, и вложенный запрос. Для вычисления средней цены книг используйте табличное выражение, а для отбора авторов, у которых есть хотя бы одна книга по цене выше средней - вложенный запрос.

**Структура таблицы:**

book	
PK	book_id
	title
	author
	price
	amount

**Подсказка**

Шаблон запроса:

```
WITH get_avg_price(avg_price)
AS (
    вычислить среднюю цену
)
SELECT ...
FROM ...
GROUP BY ...
HAVING author IN (SELECT ...
                  FROM book, get_avg_price
                  WHERE цена больше средней
                  )
ORDER BY ...
```

**Результат**

```
+-----+-----+
| author          | Стоимость |
+-----+-----+
| Дойл Артур Конан | 14923.30  |
| Пушкин А.С.     | 12265.36  |
| Достоевский Ф.М. | 11802.03  |
| Есенин С.А.     | 9750.00   |
| Лермонтов М.Ю.   | 9048.28   |
| Булгаков М.А.    | 7000.87   |
+-----+-----+
Affected rows: 6
```

## 1.6 Вложенный запросы и табличные выражения, база данных «Резюме»

В этом уроке предлагается написать SQL запросы для таблицы `resume`, в которой хранится информация о резюме для трудоустройства. Каждое резюме содержит имя соискателя, профессиональную область, в которой он ищет работу, должность, минимальную зарплату, стаж работы в этой должности, а также дата рождения соискателя. Дату рождения, как правило, в резюме не указывают, но здесь она включена, чтобы повторить и изучить новые функции работы с датами.

### 1.6.1 Структура и наполнение таблицы

Все запросы будут формулироваться для таблицы `resume`, которая создана следующим запросом:

```
CREATE TABLE resume (  
    resume_id INT PRIMARY KEY AUTO_INCREMENT,  
    applicant VARCHAR(60), # фамилия и инициалы соискателя  
    specialisation VARCHAR(80), # область профессиональной  
    деятельности  
    position VARCHAR(50), # должность  
    min_salary INT, # минимальная заработная плата  
    experience INT, # опыт работы  
    date_birth DATE # дата рождения  
);
```

Наполнение таблицы `resume`:

resume_id	applicant	specialisation	position	min_salary	experience	date_birth
1	Иванов М.С.	IT	web-программист	50000	2	1998-12-15
2	Петров Ф.С.	Строительство	инженер-строитель	45000	5	1995-1-7
3	Федоров П.Р.	Продажи	менеджер по продажам	35000	7	1993-8-26
4	Абрамова А.А.	Продажи	региональный менеджер	90000	4	1992-9-23
5	Самарин С.С.	Транспорт	логист-диспетчер	30000	0	2001-7-18
6	Туполев И.Д.	Строительство	инженер-строитель	40000	10	1985-3-16
7	Баранов П.В.	Юристы	адвокат	60000	5	1982-2-24
8	Шевцова А.Р.	Продажи	менеджер по продажам	30000	2	1995-11-12
9	Белых К.Д.	IT	Python программист	100000	5	1991-12-9
10	Садиев С.И.	IT	программист PHP и Питон	90000	6	1982-11-7
...						

### Дамп базы данных этого урока:

```
CREATE TABLE resume (
    resume_id INT PRIMARY KEY AUTO_INCREMENT,
    applicant VARCHAR(60),
    specialisation VARCHAR(80),
    position VARCHAR(50),
    min_salary INT,
    experience INT,
    date_birth DATE);

INSERT INTO resume(applicant, specialisation, position,min_salary, experience,
date_birth) VALUES
('Иванов М.С.', 'IT', 'web-программист', 50000, 2, '1998-12-15'),
('Петров Ф.С.', 'Строительство', 'инженер-строитель', 45000, 5, '1995-01-07'),
('Федоров П.Р.', 'Продажи', 'менеджер по продажам', 35000, 7, '1993-08-26'),
('Абрамова А.А.', 'Продажи', 'региональный менеджер', 90000, 4, '1992-09-23'),
('Самарин С.С.', 'Транспорт', 'логист-диспетчер', 30000, 0, '2001-07-18'),
('Туполев И.Д.', 'Строительство', 'инженер-строитель', 40000, 10, '1985-03-16'),
('Баранов П.В.', 'Юристы', 'адвокат', 60000, 5, '1982-02-24'),
('Шевцова А.Р.', 'Продажи', 'менеджер по продажам', 30000, 2, '1995-11-12'),
('Белых К.Д.', 'IT', 'Python программист', 100000, 5, '1991-12-09'),
('Садиев С.И.', 'IT', 'программист PHP и Питон', 90000, 6, '1982-11-07'),
('Астахов И.И.', 'Строительство', 'прораб', 100000, 15, '1976-01-30'),
('Борисов В.В.', 'Продажи', 'менеджер по продажам', 50000, 12, '1980-03-01'),
('Тошаков П.С.', 'Строительство', 'технолог', 50000, 15, '1979-04-25'),
('Солиева К.П.', 'Продажи', 'директор по продажам', 130000, 7, '1973-04-30'),
('Левченко А.Г.', 'IT', 'PHP разработчик', 110000, 9, '1967-12-27'),
('Хованец Г.П.', 'Строительство', 'прораб', 90000, 16, '1970-10-01'),
('Тихомиров Т.М.', 'Транспорт', 'экспедитор', 45000, 20, '1975-08-08'),
('Дворников М.А.', 'Транспорт', 'логист-диспетчер', 35000, 2, '1998-11-10'),
('Жидкова Р.Л.', 'Продажи', 'менеджер по продажам', 30000, 5, '1987-10-20'),
('Бондарев С.С.', 'IT', 'web-программист', 130000, 9, '1987-02-16'),
('Иванов П.А.', 'Продажи', 'директор по продажам', 140000, 11, '1979-04-24'),
('Прохоров А.Д.', 'IT', 'web-программист', 200000, 8, '1985-03-23'),
('Соболев А.И.', 'Транспорт', 'водитель', 35000, 15, '1984-03-30'),
('Ромашко Ф.Ю.', 'Продажи', 'продавец-консультант', 25000, 0, '2002-05-07'),
('Краско С.С.', 'IT', 'программист 1С', 85000, 3, '1997-09-02'),
('Паршина Г.И.', 'IT', 'программист C и Питон', 110000, 9, '1981-08-11'),
('Светлова С.Т.', 'Транспорт', 'водитель', 40000, 10, '1991-09-17'),
('Буянов П.Д.', 'Юристы', 'адвокат', 55000, 15, '1980-06-01'),
('Ульянов М.А.', 'Продажи', 'региональный менеджер', 130000, 11, '1979-06-19'),
('Тупикин К.Л.', 'IT', 'программист 1.7C', 140000, 12, '1981-07-03'),
('Ермоленко Г.С.', 'IT', 'программист C#', 90000, 6, '1993-12-09'),
('Сомов Я.Г.', 'IT', 'программист C++', 115000, 8, '1990-11-16'),
('Орлова Е.Р.', 'Строительство', 'head', 170000, 15, '1972-01-16'),
('Ларин В.П.', 'Строительство', 'инженер-строитель', 70000, 12, '1970-05-15');
```

## 1.6.2 Вложенные запросы и табличные выражения, пример

### Задание

Для каждой профессиональной области вычислить сколько человек в процентном отношении от общего количества человек в таблице **resume** ищет работу в этой области. Вывести профессиональную область и вычисленный процент, округленный до целого. Последний столбец назвать **Популярность**. Информацию отсортировать по убыванию процента.

#### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary,
        experience, date_birth)
```

#### Пояснение

1. Вложенным запросом посчитать количество претендентов в таблице;
2. Процент вычислить по формуле: количество претендентов в профессиональной области разделить на общее количество претендентов и умножить на 100.

#### Результат

specialisation	Популярность
IT	32
Продажи	26
Строительство	21
Транспорт	15
Юристы	6

Affected rows: 5

### 1.6.3 Вложенные запросы и табличные выражения, пример

#### Задание

Вывести информацию о тех соискателях, которые претендуют на минимальную заработную плату меньше самой маленькой из всех средних заработных плат по каждой профессиональной области. Вывести фамилию соискателя, профессиональную область, должность и минимальную заработную плату. Информацию отсортировать сначала по убыванию минимальной заработной платы, затем по фамилии соискателя в алфавитном порядке.

#### Структура таблицы

resume	
PK	resume_id
applicant	
specialisation	
position	
min_salary	
experience	
date_birth	

```
resume(resume_id, applicant, specialisation, position,min_salary,
        experience, date_birth)
```

#### Результат

applicant	specialisation	position	min_salary
Дворников М.А.	Транспорт	логист-диспетчер	35000
Соболев А.И.	Транспорт	водитель	35000
Федоров П.Р.	Продажи	менеджер по продажам	35000
Жидкова Р.Л.	Продажи	менеджер по продажам	30000
Самарин С.С.	Транспорт	логист-диспетчер	30000
Шевцова А.Р.	Продажи	менеджер по продажам	30000
Ромашко Ф.Ю.	Продажи	продавец-консультант	25000

Affected rows: 7

## 1.6.4 Вложенные запросы и табличные выражения, пример

### Задание

Посчитать среднюю минимальную заработную плату и средний опыт работы по каждой профессиональной должности по следующему алгоритму:

- для каждой должности посчитать среднюю минимальную заработную плату и средний опыт работы;
- на основе средних значений по каждой должности вычислить средние показатели по профессиональной области.

Средние значения округлить до целого. Столбцы назвать **Профессия**, **Мин\_зарплата**, **Опыт\_работы**. Информацию отсортировать сначала по убыванию средней минимальной заработной платы, затем по убыванию среднего опыта работы.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position,min_salary,
        experience, date_birth)
```

### Пояснение

Создать вложенный запрос для вычисления средних характеристик по должности (в него включить и профессиональную область), разместить его в разделе **FROM** основного запроса.

### Результат

Профессия	Мин_зарплата	Опыт_работы
IT	107407	7
Строительство	91667	14
Продажи	76563	6
Юристы	57500	10
Транспорт	38333	11

Affected rows: 5

## 1.6.5 Соединение таблиц, оператор UNION

### Задание

Провести анализ возрастного состава соискателей. Для этого посчитать сколько человек относится к каждой возрастной группе:

- моложе 21 года;
- от 21 до 30 лет;
- от 31 до 40 лет;
- старше 40 лет.

Для каждого соискателя посчитать полное количество лет на 7 августа 2021 года (если человек родился 1 августа 2000 года, то ему полных 21 год, если же 10 августа 2000 года, то ему полных 20 лет).

Столбцы назвать **Возрастная\_группа**, **Количество**. В первом столбце вывести название группы из задания (моложе 21 года и пр.)

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position,min_salary,
        experience, date_birth)
```

### Немного теории

Для объединения двух и более SQL запросов используются операторы **UNION**, его синтаксис:

```
SELECT столбец_1_1, столбец_1_2, ...
FROM
```

```
...
```

```
UNION
```

```
SELECT столбец_2_1, столбец_2_2, ...
FROM
```

```
...
```

или

```
SELECT столбец_1_1, столбец_1_2, ...
FROM
```

```
...
```

```
UNION ALL
```

```
SELECT столбец_2_1, столбец_2_2, ...
FROM
```

```
...
```



Важно отметить, что каждый из **SELECT** должен иметь одинаковое количество столбцов и совместимые типы возвращаемых данных. Каждый запрос может включать разделы **WHERE**, **GROUP BY** и пр.

В результате выполнения этой конструкции будет выведена таблица, имена столбцов которой соответствуют именам столбцов в первом запросе. А в таблице результата сначала отображаются записи-результаты первого запроса, а затем второго.

Если указано ключевое слово **ALL**, то в результат включаются все записи запросов, в противном случае - различные.

### Пример

Вывести статистическую информацию по таблице **resume**, которая включает следующие характеристики:

- Количество соискателей;
- Количество должностей;
- Средний опыт работы;
- Средняя минимальная заработная плата.

*Запрос:*

```
SELECT
    "Количество соискателей" AS Характеристика,
    COUNT(*) AS Значение
FROM resume
UNION
SELECT
    "Количество должностей",
    COUNT(DISTINCT position)
FROM resume
UNION
SELECT
    "Средний опыт работы",
    ROUND(AVG(experience))
FROM resume
UNION
SELECT
    "Средняя минимальная заработная плата",
    ROUND(AVG(min_salary))
FROM resume;
```

### Результат

Характеристика	Значение
Количество соискателей	34
Количество должностей	21
Средний опыт работы	9
Средняя минимальная заработная плата	80735

Affected rows: 4

**Текст задания** (чтобы не прокручивать страницу):

Провести анализ возрастного состава соискателей. Для этого посчитать сколько человек относится к каждой возрастной группе:

- моложе 21 года;

- от 21 до 30 лет;
- от 31 до 40 лет;
- старше 40 лет.

Для каждого соискателя посчитать полное количество лет на 7 августа 2021 года (если человек родился 1 августа 2000 года, то ему полных 21 год, если же 10 августа 2000 года, то ему полных 20 лет).

Столбцы назвать **Возрастная\_группа**, **Количество**. В первом столбце вывести название группы из задания (моложе 21 года и пр.)

### Пояснение

Возможный алгоритм решения:

- в табличном выражении посчитать количество полных лет каждого соискателя;
- основной запрос реализовать с использованием оператора **UNION**, каждая часть запроса вычисляет людей в группе в одной группе на основе результата табличного выражения.

### Результат

Возрастная_группа	Количество
моложе 21 года	2
от 21 до 30 лет	11
от 31 до 40 лет	9
старше 40 лет	12

Affected rows: 4

## 1.6.6 Табличные выражения, оператор LIMIT

### Задание

Вывести самых молодых соискателей из таблицы **resume**. Самыми молодыми считать трех соискателей с минимальным количеством полных лет по отношению к 7 августа 2021 года. Если у четырех и более соискателей будет одинаковый возраст - то вывести их всех.

Например, самые молодые соискатели имеют возраст 18, 18, 20, 20, 20 лет. Возраст третьего соискателя 20 лет, но и четвертого, и пятого тоже 20 лет. В этом случае вывести всех пятерых.

В результат включить фамилию и инициалы соискателя, профессиональную область, должность и количество полных лет. Последний столбец назвать **Возраст**. Информацию отсортировать сначала по возрастанию возраста, затем по фамилии в алфавитном порядке.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position, min_salary,
        experience, date_birth)
```

### Немного теории

Для того, чтобы ограничить выборку заданным количеством строк используется оператор **LIMIT**, после которого задается количество строк, которое нужно включить в выборку. Этот оператор указывается после **ORDER BY** и выполняется самым последним.

```
SELECT ...
FROM ...
WHERE ...
GROUP BY ...
HAVING ...
ORDER BY ...
LIMIT число;
```

### Пример

Вывести информацию о пяти соискателях, претендующих на самую высокую заработную плату. Информацию отсортировать сначала по убыванию минимальной заработной платы, затем по фамилиям соискателей в алфавитном порядке.

*Запрос:*

```
SELECT min_salary
```

```
FROM resume
ORDER BY min_salary DECS
LIMIT 5;
```

### Результат

applicant	specialisation	position	min_salary
Прохоров А.Д.	IT	web-программист	200000
Орлова Е.Р.	Строительство	head	170000
Тупикин К.Л.	IT	программист 1.7С	140000
Иванов П.А.	Продажи	директор по продажам	140000
Бондарев С.С.	IT	web-программист	130000

Affected rows: 5

Но обратите внимание, что такое решение является не совсем верным, так как у 6-го и 7-го соискателей заработная плата тоже 130000 рублей. Поэтому либо их тоже нужно включать в выборку, либо добавлять критерии, по которым отбирать соискателей.

Рассмотрим реализацию первого варианта (вывести всех соискателей, если у них минимальная заработная плата больше или равна заработной плате 5-го человека).

**Шаг 1.** Отберем 5 максимальных заработных плат и оформим выборку в виде табличного выражения:

```
WITH get_list_salary(salary)
AS (
    SELECT min_salary
    FROM resume
    ORDER BY min_salary desc
    LIMIT 5
)
SELECT * FROM get_list_salary;
```

### Результат

min_salary
200000
170000
140000
140000
130000

Affected rows: 5

**Шаг 2.** Выберем соискателей, заработная плата которых больше или равна самой маленькой заработной платы, полученной табличным выражением (то есть больше или равна заработной плате 5-го соискателя).

```
WITH get_list_salary(salary)
AS (
    SELECT min_salary
    FROM resume
    ORDER BY min_salary desc
    LIMIT 5
)
SELECT applicant, specialisation, position, min_salary
FROM resume
```

```
WHERE min_salary >= ANY( SELECT *
                           FROM get_list_salary)
ORDER BY 4 DESC, 1
```

### Результат

applicant	specialisation	position	min_salary
Прохоров А.Д.	IT	web-программист	200000
Орлова Е.Р.	Строительство	head	170000
Иванов П.А.	Продажи	директор по продажам	140000
Тупикин К.Л.	IT	программист 1.7С	140000
Бондарев С.С.	IT	web-программист	130000
Солиева К.П.	Продажи	директор по продажам	130000
Ульянов М.А.	Продажи	региональный менеджер	130000

Affected rows: 7

### Текст задания (чтобы не прокручивать страницу):

Вывести самых молодых соискателей из таблицы **resume**. Самыми молодыми считать трех соискателей с минимальным количеством полных лет по отношению к 7 августа 2021 года. Если у четырех и более соискателей будет одинаковый возраст - то вывести их всех.

Например, самые молодые соискатели имеют возраст 18, 18, 20, 20, 20 лет. Возраст третьего соискателя 20 лет, но и четвертого, и пятого тоже 20 лет. В этом случае вывести всех пятерых.

В результат включить фамилию и инициалы соискателя, профессиональную область, должность и количество полных лет. Последний столбец назвать **Возраст**. Информацию отсортировать сначала по возрастанию возраста, затем по фамилии в алфавитном порядке.

### Результат

applicant	specialisation	position	Возраст
Ромашко Ф.Ю.	Продажи	продавец-консультант	19
Самарин С.С.	Транспорт	логист-диспетчер	20
Дворников М.А.	Транспорт	логист-диспетчер	22
Иванов М.С.	IT	web-программист	22

Affected rows: 4

## 1.6.7 Табличные выражения, функции DATE\_ADD(), DATE\_SUB(), DAYOFWEEK()

### Задание

Для соискателей профессиональной области "ИТ" назначить дату собеседования на октябрь 2021 года по следующему алгоритму:

- день собеседования должен совпадать с днем рождения соискателя, например, если человек родился 12 декабря, то дата его собеседования - 12 октября;
- если день собеседования приходится на субботу, то перенести дату собеседования на ближайшую пятницу, если дата собеседования приходится на воскресенье, то перенести его на ближайший понедельник.

Вывести даты собеседований и в одном поле перечислить фамилия и инициалы соискателей в алфавитном порядке через запятую. Столбцы назвать **Дата** и **Соискатели**. Информацию отсортировать по возрастанию даты собеседований.

### Структура таблицы

resume	
PK	resume_id
	applicant
	specialisation
	position
	min_salary
	experience
	date_birth

```
resume(resume_id, applicant, specialisation, position,min_salary,
        experience, date_birth)
```

### Немного теории

В SQL даты можно складывать с числом, а также отнимать от даты число. Для этого используются функции:

```
DATE_ADD(дата, INTERVAL число единица_измерения),
DATE_SUB(дата, INTERVAL число единица_измерения),
```

где

единица\_измерения (использовать прописные буквы) - это день (DAY),  
месяц (MONTH), неделя (WEEK) и пр.,

число - целое число,

дата - значение даты или даты и времени.

Например:

```
DATE_ADD('2021-08-02', INTERVAL 45 DAY) = "2021-09-16"
DATE_ADD('2021-07-02', INTERVAL 6 MONTH) = "2022-01-02"
```

```
DATE_SUB('2021-08-02', INTERVAL 45 DAY) = "2021-06-18"
DATE_SUB('2021-07-02', INTERVAL 6 MONTH) = "2021-01-02"
```

Также можно узнать на какой день недели приходится некоторая дата. Для этого используется функция:

**DAYOFWEEK (дата)** - возвращает 1, если это воскресенье, 2 - понедельник, 3 - вторник и т.д.

Например:

```
DAYOFWEEK("2021-08-08") = 1  
DAYOFWEEK("2021-12-31") = 6
```

### Пример

Для всех родившихся в августе вывести даты, когда им отправить поздравление. Считаем, что поздравление нужно отправить за один день до дня рождения.

Запрос:

```
SELECT  
    applicant,  
    MAKEDATE('2021', DAYOFYEAR(DATE_SUB(date_birth, INTERVAL 1 DAY))) AS  
Дата  
FROM resume  
WHERE MONTH(date_birth) = 8
```

### Результат

```
+-----+  
| applicant | Дата |  
+-----+  
| Федоров П.Р. | 2021-08-25 |  
| Тихомиров Т.М. | 2021-08-07 |  
| Паршина Г.И. | 2021-08-29 |  
+-----+  
Affected rows: 3
```

**Текст задания** (чтобы не прокручивать страницу):

Для соискателей профессиональной области "**IT**" назначить дату собеседования на октябрь 2021 года по следующему алгоритму:

- день собеседования должен совпадать с днем рождения соискателя, например, если человек родился 12 декабря, то дата его собеседования - 12 октября;
- если день собеседования приходится на субботу, то перенести дату собеседования на ближайшую пятницу, если дата собеседования приходится на воскресенье, то перенести его на ближайший понедельник.

Вывести даты собеседований и в одном поле перечислить фамилия и инициалы соискателей в алфавитном порядке через запятую. Столбцы назвать **Дата** и **Соискатели**. Информацию отсортировать по возрастанию даты собеседований.

### Пояснение

Возможный алгоритм решения:

- в первом табличном выражении вычислить дату собеседования соискателей, для этого нужно к 1 октября 2021 года прибавить день рождения соискателя минус один, например, для дня рождения "2000-02-22", это будет "2021-10-01" + (22-1) = 2021-10-22;

- во втором табличном выражении, построенном на основе первого, отнять или прибавить 1 день, если дата собеседования пришлась на субботу или воскресенье;
- основной запрос построить по второму табличному выражению, сгруппировать информацию по дате собеседования и вывести список кандидатов с помощью функции **GROUP\_CONCAT()**.

### Результат

Дата	Соискатели
2021-10-01	Краско С.С.
2021-10-04	Тупикин К.Л.
2021-10-07	Садиев С.И.
2021-10-08	Белых К.Д., Ермоленко Г.С.
2021-10-11	Паршина Г.И.
2021-10-15	Бондарев С.С., Иванов М.С., Сомов Я.Г.
2021-10-22	Прохоров А.Д.
2021-10-27	Левченко А.Г.

Affected rows: 8



## 1.7 Простые запросы на выборку, база данных склад

### 1.7.1 Структура и наполнение базы данных «Склад»

В этом уроке предлагается написать SQL запросы для таблицы `book`, в которой хранится информация о книгах на некотором складе. Все запросы будут формулироваться для таблицы `book`, которая создана следующим запросом:

```
CREATE TABLE book (  
  book_id INT PRIMARY KEY AUTO_INCREMENT,  
  title VARCHAR(50),  
  author VARCHAR(30),  
  price decimal(8,2),  
  amount INT );
```

Наполнение таблицы `book`:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
...				

Дамп базы данных для этого урока:

```
CREATE TABLE book (  
  book_id INT PRIMARY KEY AUTO_INCREMENT,  
  title VARCHAR(50),  
  author VARCHAR(30),  
  price decimal(8,2),  
  amount INT  
);  
  
INSERT INTO book (title, author, price, amount)  
VALUES  
( 'Мастер и Маргарита', 'Булгаков М.А.', 670.99, 3),  
( 'Белая гвардия', 'Булгаков М.А.', 540.50, 5),  
( 'Идиот', 'Достоевский Ф.М.', 460, 10),  
( 'Братья Карамазовы', 'Достоевский Ф.М.', 799.01, 3),  
( 'Игрок', 'Достоевский Ф.М.', 480.50, 10),  
( 'Стихотворения и поэмы', 'Есенин С.А.', 650.00, 15),  
( 'Таинственный остров', 'Жюль Верн', 169.99, 0),  
( 'Пуаро ведет следствие', 'Агата Кристи', 272.50, 2),  
( 'Евгений Онегин', 'Пушкин А.С.', 200.11, 8),  
( 'Бородино', 'Лермонтов М.Ю.', 316.55, 10),  
( 'Дубровский', 'Пушкин А.С.', 114.99, 7),  
( 'Собачье сердце', 'Булгаков М.А.', 380.90, 6),  
( 'Вокруг света за 80 дней', 'Жюль Верн', 201.01, 15),  
( 'Смерть на Ниле', 'Агата Кристи', 250.12, 18),  
( 'Убийства по алфавиту', 'Агата Кристи', 231.70, 9),  
( 'Загадочное происшествие', 'Агата Кристи', 260.99, 15),  
( 'Капитанская дочка', 'Пушкин А.С.', 199.99, 5),  
( 'Этюд в багровых тонах', 'Дойл Артур Конан', 215.00, 3),  
( 'Приключения Шерлока Холмса', 'Дойл Артур Конан', 590.50, 11),  
( 'Записки о Шерлоке Холмсе', 'Дойл Артур Конан', 470.20, 14),  
( 'Затерянный мир', 'Дойл Артур Конан', 400.00, 3),  
( 'Стихи', 'Лермонтов М.Ю.', 550.60, 6),  
( 'Поэмы', 'Лермонтов М.Ю.', 202.40, 8),  
( 'Герой нашего времени', 'Лермонтов М.Ю.', 479.99, 2),  
( 'Стихи', 'Пушкин А.С.', 600.50, 5),  
( 'Слово милой', 'Пушкин А.С.', 120.30, 12),  
( 'Поэмы', 'Пушкин А.С.', 630.50, 7),  
( 'Скрюченный домишко', 'Агата Кристи', 150.01, 13);
```

## 1.7.2 Запросы на обновление, табличные выражения

### Задание

В конце года производится переоценка книг на складе. Цену книг тех авторов, общее количество экземпляров которых максимально, снизить на 15%, а цену книг тех авторов, общее количество экземпляров книг которых минимально, поднять на 10%.

При решении задачи используйте табличные выражения.

### Структура таблиц:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

Табличные выражения можно использовать в запросах **INSERT**, **UPDATE**, **CREATE**, **DELETE**. Синтаксис запроса **UPDATE** в этом случае следующий:

```
WITH табличное_выражение (результат)
AS (
    ...
)
UPDATE ..., табличное выражение
SET ...
WHERE ...;
```

Либо табличные выражения можно использовать во вложенных запросах (в разделах **SET** и **WHERE**):

```
WITH табличное_выражение (результат)
AS (
    ...
)
UPDATE ...
SET ... (SELECT ... FROM табличное_выражение)
WHERE ...;
```

### Пример

Уменьшить на 10 % цену книг тех авторов, средняя арифметическая цена книг которых больше 500 рублей.

### Запрос:

```
WITH get_author_avg(auth, avg_price)
AS (
    select author, AVG(price)
    FROM book
    GROUP BY author
    HAVING AVG(price) > 500
)
UPDATE book
```

```

set price =
CASE
    WHEN author IN (SELECT auth FROM get_author_avg) THEN 0.9 * price
    ELSE price
END;

```

```
SELECT * FROM book;
```

## Результат

Affected rows: 7

Query result:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	603.89	3
2	Белая гвардия	Булгаков М.А.	486.45	5
3	Идиот	Достоевский Ф.М.	414.00	10
4	Братья Карамазовы	Достоевский Ф.М.	719.11	3
5	Игрок	Достоевский Ф.М.	432.45	10
6	Стихотворения и поэмы	Есенин С.А.	585.00	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	342.81	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
16	Загадочное происшествие	Агата Кристи	260.99	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
18	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
21	Затерянный мир	Дойл Артур Конан	400.00	3
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
24	Герой нашего времени	Лермонтов М.Ю.	479.99	2
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	150.01	13

Affected rows: 28

## Текст задания (чтобы не прокручивать страницу):

В конце года производится переоценка книг на складе. Цену книг тех авторов, общее количество экземпляров которых максимально, снизить на 15%, а цену книг тех авторов, общее количество экземпляров книг которых минимально, поднять на 10%.

При решении задачи используйте табличные выражения.

## Результат

Affected rows: 8

Query result:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	738.09	3
2	Белая гвардия	Булгаков М.А.	594.55	5
3	Идиот	Достоевский Ф.М.	460.00	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3

5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	231.63	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	418.99	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	212.60	18
15	Убийства по алфавиту	Агата Кристи	196.95	9
16	Загадочное происшествие	Агата Кристи	221.84	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
18	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
21	Затерянный мир	Дойл Артур Конан	400.00	3
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
24	Герой нашего времени	Лермонтов М.Ю.	479.99	2
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	127.51	13

Affected rows: 28

### 1.7.3 Запрос на обновление, табличные выражения, пример

#### Задание

В конце года производится переоценка книг на складе. Для всех книг в цене будет установлено

количество копеек 99 по следующему правилу:

- цена книги, количество экземпляров которой меньше среднего количество экземпляров книг на складе, сначала округляется в меньшую сторону, а затем к ней прибавляется 0.99;
- для остальных книг ее цена округляется в большую сторону, а затем к ней тоже прибавляется 0.99.

#### Структура таблиц:

book	
PK	book_id
	title
	author
	price
	amount

#### Результат

Affected rows: 23

Query result:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.99	5
3	Идиот	Достоевский Ф.М.	460.99	10
4	Братья Карамазовы	Достоевский Ф.М.	799.99	3
5	Игрок	Достоевский Ф.М.	481.99	10
6	Стихотворения и поэмы	Есенин С.А.	650.99	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.99	2
9	Евгений Онегин	Пушкин А.С.	200.99	8
10	Вородино	Лермонтов М.Ю.	317.99	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	380.99	6
13	Вокруг света за 80 дней	Жюль Верн	202.99	15
14	Смерть на Ниле	Агата Кристи	251.99	18
15	Убийства по алфавиту	Агата Кристи	232.99	9
16	Загадочное происшествие	Агата Кристи	261.99	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
18	Этюд в багровых тонах	Дойл Артур Конан	215.99	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	591.99	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	471.99	14
21	Затерянный мир	Дойл Артур Конан	400.99	3
22	Стихи	Лермонтов М.Ю.	550.99	6
23	Поэмы	Лермонтов М.Ю.	202.99	8
24	Герой нашего времени	Лермонтов М.Ю.	479.99	2
25	Стихи	Пушкин А.С.	600.99	5
26	Слово милой	Пушкин А.С.	121.99	12
27	Поэмы	Пушкин А.С.	630.99	7
28	Скрюченный домишко	Агата Кристи	151.99	13

Affected rows: 28

## 1.7.4 Запрос на удаление, табличные выражения

### Задание

Из таблицы `book` удалить книги, количество экземпляров которых меньше наименьшего среднего количества экземпляров среди средних количеств экземпляров каждого автора.

При решении задачи используйте табличные выражения.

### Структура таблиц:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

Табличные выражения можно использовать в запросах `INSERT`, `UPDATE`, `CREATE`, `DELETE`. Синтаксис запроса `DELETE` в этом случае следующий:

```
WITH табличное_выражение (результат)
AS (
    ...
)
DELETE таблица
FROM таблица, табличное выражение, ...
WHERE ...;
```

**Важно!** Если табличное выражение возвращает несколько значений, в `WHERE` необходимо указать условие их соединения.

Также табличные выражения можно использовать во вложенных запросах в `WHERE`.

### Пример

Из таблицы `book` удалить книги тех авторов, средняя арифметическая цена книг которых больше 500 рублей.

*Запрос:*

```
WITH get_author_avg(auth, avg_price)
AS (
    select author, AVG(price)
    FROM book
    GROUP BY author
    HAVING AVG(price) > 500
)
DELETE book
FROM book
WHERE author IN (SELECT auth FROM get author avg);
SELECT * FROM book;
```

### Результат

Affected rows: 7

Query result:

book_id	title	author	price	amount
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
16	Загадочное происшествие	Агата Кристи	260.99	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
18	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
21	Затерянный мир	Дойл Артур Конан	400.00	3
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
24	Герой нашего времени	Лермонтов М.Ю.	479.99	2
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	150.01	13

Affected rows: 21

**Текст задания** (чтобы не прокручивать страницу):

Из таблицы **book** удалить книги, количество экземпляров которых меньше наименьшего среднего количества экземпляров среди средних количеств экземпляров каждого автора.

**Результат**

Affected rows: 7

Query result:

book_id	title	author	price	amount
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460.00	10
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
16	Загадочное происшествие	Агата Кристи	260.99	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	150.01	13

Affected rows: 21



## 1.7.5 Запрос на удаление, табличные выражения, пример

### Задание

Из таблицы **book** удалить книги, цена которых больше наименьшей средней цены среди средних цен книг каждого автора. Среднюю цену каждого автора посчитать как общую стоимость экземпляров книг деленное на общее количество экземпляров.

### Структура таблиц:

book	
PK	book_id
	title
	author
	price
	amount

### Результат

Affected rows: 21

Query result:

book_id	title	author	price	amount
7	Таинственный остров	Жюль Верн	169.99	0
9	Евгений Онегин	Пушкин А.С.	200.11	8
11	Дубровский	Пушкин А.С.	114.99	7
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
26	Слово милой	Пушкин А.С.	120.30	12
28	Скрюченный домишко	Агата Кристи	150.01	13

Affected rows: 7

## 1.7.6 Запрос на удаление, табличные выражения, пример

**Задание** (автор - Анатолий Алексеев)

Удалить из таблиц **book** книги, цены которых заканчиваются на 99 копеек. Например, книга с ценой 670.99 должна быть удалена.

**Структура таблиц:**

book	
PK	book_id
	title
	author
	price
	amount

### Результат

Affected rows: 6

Query result:

book_id	title	author	price	amount
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460.00	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
12	Собачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
18	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
21	Затерянный мир	Дойл Артур Конан	400.00	3
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	150.01	13

Affected rows: 22

## 1.7.7 Запрос на создание таблицы, табличные выражения

### Задание

Создать таблицу заказ `book_order`, в которую включить те книги из таблицы `book`, количество которых меньше или равно минимальному ненулевому количеству книг в таблице `book`. Название, автора и цену взять из таблицы `book`. Количество указать как среднее количество книг на складе, округленное к ближайшему меньшему целому. Информацию в таблице отсортировать сначала по фамилии автора в алфавитном порядке, а затем по названию книги тоже в алфавитном порядке.

При решении задачи используйте табличные выражения.

### Структура таблиц:

book	
PK	book_id
	title
	author
	price
	amount

### Немного теории

Табличные выражения можно использовать в запросах `INSERT`, `UPDATE`, `CREATE`, `DELETE`. Синтаксис запроса `CREATE` следующий:

```
CREATE TABLE новая_таблица AS
WITH табличное_выражение (результат)
AS (
    ...
)
SELECT ..., результат, ...
FROM ..., табличное выражение, ...
WHERE ...
...;
```

**Важно!** Если табличное выражение возвращает больше одной записи, то в `WHERE` необходимо прописать условие связи табличного выражения и таблиц в `FROM`.

### Пример

Создать таблицу заказ `book_order`, в которую включить все книги из таблицы `book`, у которых количество экземпляров в два раза меньше среднего количества экземпляров книг на складе. В качестве количества указать среднее количества книг на складе, округленное в большую сторону.

### Запрос:

```
CREATE TABLE book_order
WITH get_avg(avg_amount)
AS (
    select CEIL(AVG(amount))
```

```

FROM book
)
SELECT title, author, price, avg_amount
FROM book, get_avg
WHERE amount < avg_amount / 2;

SELECT * FROM book_order;

```

## Результат

Affected rows: 7

Query result:

title	author	price	avg_amount
Мастер и Маргарита	Булгаков М.А.	670.99	9
Братья Карамазовы	Достоевский Ф.М.	799.01	9
Таинственный остров	Жюль Верн	169.99	9
Пуаро ведет следствие	Агата Кристи	272.50	9
Этюд в багровых тонах	Дойл Артур Конан	215.00	9
Затерянный мир	Дойл Артур Конан	400.00	9
Герой нашего времени	Лермонтов М.Ю.	479.99	9

Affected rows: 7

## Текст задания (чтобы не прокручивать страницу):

Создать таблицу заказ **book\_order**, в которую включить те книги из таблицы **book**, количество которых меньше или равно минимальному ненулевому количеству книг в таблице **book**. Название, автора и цену взять из таблицы **book**. Количество указать как среднее количество книг на складе, округленное к ближайшему меньшему целому. Информацию в таблице отсортировать сначала по фамилии автора в алфавитном порядке, а затем по названию книги тоже в алфавитном порядке.

## Пояснение

Сортировку необходимо указать при СОЗДАНИИ таблицы **book\_order**, а не при ее просмотре.

## Результат

Affected rows: 2

Query result:

title	author	price	avg_amount
Пуаро ведет следствие	Агата Кристи	272.50	8
Таинственный остров	Жюль Верн	169.99	8
Герой нашего времени	Лермонтов М.Ю.	479.99	8

Affected rows: 3

### 1.7.8 Создание таблицы

#### Задание

Создайте таблицу **supply** следующей структуры:

Поле	Тип, описание
<b>supply_id</b>	ключевой столбец целого типа с автоматическим увеличением значения ключа на 1
<b>title</b>	строка переменной длины, размер 50
<b>author</b>	строка переменной длины, размер 30
<b>price</b>	вещественное число, максимальная длина 8, количество знаков после запятой 2
<b>amount</b>	целое число

В этой таблице будет храниться информация о книгах, поставляемых на склад.

## 1.7.9 Запросы на добавление, табличные выражения

### Задание

Включить из таблицы **supply** в таблицу **book** "новые" книги тех авторов, книги которых уже есть на складе. То есть автор которых в таблице **book** есть, но название книги этого автора - встречается впервые.

При решении задачи используйте табличные выражения.

### Структура таблиц:

book	
PK	book_id
	title
	author
	price
	amount

supply	
PK	supply_id
	title
	author
	price
	amount

### Немного теории

Табличные выражения можно использовать в запросах **INSERT**, **UPDATE**, **CREATE**, **DELETE**. Синтаксис запроса **INSERT**, если данные добавляются с помощью **SELECT**, следующий:

```
INSERT INTO таблица (поле_1, поле_2, ....)
WITH табличное_выражение (...)
AS (
    ...
)
SELECT ...
FROM табличное выражение, ...
...;
```

### Пример

Включить в таблицу **book** все "новые" книги из таблицы **supply**. То есть либо книги нового автора, либо книги, автор которых в таблице **book** есть, но название книги - встречается впервые.

*Запрос:*

```
INSERT INTO book(title, author, price, amount)
WITH get_new_book(title, author, price, amount)
AS (
    select title, author, price, amount
    FROM supply
    WHERE (author, title) NOT IN (
        select author, title
        FROM book
    )
)
SELECT title, author, price, amount
FROM get_new_book;
```

```
SELECT * FROM book;
```

## Результат

Affected rows: 6

Query result:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460.00	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15
7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
16	Загадочное происшествие	Агата Кристи	260.99	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
18	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
21	Затерянный мир	Дойл Артур Конан	400.00	3
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
24	Герой нашего времени	Лермонтов М.Ю.	479.99	2
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	150.01	13
29	Лирика	Пастернак Б.Л.	518.99	2
30	Доктор Живаго	Пастернак Б.Л.	680.09	5
31	Черный человек	Есенин С.А.	570.20	6
32	Стихи	Есенин С.А.	520.50	5
33	Война и мир	Толстой Л.Н.	799.99	6
34	Анна Каренина	Толстой Л.Н.	570.00	5

Affected rows: 34

**Текст задания** (чтобы не прокручивать страницу):

Включить из таблицы **supply** в таблицу **book** "новые" книги тех авторов, книги которых уже есть на складе. То есть автор которых в таблице **book** есть, но название книги этого автора - встречается впервые.

## Результат

Affected rows: 2

Query result:

book_id	title	author	price	amount
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460.00	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	3
5	Игрок	Достоевский Ф.М.	480.50	10
6	Стихотворения и поэмы	Есенин С.А.	650.00	15

7	Таинственный остров	Жюль Верн	169.99	0
8	Пуаро ведет следствие	Агата Кристи	272.50	2
9	Евгений Онегин	Пушкин А.С.	200.11	8
10	Бородино	Лермонтов М.Ю.	316.55	10
11	Дубровский	Пушкин А.С.	114.99	7
12	Собаачье сердце	Булгаков М.А.	380.90	6
13	Вокруг света за 80 дней	Жюль Верн	201.01	15
14	Смерть на Ниле	Агата Кристи	250.12	18
15	Убийства по алфавиту	Агата Кристи	231.70	9
16	Загадочное происшествие	Агата Кристи	260.99	15
17	Капитанская дочка	Пушкин А.С.	199.99	5
18	Этюд в багровых тонах	Дойл Артур Конан	215.00	3
19	Приключения Шерлока Холмса	Дойл Артур Конан	590.50	11
20	Записки о Шерлоке Холмсе	Дойл Артур Конан	470.20	14
21	Затерянный мир	Дойл Артур Конан	400.00	3
22	Стихи	Лермонтов М.Ю.	550.60	6
23	Поэмы	Лермонтов М.Ю.	202.40	8
24	Герой нашего времени	Лермонтов М.Ю.	479.99	2
25	Стихи	Пушкин А.С.	600.50	5
26	Слово милой	Пушкин А.С.	120.30	12
27	Поэмы	Пушкин А.С.	630.50	7
28	Скрюченный домишко	Агата Кристи	150.01	13
29	Черный человек	Есенин С.А.	570.20	6
30	Стихи	Есенин С.А.	520.50	5

Affected rows: 30

## Наполнение таблицы `supply`

supply_id	title	author	price	amount
1	Лирика	Пастернак Б.Л.	518.99	2
2	Доктор Живаго	Пастернак Б.Л.	680.09	5
3	Черный человек	Есенин С.А.	570.20	6
4	Белая гвардия	Булгаков М.А.	540.50	7
5	Герой нашего времени	Лермонтов М.Ю.	501.00	8
6	Стихи	Есенин С.А.	520.50	5
7	Таинственный остров	Жюль Верн	169.99	5
8	Пуаро ведет следствие	Агата Кристи	272.50	8
9	Стихи	Пушкин А.С.	600.50	3
10	Идиот	Достоевский Ф.М.	360.80	3
11	Война и мир	Толстой Л.Н.	799.99	6
12	Анна Каренина	Толстой Л.Н.	570.00	5



## 1.7.10 Запрос на создание таблицы, пример

**Задание** (автор - [Артём Чепк](#))

Создать новую таблицу **store** на основе запроса. В эту таблицу занести одинаковые книги из таблиц **book** и **supply** (совпадают название и автор), при условии, что общее количество экземпляров книги будет больше среднего количества экземпляров книг по двум таблицам. Цену выбрать большую из двух, количество просуммировать. Столбцы назвать **Книга**, **Автор**, **Цена**, **Количество**. Данные в таблице должны быть отсортированы сначала по автору в алфавитном порядке, а затем по убыванию цены.

Среднее количество книг по двум таблицам посчитать как сумму всех экземпляров книг в таблицах **book** и **supply**, деленное на сумму количеств уникальных книг в таблицах **book** и **supply**.

**Структура таблиц:**

book	
PK	book_id
	title
	author
	price
	amount

supply	
PK	supply_id
	title
	author
	price
	amount

**Результат**

Affected rows: 5

Query result:

Книга	Автор	Цена	Количество
Пуаро ведет следствие	Агата Кристи	272.50	10
Белая гвардия	Булгаков М.А.	540.50	12
Идиот	Достоевский Ф.М.	460.00	13
Герой нашего времени	Лермонтов М.Ю.	501.00	10
Стихи	Пушкин А.С.	600.50	8

Affected rows: 5