**Структура и наполнение таблицы book:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **book\_id** | **title** | **author** | **price** | **amount** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **VARCHAR(50)** | **VARCHAR(30)** | **DECIMAL(8,2)** | **INT** |
| 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 |

**Задание**

В конце года цену всех книг на складе пересчитывают – снижают ее на 30%. Написать SQL запрос, который из таблицы **book** выбирает названия, авторов, количества и вычисляет новые цены книг. Столбец с новой ценой назвать **new\_price**, цену округлить до 2-х знаков после запятой.

SELECT title, author, amount,

ROUND(price\*0.7, 2) AS new\_price

FROM book;

## Задание

При анализе продаж книг выяснилось, что наибольшей популярностью пользуются книги Михаила Булгакова, на втором месте книги Сергея Есенина. Исходя из этого решили поднять цену книг Булгакова на 10%, а цену книг Есенина - на 5%. Написать запрос, куда включить автора, название книги и новую цену, последний столбец назвать new\_price. Значение округлить до двух знаков после запятой.

SELECT author, title,

ROUND(IF(author = "Булгаков М.А.", price \* 1.1, IF(author = "Есенин С.А.", price \* 1.05, price)), 2) AS new\_price

FROM book;

## Задание

Вывести автора, название и цены тех книг, количество которых меньше 10.

SELECT author, title, price

FROM book

WHERE amount < 10;

**Задание**

Вывести название, автора, цену и количество всех книг, цена которых меньше 500 или больше 600, а стоимость всех экземпляров этих книг больше или равна 5000.

SELECT title, author, price, amount

FROM book

WHERE (price < 500 OR price > 600) AND price\*amount>=5000;

## Задание

Вывести название и авторов тех книг, цены которых принадлежат интервалу от 540.50 до 800 (включая границы), а количество или 2, или 3, или 5, или 7.

SELECT title, author

FROM book

WHERE (price >= 540.50 AND price <=800) AND (amount IN(2, 3, 5, 7));

**Задание**

Вывести  автора и название  книг, количество которых принадлежит интервалу от 2 до 14 (включая границы). Информацию  отсортировать сначала по авторам (в обратном алфавитном порядке), а затем по названиям книг (по алфавиту).

SELECT author, title

FROM book

WHERE amount >= 2 AND amount <=14

ORDER BY 1 DESC, 2 ASC;

## Задание

Вывести название и автора тех книг, название которых состоит из двух и более слов, а инициалы автора содержат букву «С». Считать, что в названии слова отделяются друг от друга пробелами и не содержат знаков препинания, между фамилией автора и инициалами обязателен пробел, инициалы записываются без пробела в формате: буква, точка, буква, точка. Информацию отсортировать по названию книги в алфавитном порядке.

SELECT title, author FROM book

WHERE title LIKE "\_% \_%" and

(author like '% С.\_.%' or author like '% \_.С.%')

ORDER BY 1 ASC, 2 DESC;

## Задание

Отобрать различные (уникальные) элементы столбца amount таблицы book.

SELECT DISTINCT amount

FROM book;

## Задание

Посчитать, количество различных книг и количество экземпляров книг каждого автора хранящихся на складе.  Столбцы назвать **Автор**, **Различных\_книг** и **Количество\_экземпляров** соответственно.

SELECT author AS Автор, COUNT(author) AS Различных\_книг, SUM(amount) AS Количество\_экземпляров

FROM book

GROUP BY author;

## Задание

Вывести фамилию и инициалы автора, минимальную, максимальную и среднюю цену книг каждого автора. Вычисляемые столбцы назвать **Минимальная\_цена, Максимальная\_цена** и**Средняя\_цена** соответственно.

SELECT author, MIN(price) AS Минимальная\_цена, MAX(price) AS Максимальная\_цена,

AVG(price) AS Средняя\_цена

FROM book

GROUP BY author;

## Задание

Для каждого автора вычислить суммарную стоимость книг **S** (имя столбца **Стоимость**), а также вычислить налог на добавленную стоимость  для полученных сумм (имя столбца **НДС** ) , который **включен в стоимость** и составляет k = 18%,  а также стоимость книг  (**Стоимость\_без\_НДС**) без него. Значения округлить до двух знаков после запятой. В запросе для расчета НДС(**tax**)  и Стоимости без НДС(**S\_without\_tax**) использовать следующие формулы:

tax= {{S \*{ k \over 100}} \over {1+{k\over 100}}},

*tax*=1+100*k*​*S*∗100*k*​​,

S\\_without\\_tax= {{S} \over {1+{k\over 100}}}*S*\_*without*\_*tax*=1+100*k*​*S*​

SELECT author,

SUM(price \* amount) AS Стоимость,

ROUND((SUM(price \* amount)\*18/100)/1.18,2) AS НДС,

ROUND(SUM(price \* amount)/1.18,2) AS Стоимость\_без\_НДС

FROM book

GROUP BY author;

## Задание

Вывести  цену самой дешевой книги, цену самой дорогой и среднюю цену уникальных книг на складе. Названия столбцов **Минимальная\_цена, Максимальная\_цена, Средняя\_цена** соответственно. Среднюю цену округлить до двух знаков после запятой.

**Пояснение**. В задании нужно посчитать среднюю цену уникальных книг на складе, а не среднюю цену всех экземпляров книг.

SELECT MIN(price) AS Минимальная\_цена, MAX(price) AS Максимальная\_цена,

ROUND(AVG(price), 2) AS Средняя\_цена FROM book;

## Задание

Вычислить среднюю цену и суммарную стоимость тех книг, количество экземпляров которых принадлежит интервалу от 5 до 14, включительно. Столбцы назвать **Средняя\_цена** и **Стоимость**, значения округлить до 2-х знаков после запятой.

SELECT ROUND (AVG(price), 2) AS Средняя\_цена, SUM(price\*amount) AS Стоимость

FROM book WHERE amount >=5 AND amount<=14;

## ****Задание****

Посчитать стоимость всех экземпляров каждого автора без учета книг «Идиот» и «Белая гвардия». В результат включить только тех авторов, у которых суммарная стоимость книг (без учета книг «Идиот» и «Белая гвардия») более 5000 руб. Вычисляемый столбец назвать **Стоимость**. Результат отсортировать по убыванию стоимости.

SELECT author, SUM(price\*amount) AS Стоимость

FROM book

WHERE title <> 'Идиот' AND title <> 'Белая гвардия'

GROUP BY author

HAVING SUM(price\*amount) > 5000 ORDER BY SUM(price\*amount) DESC;

## Задание

Вывести информацию (автора, название и цену) о  книгах, цены которых меньше или равны средней цене книг на складе. Информацию вывести в отсортированном по убыванию цены виде. Среднее вычислить как среднее по цене книги.

SELECT author, title, price FROM book

WHERE price<=(SELECT AVG(price) FROM book)

ORDER BY price DESC;

## Задание

Вывести информацию (автора, название и цену) о тех книгах, цены которых превышают минимальную цену книги на складе не более чем на 150 рублей в отсортированном по возрастанию цены виде.

SELECT author, title, price

FROM book

WHERE (price - (SELECT MIN(price) FROM book)) <=150

ORDER BY price ASC;

## ****Задание****

Вывести информацию (автора, книгу и количество) о тех книгах, количество экземпляров которых в таблице **book** не дублируется.

SELECT author, title, amount FROM book

WHERE amount IN (

SELECT amount FROM book

GROUP BY amount

HAVING COUNT(amount) = 1);

## Задание

Вывести информацию о книгах(автор, название, цена), цена которых меньше самой большой из минимальных цен, вычисленных для каждого автора.

SELECT author, title, price

FROM book

WHERE price < ANY (

SELECT MIN(price)

FROM book

GROUP BY author

);

## Задание

Посчитать сколько и каких экземпляров книг нужно заказать поставщикам, чтобы на складе стало одинаковое количество экземпляров каждой книги, равное значению самого большего количества экземпляров одной книги на складе. Вывести название книги, ее автора, текущее количество экземпляров на складе и количество заказываемых экземпляров книг. Последнему столбцу присвоить имя **Заказ**. В результат не включать книги, которые заказывать не нужно.

SELECT title, author, amount,

ABS(amount - (SELECT MAX(amount) FROM book)) AS Заказ

FROM book

WHERE ABS(amount - (SELECT MAX(amount) FROM book))>0;

**Задание**

Добавить из таблицы **supply** в таблицу **book**, все книги, кроме книг, написанных Булгаковым М.А. и Достоевским Ф.М.

INSERT INTO book (title, author, price, amount)

SELECT title, author, price, amount

FROM supply

WHERE author NOT IN ('Булгаков М.А.', 'Достоевский Ф.М.');

SELECT \* FROM book;

## Задание

Занести из таблицы **supply** в таблицу **book** только те книги, авторов которых нет в  **book.**

INSERT INTO book (title, author, price, amount)

SELECT title, author, price, amount

FROM supply

WHERE author NOT IN (

SELECT author

FROM book

);

SELECT \* FROM book;

## Задание

Уменьшить на 10% цену тех книг в таблице **book**, количество которых принадлежит интервалу от 5 до 10, включая границы.

UPDATE book

SET price = 0.9 \* price

WHERE amount >= 5 AND amount <= 10;

SELECT \* FROM book;

## Задание

Для тех книг в таблице **book** , которые есть в таблице **supply,**не только увеличить их количество в таблице **book** ( увеличить их количество на значение столбца **amount**таблицы **supply**), но и пересчитать их цену (для каждой книги найти сумму цен из таблиц **book** и **supply** и разделить на 2).

UPDATE book, supply

SET book.amount = book.amount + supply.amount, book.price = ((book.price + supply.price)/2)

WHERE book.title = supply.title AND book.author = supply.author;

SELECT \* FROM book;

## ****Задание****

Удалить из таблицы **supply** книги тех авторов, общее количество экземпляров книг которых в таблице **book** превышает 10.

DELETE FROM supply

WHERE author IN(SELECT author

FROM book

GROUP BY author

HAVING SUM(amount)>10);

SELECT \* FROM supply;

## Задание

Создать таблицу заказ (**ordering**), куда включить авторов и названия тех книг, количество экземпляров которых в таблице **book** меньше среднего количества экземпляров книг в таблице **book**. В таблицу включить столбец   **amount,** в котором для всех книг указать одинаковое значение - среднее количество экземпляров книг в таблице **book**.

CREATE TABLE ordering AS

SELECT author, title,

(

SELECT ROUND(AVG(amount))

FROM book

) AS amount

FROM book

WHERE amount < (SELECT ROUND(AVG(amount)) FROM book);

SELECT \* FROM ordering;

## Структура и наполнение таблицы

Таблица **trip**, в которой представлена информация о командировках сотрудников некоторой организации (фамилия сотрудника, город, куда он ездил, размер суточных, даты первого и последнего дня командировки) :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **trip\_id** | **name** | **city** | **per\_diem** | **date\_first** | **date\_last** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **VARCHAR(30)** | **VARCHAR(25)** | **DECIMAL(8,2)** | **DATЕ** | **DATE** |
| 1 | Баранов П.Е. | Москва | 700 | 2020-01-12 | 2020-01-17 |
| 2 | Абрамова К.А. | Владивосток | 450 | 2020-01-14 | 2020-01-27 |
| 3 | Семенов И.В. | Москва | 700 | 2020-01-23 | 2020-01-31 |
| 4 | Ильиных Г.Р. | Владивосток | 450 | 2020-01-12 | 2020-02-02 |
| 5 | Колесов С.П. | Москва | 700 | 2020-02-01 | 2020-02-06 |
| 6 | Баранов П.Е. | Москва | 700 | 2020-02-14 | 2020-02-22 |
| 7 | Абрамова К.А. | Москва | 700 | 2020-02-23 | 2020-03-01 |
| 8 | Лебедев Т.К. | Москва | 700 | 2020-03-03 | 2020-03-06 |
| 9 | Колесов С.П. | Новосибирск | 450 | 2020-02-27 | 2020-03-12 |
| 10 | Семенов И.В. | Санкт-Петербург | 700 | 2020-03-29 | 2020-04-05 |
| 11 | Абрамова К.А. | Москва | 700 | 2020-04-06 | 2020-04-14 |
| 12 | Баранов П.Е. | Новосибирск | 450 | 2020-04-18 | 2020-05-04 |
| 13 | Лебедев Т.К. | Томск | 450 | 2020-05-20 | 2020-05-31 |
| 14 | Семенов И.В. | Санкт-Петербург | 700 | 2020-06-01 | 2020-06-03 |
| 15 | Абрамова К.А. | Санкт-Петербург | 700 | 2020-05-28 | 2020-06-04 |
| 16 | Федорова А.Ю. | Новосибирск | 450 | 2020-05-25 | 2020-06-04 |
| 17 | Колесов С.П. | Новосибирск | 450 | 2020-06-03 | 2020-06-12 |
| 18 | Федорова А.Ю. | Томск | 450 | 2020-06-20 | 2020-06-26 |
| 19 | Абрамова К.А. | Владивосток | 450 | 2020-07-02 | 2020-07-13 |
| 20 | Баранов П.Е. | Воронеж | 450 | 2020-07-19 | 2020-07-25 |

## Задание

Вывести из таблицы **trip** информацию о командировках тех сотрудников, фамилия которых заканчивается на букву «а», в отсортированном по убыванию даты последнего дня командировки виде. В результат включить столбцы **name, city, per\_diem, date\_first, date\_last.**

SELECT name, city, per\_diem, date\_first, date\_last

FROM trip

WHERE name LIKE "%%а %.%."

ORDER BY date\_last DESC;

## Задание

Вывести в алфавитном порядке фамилии и инициалы тех сотрудников, которые были в командировке в Москве.

SELECT DISTINCT name FROM trip

WHERE city='Москва'

ORDER BY name ASC;

## Задание

Для каждого города посчитать, сколько раз сотрудники в нем были.  Информацию вывести в отсортированном в алфавитном порядке по названию городов. Вычисляемый столбец назвать **Количество**.

SELECT city, COUNT(city) AS 'Количество'

FROM trip

GROUP BY city

ORDER BY city;

## Задание

Вывести два города, в которых чаще всего были в командировках сотрудники. Вычисляемый столбец назвать **Количество**.

SELECT city, COUNT(city) AS 'Количество'

FROM trip

GROUP BY city

ORDER BY COUNT(city)DESC LIMIT 2;

## Задание

Вывести информацию о командировках во все города кроме Москвы и Санкт-Петербурга (фамилии и инициалы сотрудников, город ,  длительность командировки в днях, при этом первый и последний день относится к периоду командировки). Последний столбец назвать **Длительность**. Информацию вывести в упорядоченном по убыванию длительности поездки, а потом по убыванию названий городов (в обратном алфавитном порядке).

SELECT name, city, (DATEDIFF(date\_last, date\_first)+1) AS Длительность

FROM trip

WHERE city <> 'Москва' AND city <>'Санкт-Петербург'

ORDER BY (DATEDIFF(date\_last, date\_first)+1) DESC, city DESC;

## Задание

Вывести информацию о командировках сотрудника(ов), которые были самыми короткими по времени. В результат включить столбцы **name**, **city**, **date\_first**, **date\_last**

Используйте вложенный запрос, чтобы найти длительность самой короткой командировки

SELECT name, city, date\_first, date\_last

FROM trip

WHERE DATEDIFF(date\_last, date\_first) IN

(SELECT MIN(DATEDIFF(date\_last, date\_first))

FROM trip);

## Задание

Вывести информацию о командировках, начало и конец которых относятся к одному месяцу (год может быть любой). В результат включить столбцы **name**, **city**, **date\_first**, **date\_last**. Строки отсортировать сначала  в алфавитном порядке по названию города, а затем по фамилии сотрудника .

SELECT name, city, date\_first, date\_last

FROM trip

WHERE MONTH(date\_first)=MONTH(date\_last)

ORDER BY city, name;

## Задание

Вывести название месяца и количество командировок для каждого месяца. Считаем, что командировка относится к некоторому месяцу, если она началась в этом месяце. Информацию вывести сначала в отсортированном по убыванию количества, а потом в алфавитном порядке по названию месяца виде. Название столбцов – **Месяц** и **Количество**.

SELECT MONTHNAME(date\_first) AS 'Месяц', COUNT(MONTHNAME(date\_first)) AS 'Количество'

FROM trip

GROUP BY MONTHNAME(date\_first)

ORDER BY COUNT(MONTHNAME(date\_first)) DESC, MONTHNAME(date\_first);

## Задание

Вывести сумму суточных (произведение количества дней командировки и размера суточных) для командировок, первый день которых пришелся на февраль или март 2020 года. Значение суточных для каждой командировки занесено в столбец **per\_diem**. Вывести фамилию и инициалы сотрудника, город, первый день командировки и сумму суточных. Последний столбец назвать **Сумма**. Информацию отсортировать сначала  в алфавитном порядке по фамилиям сотрудников, а затем по убыванию суммы суточных.

SELECT name, city, date\_first, (DATEDIFF(date\_last, date\_first)+1)\*per\_diem as Сумма

FROM trip

WHERE YEAR(date\_first)=2020 and MONTH(date\_first)=3 OR MONTH(date\_first)=2

ORDER BY name,((DATEDIFF(date\_last, date\_first)+1)\*per\_diem)DESC;

**Задание**

Вывести фамилию с инициалами и общую сумму суточных, полученных за все командировки для тех сотрудников, которые были в командировках больше чем 3 раза, в отсортированном по убыванию сумм суточных виде. Последний столбец назвать **Сумма**.

Только для этого задания изменена строка таблицы **trip**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | Ильиных Г.Р. | Владивосток | 450 | 2020-01-12 | 2020-03-02 |

SELECT name, SUM((DATEDIFF(date\_last,date\_first)+1)\*per\_diem) AS Сумма

FROM trip

GROUP BY name

HAVING COUNT(name) > 3

ORDER BY Сумма DESC;

## Структура и наполнение таблиц

В таблице **fine** представлена информация о начисленных водителям штрафах за нарушения правил дорожного движения (ПДД) (фамилия водителя, номер машины, описание нарушения, сумма штрафа, дата совершения нарушения и дата оплаты штрафа):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **fine\_id** | **name** | **number\_plate** | **violation** | **sum\_fine** | **date\_violation** | **date\_payment** |
| **INT PRIMARY KEY AUTO\_INCREMENT** | **VARCHAR(30)** | **VARCHAR(6)** | **VARCHAR(50)** | **DECIMAL(8,2)** | **DATЕ** | **DATE** |
| 1 | Баранов П.Е. | Р523ВТ | Превышение скорости (от 40 до 60) | 500.00 | 2020-01-12 | 2020-01-17 |
| 2 | Абрамова К.А. | О111АВ | Проезд на запрещающий сигнал | 1000.00 | 2020-01-14 | 2020-02-27 |
| 3 | Яковлев Г.Р. | Т330ТТ | Превышение скорости (от 20 до 40) | 500.00 | 2020-01-23 | 2020-02-23 |
| 4 | Яковлев Г.Р. | М701АА | Превышение скорости (от 20 до 40) |  | 2020-01-12 |  |
| 5 | Колесов С.П. | К892АХ | Превышение скорости (от 20 до 40) |  | 2020-02-01 |  |
| 6 | Баранов П.Е. | Р523ВТ | Превышение скорости (от 40 до 60) |  | 2020-02-14 |  |
| 7 | Абрамова К.А. | О111АВ | Проезд на запрещающий сигнал |  | 2020-02-23 |  |
| 8 | Яковлев Г.Р. | Т330ТТ | Проезд на запрещающий сигнал |  | 2020-03-03 |  |

CREATE TABLE fine (

fine\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(30),

number\_plate VARCHAR(6),

violation VARCHAR(50),

sum\_fine DECIMAL(8, 2),

date\_violation DATE,

date\_payment DATE);

INSERT INTO fine (name, number\_plate, violation, sum\_fine, date\_violation, date\_payment)

VALUES ('Баранов П.Е.', 'Р523ВТ', 'Превышение скорости(от 40 до 60)', null, '2020-02-14', null);

INSERT INTO fine (name, number\_plate, violation, sum\_fine, date\_violation, date\_payment)

VALUES ('Абрамова К.А.', 'О111АВ', 'Проезд на запрещающий сигнал', null, '2020-02-23', null);

INSERT INTO fine (name, number\_plate, violation, sum\_fine, date\_violation, date\_payment)

VALUES ('Яковлев Г.Р.', 'Т330ТТ', 'Проезд на запрещающий сигнал', null, '2020-03-03', null);

SELECT \* FROM fine;

## Задание

Занести в таблицу **fine** суммы штрафов, которые должен оплатить водитель, в соответствии с данными из таблицы **traffic\_violation**. При этом суммы заносить только в пустые поля столбца **sum\_fine**.

Таблица **traffic\_violation**создана и заполнена.

**Важно!** Сравнение значения столбца с пустым значением осуществляется с помощью оператора **IS NULL**.

UPDATE fine AS f, traffic\_violation AS tv

SET f.sum\_fine=tv.sum\_fine

WHERE (f.sum\_fine IS NULL) AND f.violation=tv.violation;

SELECT \* FROM fine;

## Задание

Вывести фамилию, номер машины и нарушение только для тех водителей, которые на одной машине нарушили одно и то же правило   два и более раз. При этом учитывать все нарушения, независимо от того оплачены они или нет. Информацию отсортировать в алфавитном порядке, сначала по фамилии водителя, потом по номеру машины и, наконец, по нарушению.

SELECT name, number\_plate, violation

FROM fine

GROUP BY name, number\_plate, violation

HAVING COUNT(violation)>=2

ORDER BY name, number\_plate, violation DESC;

## Задание

В таблице **fine** увеличить в два раза сумму неоплаченных штрафов для отобранных на предыдущем шаге записей.

UPDATE fine AS f,

(SELECT name, number\_plate, violation

FROM fine

GROUP BY name, number\_plate, violation

HAVING COUNT(violation)>1) AS query\_in

SET f.sum\_fine = f.sum\_fine\*2

WHERE

f.date\_payment IS NULL AND

f.name = query\_in.name AND

f.number\_plate = query\_in.number\_plate AND

f.violation = query\_in.violation;

SELECT \* FROM fine;

## Задание

Водители оплачивают свои штрафы. В таблице **payment**занесены даты их оплаты:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **payment\_id** | **name** | **number\_plate** | **violation** | **date\_violation** | **date\_payment** |
| 1 | Яковлев Г.Р. | М701АА | Превышение скорости (от 20 до 40) | 2020-01-12 | 2020-01-22 |
| 2 | Баранов П.Е. | Р523ВТ | Превышение скорости (от 40 до 60) | 2020-02-14 | 2020-03-06 |
| 3 | Яковлев Г.Р. | Т330ТТ | Проезд на запрещающий сигнал | 2020-03-03 | 2020-03-23 |

Необходимо:

* в таблицу **fine**занести дату оплаты соответствующего штрафа из таблицы **payment;**
* уменьшить начисленный штраф в таблице **fine** в два раза  (только для тех штрафов, информация о которых занесена в таблицу **payment**) , если оплата произведена не позднее 20 дней со дня нарушения.

UPDATE fine f, payment p

SET f.date\_payment = p.date\_payment,

f.sum\_fine = IF(DATEDIFF(f.date\_payment, f.date\_violation) <= 20, f.sum\_fine / 2, f.sum\_fine)

WHERE f.name = p.name AND

f.number\_plate = p.number\_plate AND

f.violation = p.violation AND

f.date\_violation = p.date\_violation AND

f.date\_payment IS NULL;

SELECT name, violation, sum\_fine, date\_violation, date\_payment

FROM fine;

## Задание

Создать новую таблицу **back\_payment**, куда внести информацию о неоплаченных штрафах (Фамилию и инициалы водителя, номер машины, нарушение, сумму штрафа  и  дату нарушения) из таблицы **fine**.

Для неоплаченных штрафов столбец **date\_payment** имеет пустое значение.

CREATE TABLE back\_payment AS

SELECT name, number\_plate, violation, sum\_fine, date\_violation

FROM fine

WHERE date\_payment IS NULL;

SELECT \* FROM back\_payment;

## Задание

Удалить из таблицы **fine** информацию о нарушениях, совершенных раньше 1 февраля 2020 года.

DELETE FROM fine

WHERE date\_violation < DATE(20200201);

SELECT \* FROM fine;