**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Базы данных»**

Тема: Реализация базы данных в СУБД MySQL.

|  |  |  |
| --- | --- | --- |
| Студентка гр. 9383 |  | Лапина А.А. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2021

## Цель работы.

Изучение проектирования ER моделей и структур БД по текстовому описанию предметной области.

## Задание.

**Задание. (8 вариант)**

* Нарисовать ER модель, рекомендуется использовать draw.io или иной редактор
* Нарисовать структуру БД, содержащую названия полей, таблиц, связи, типы данных, ключи.
* Проверить и обосновать, что реляционная модель соответвует НФБК
* Прикрепить 2 изображения (er.png, db.png) в PR
* Описать полученные модели, для чего нужна каждая сущность, почему такие связи и т.п.
* В отчете описать цель, текст задания в соответствии с вариантом, 2 изображения моделей, их описание, обоснование НФБК, ссылку на PR в приложении, вывод

Описание предметной области:

Пусть требуется создать программную систему, предназначенную для директора продовольственного магазина. Такая система должна обеспечивать хранение сведений о магазине, об имеющихся в нем товарах, о торговых базах и товарах, хранящихся на этих базах. Магазин осуществляет закупку товаров на разных базах, предпочитая при этом закупать одни виды товара на одних базах, а другие на других. Магазин характеризуется классом, номером и имеет несколько отделов. Каждый товар в каждом магазине продается, по крайней мере, в одном отделе. Каждый отдел имеет заведующего. Товары, имеющиеся в магазине и хранящиеся на базах, характеризуются ценой, сортом и количеством. Розничные цены в магазине зависят от класса магазина. Директор магазина должен иметь возможность изменить цену товара по своему усмотрению, осуществить закупку недостающего товара на базе. Он может также закрыть один из отделов или открыть новый, при этом товары могут перемещаться из отдела в отдел. Директору могут потребоваться следующие сведения:

* Какие товары имеются в магазине (на базе)?
* Какие отсутствующие товары может заказать магазин на базе?
* Какие товары, и в каком количестве имеются в отделе магазина?
* Список заведующих отделами магазина?
* Суммарная стоимость товара в каждом отделе?
* На каких базах, и в каких количествах есть товар нужного наименования?

## Выполнение работы.

В ходе выполнения лабораторной работы была развёрнута локально MySQL и добавлена база данных, в которой было создано 8 таблиц, представленные на рисунках 1–8.

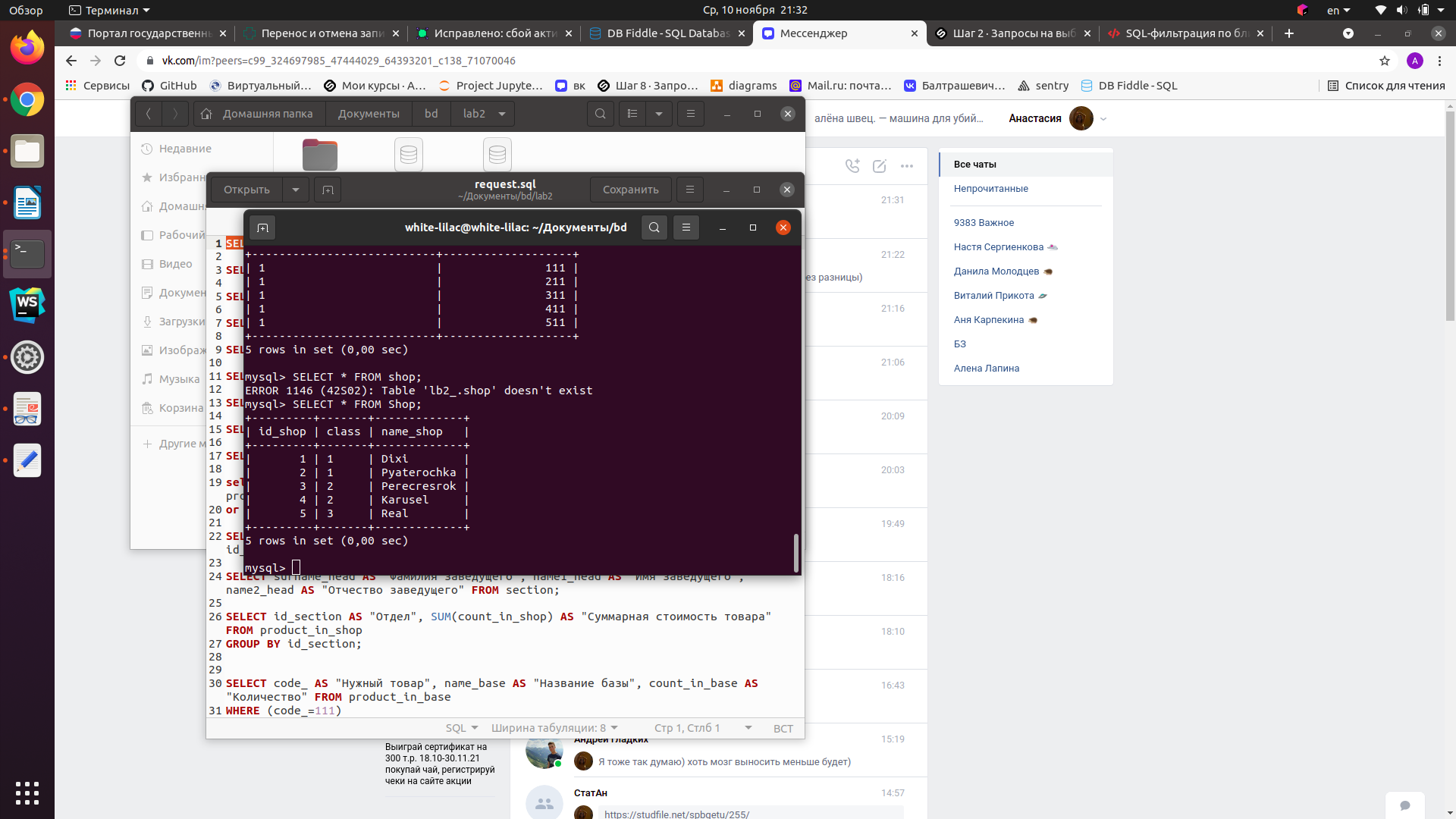


Рисунок 1 – Таблица “Магазин”

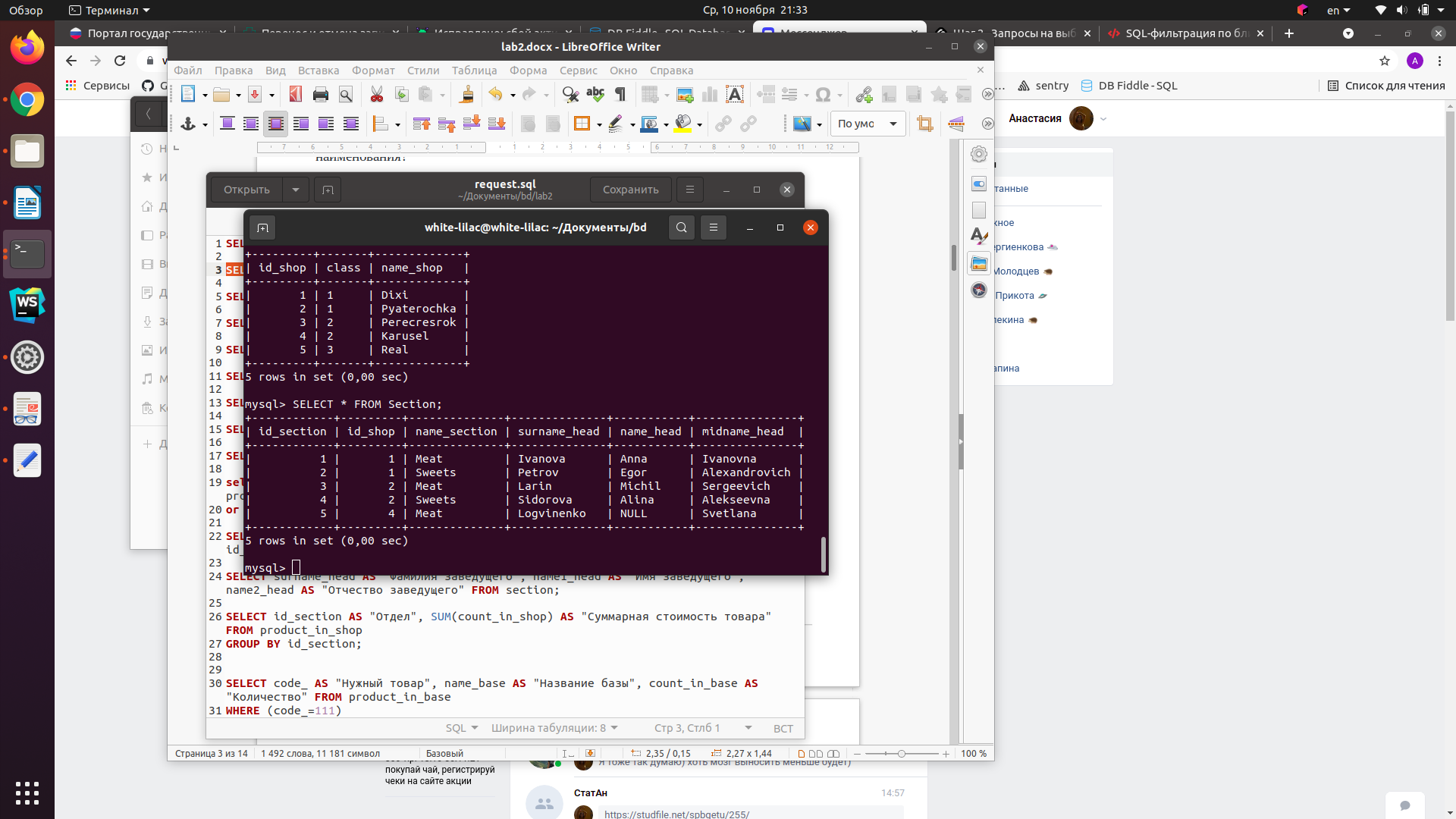
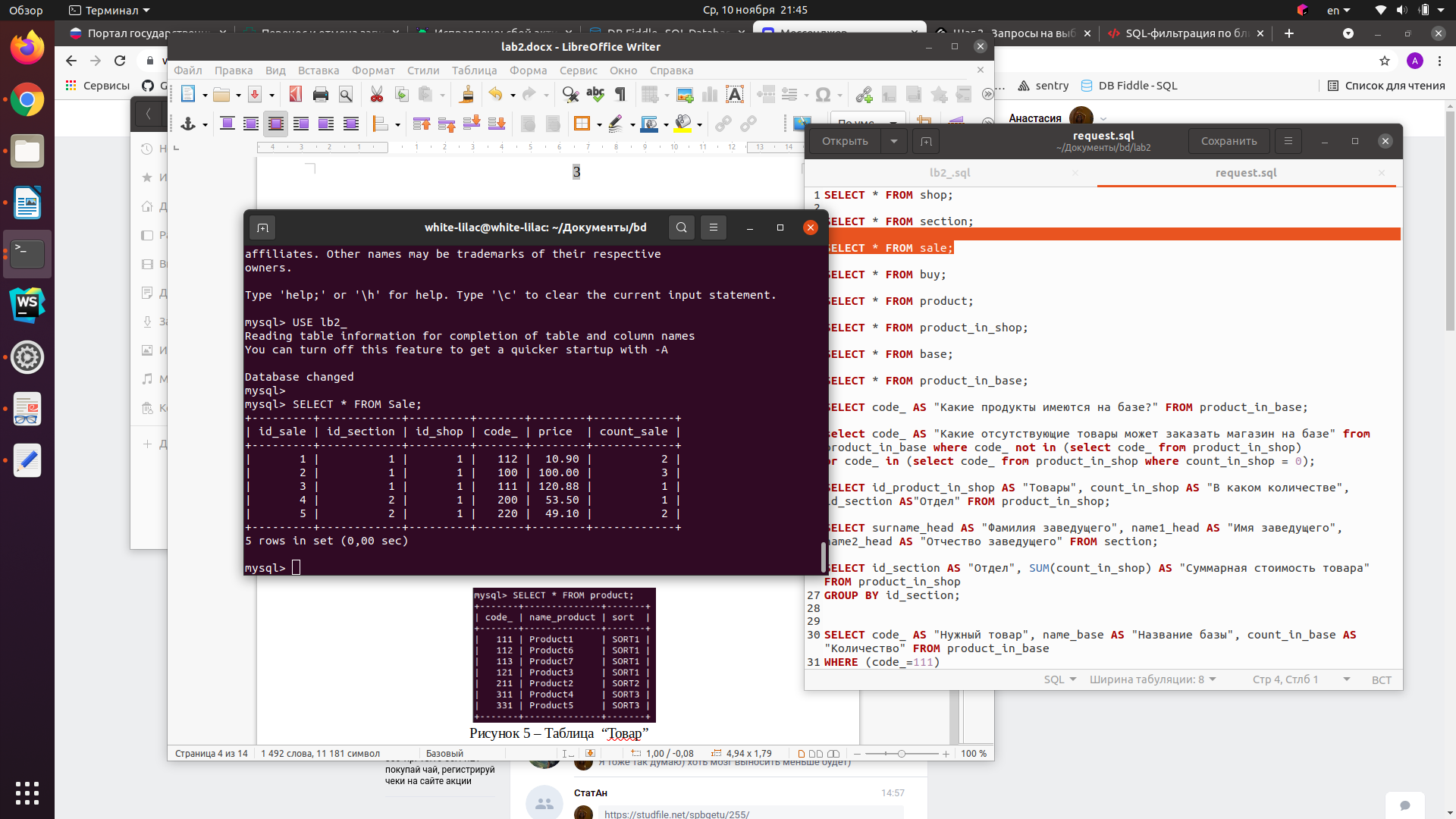


Рисунок 2 – Таблица “Отдел”

Рисунок 3 – Таблица “Продажа”

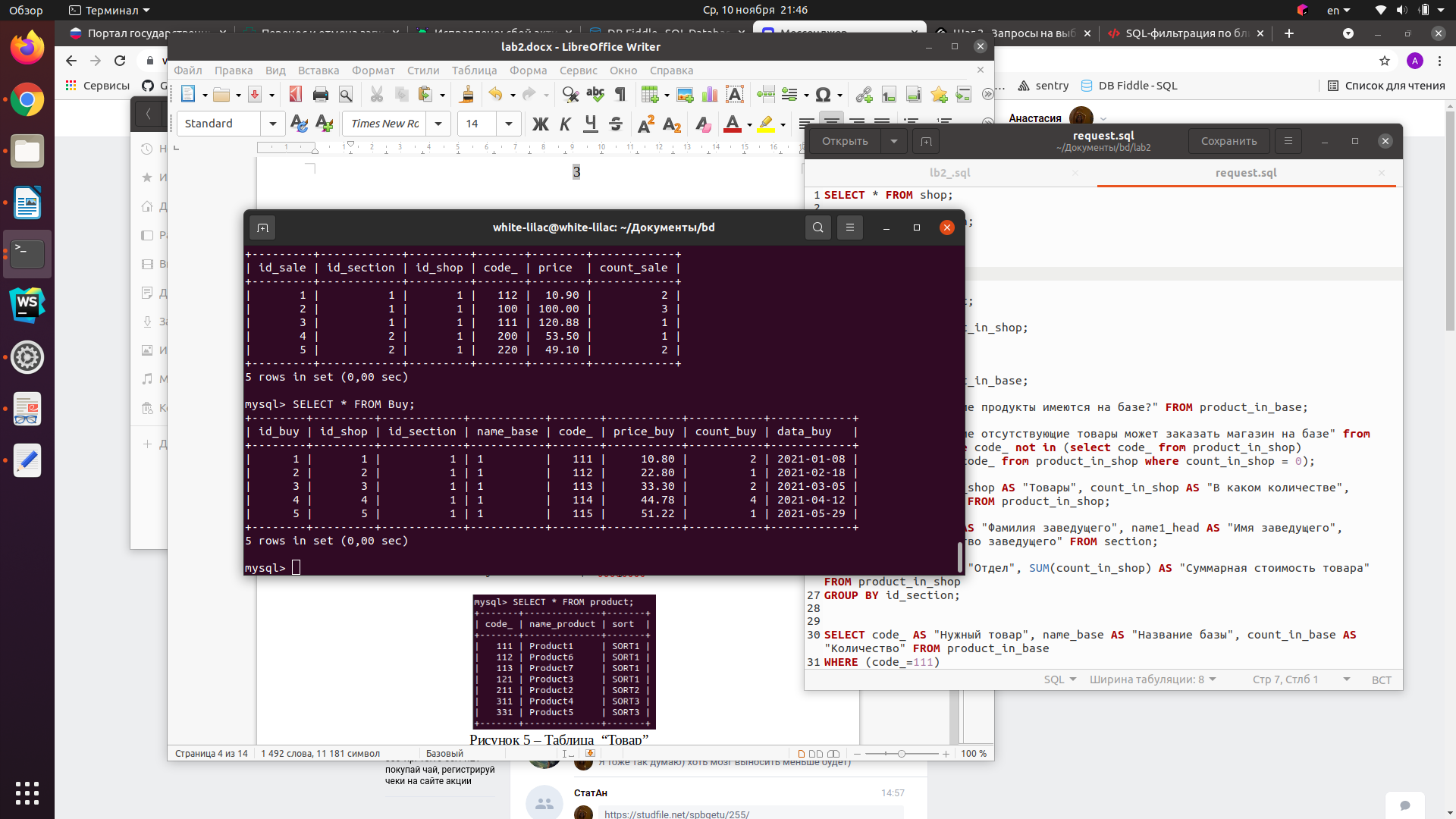


Рисунок 4 – Таблица “Закупка”

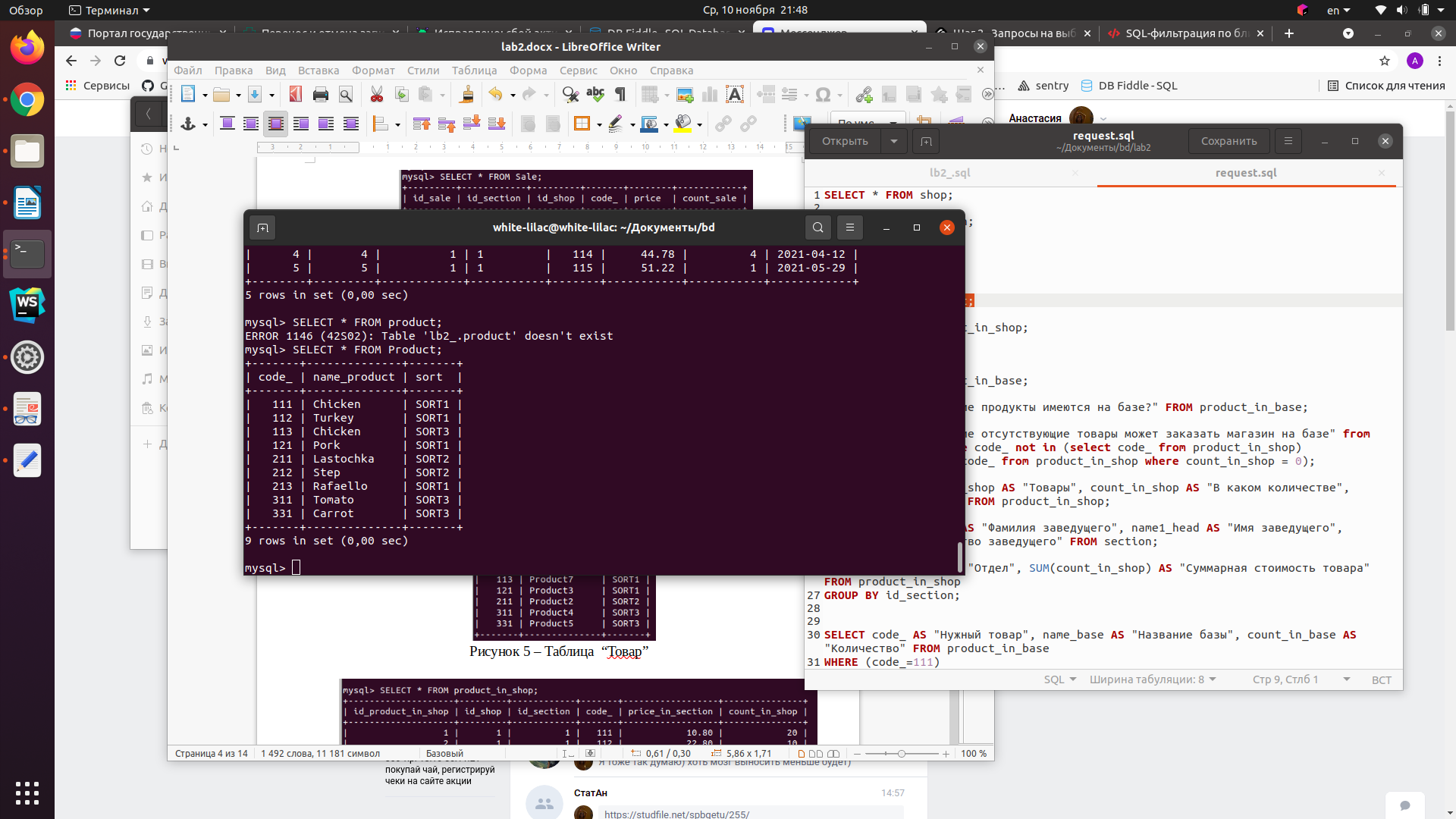


Рисунок 5 – Таблица “Товар”

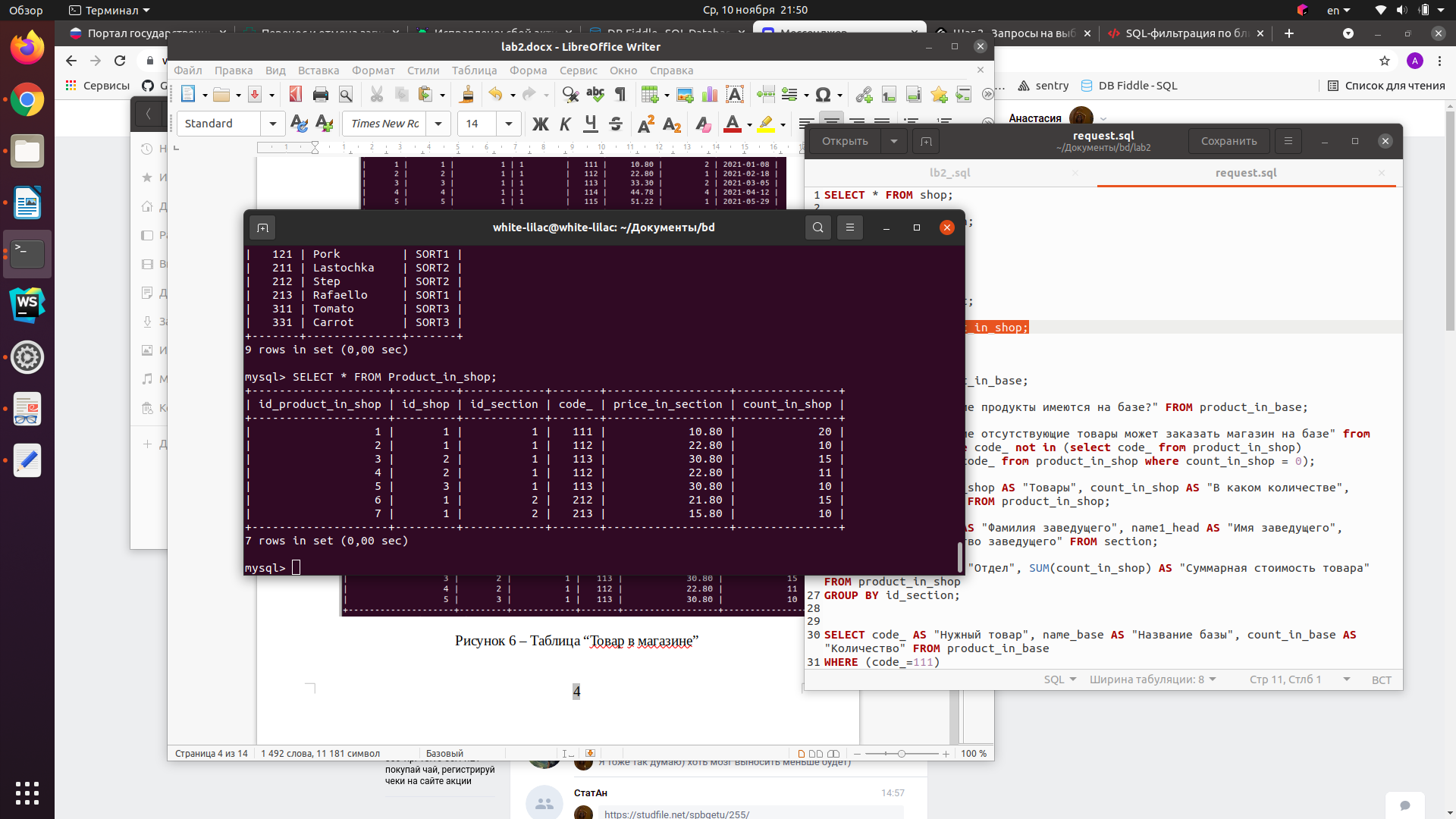


Рисунок 6 – Таблица “Товар в магазине”

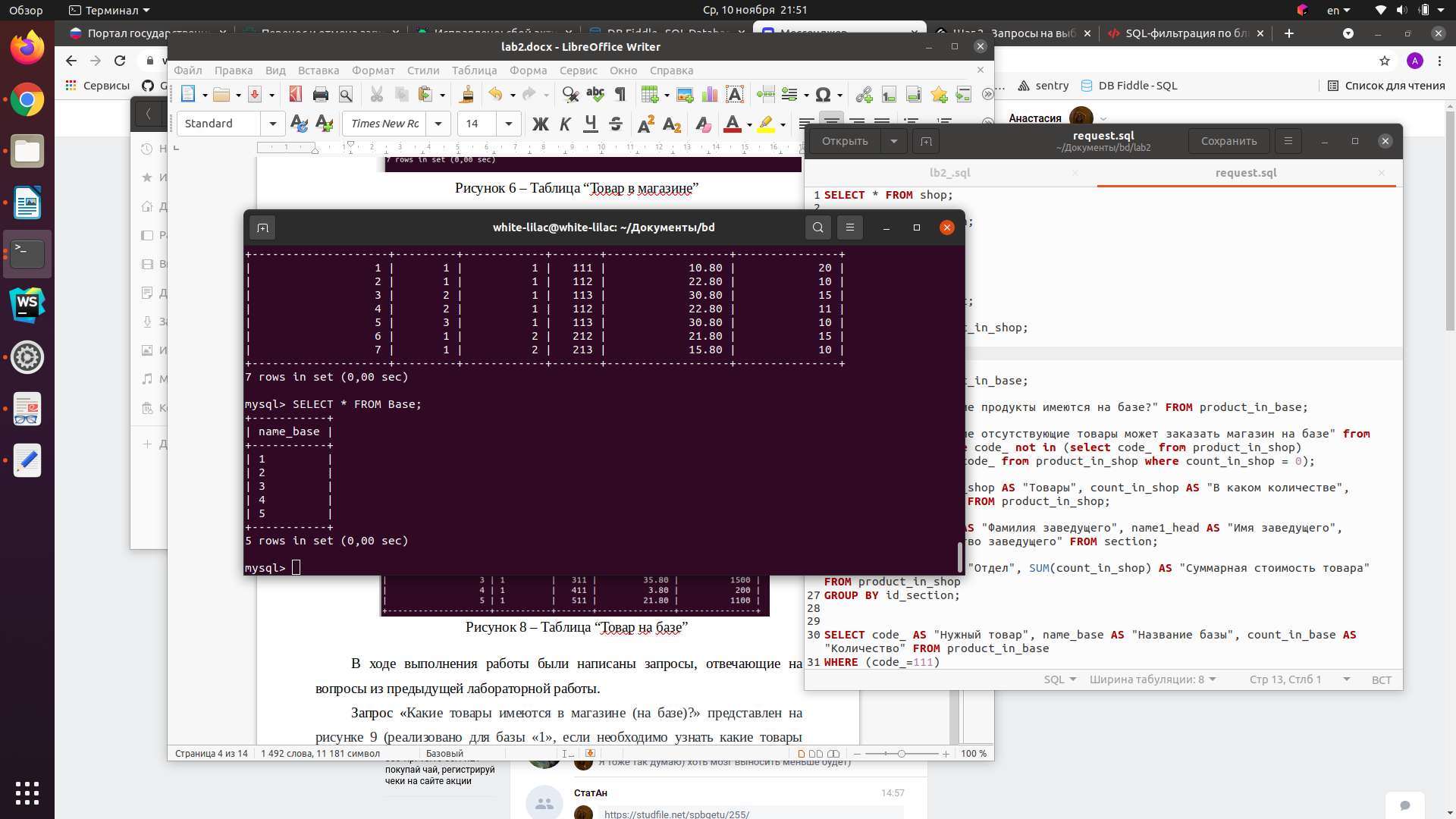


Рисунок 7 – Таблица “База”

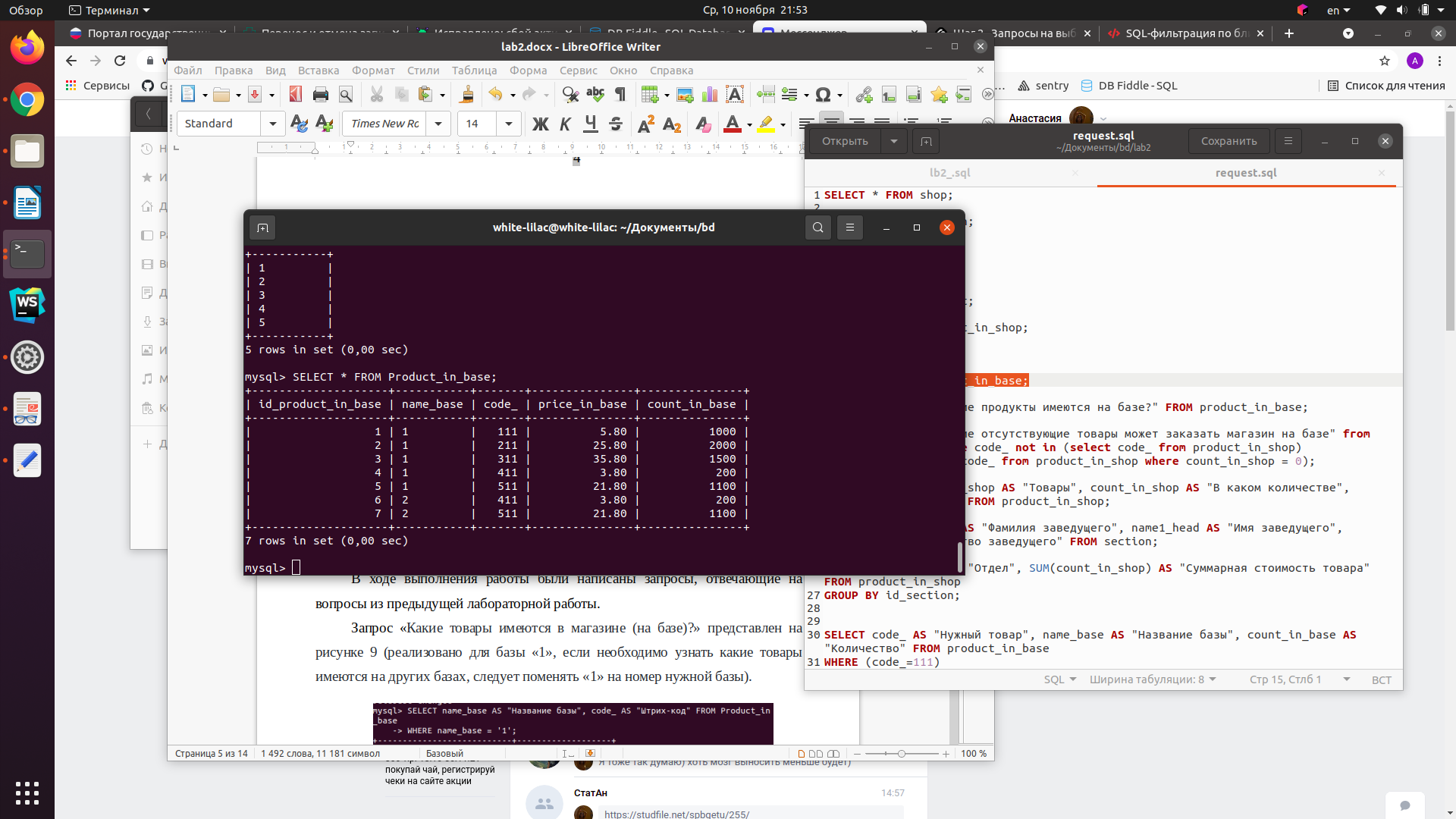


Рисунок 8 – Таблица “Товар на базе”

В ходе выполнения работы были написаны запросы, отвечающие на вопросы из предыдущей лабораторной работы.

Запрос «Какие товары имеются в магазине (на базе)?» представлен на рисунке 9 (реализовано для базы «1», если необходимо узнать какие товары имеются на других базах, следует поменять «1» на номер нужной базы).

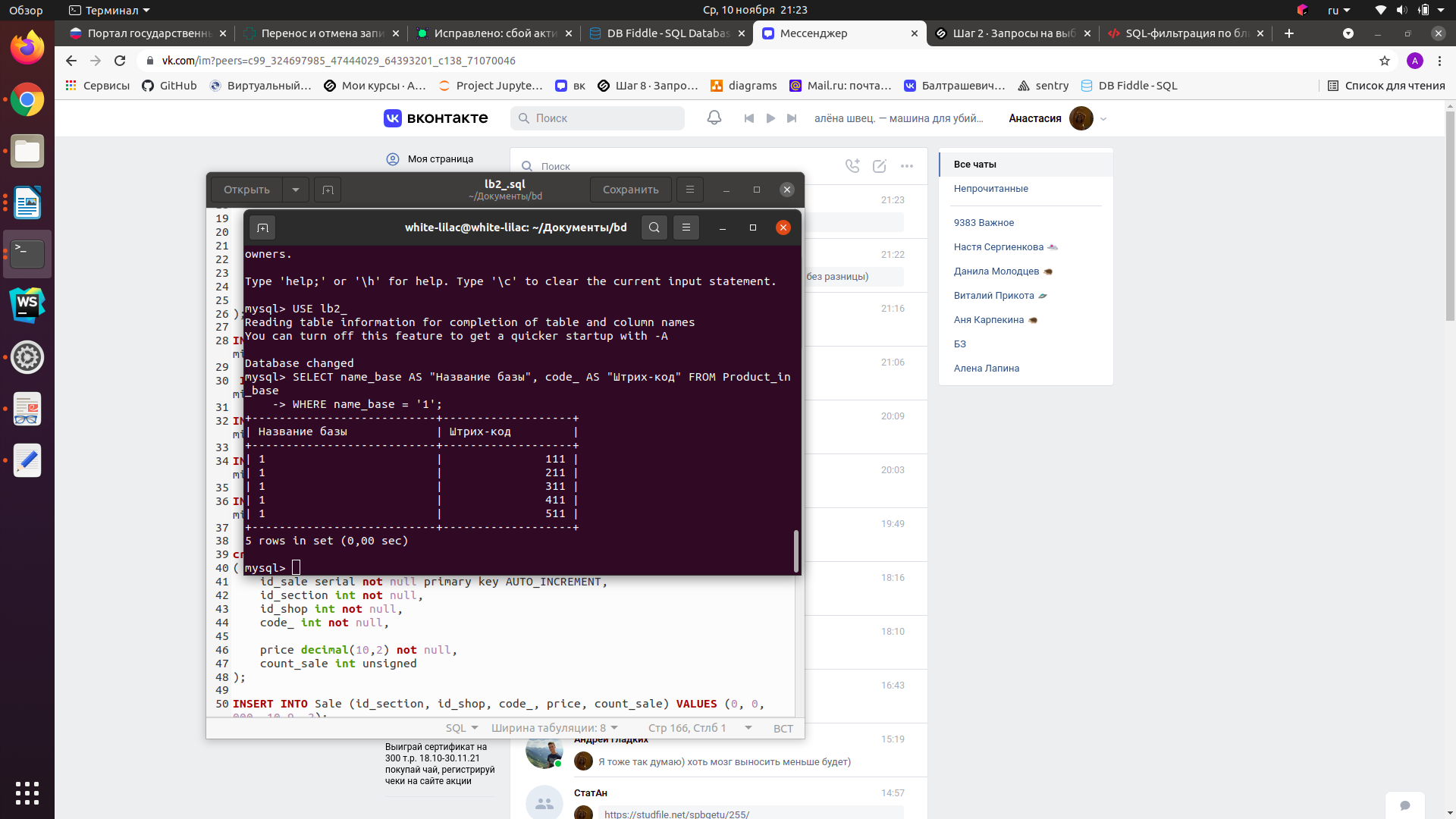


Рисунок 9 – Выборка товаров, имеющихся в магазине

Запрос «Какие отсутствующие товары может заказать магазин на базе?», представлен на рисунке 10.

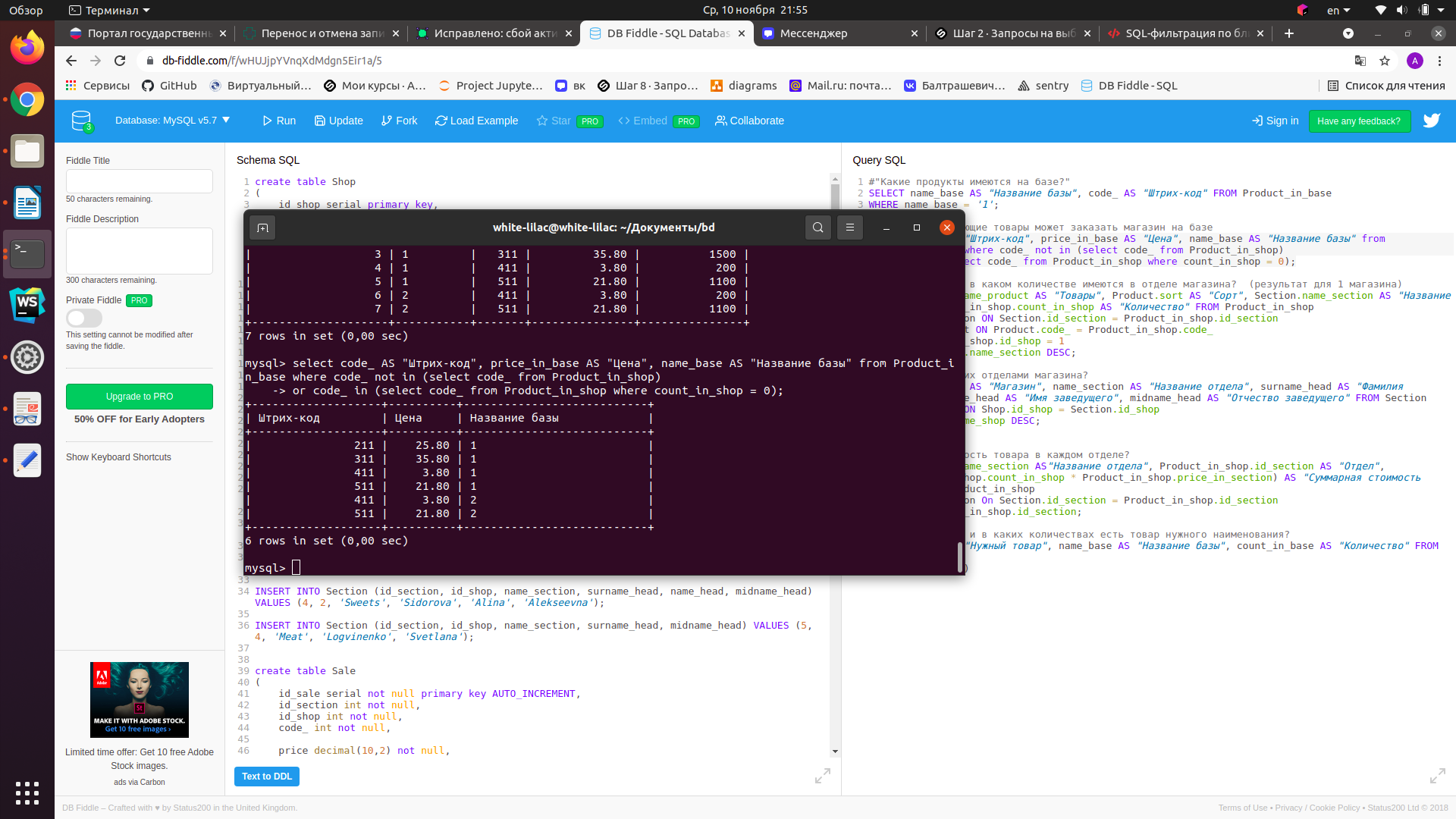


Рисунок 10 – Выборка товаров, которые магазин может заказать на базе

Запрос «Какие товары, и в каком количестве имеются в отделе магазина?» представлен на рисунке 11 (реализовано для магазина 1, если необходимо найти информацию для другого магазина, необходимо указать вместо «1» нужный номер магазина).

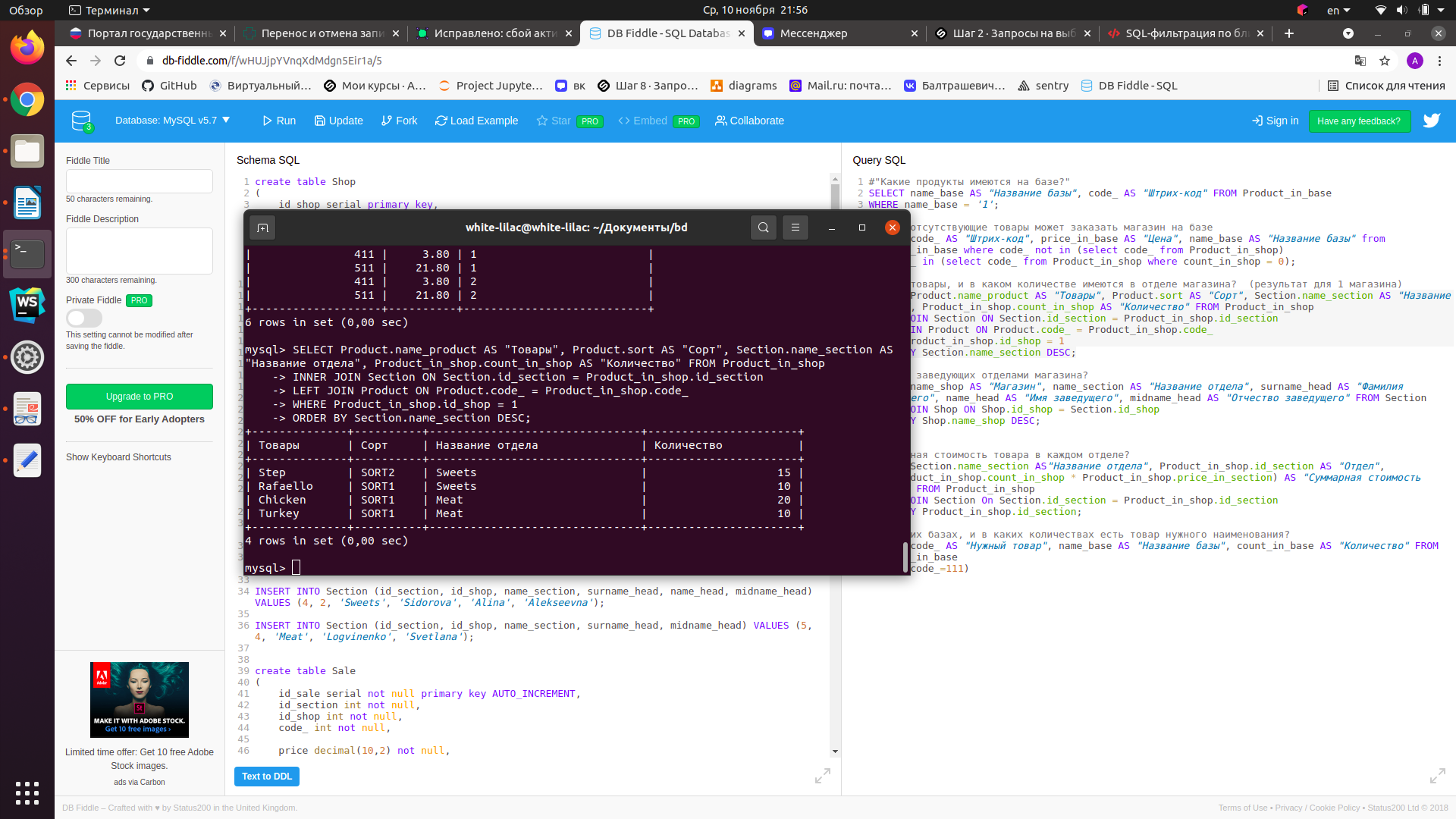


Рисунок 11 – Выборка товаров и их количества

Запрос «Список заведующих отделами магазина?» представлен на рисунке 12.

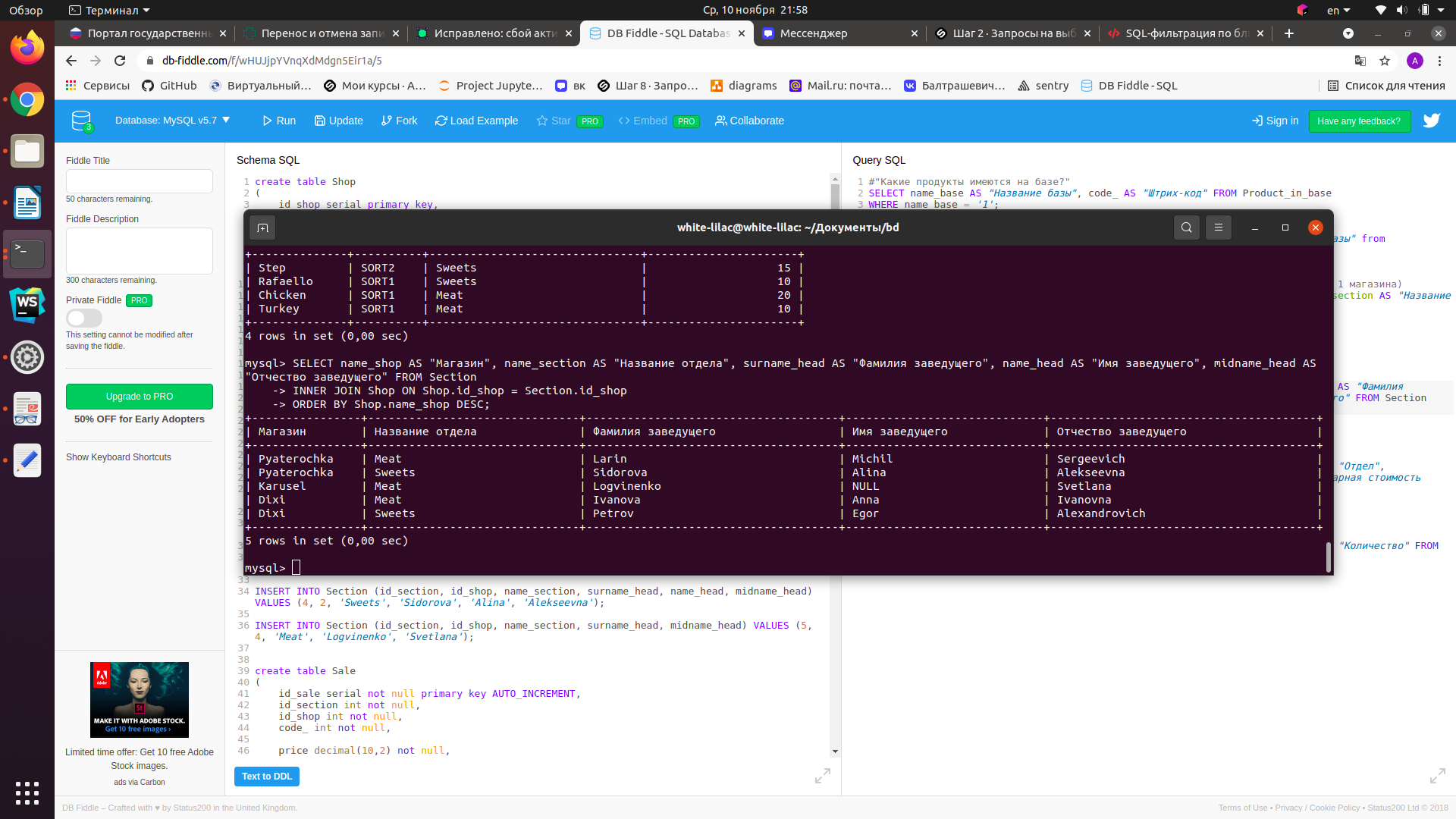


Рисунок 12 – Выборка заведующих отделами магазина

Запрос «Суммарная стоимость товара в каждом отделе?» представлен на рисунке 13.

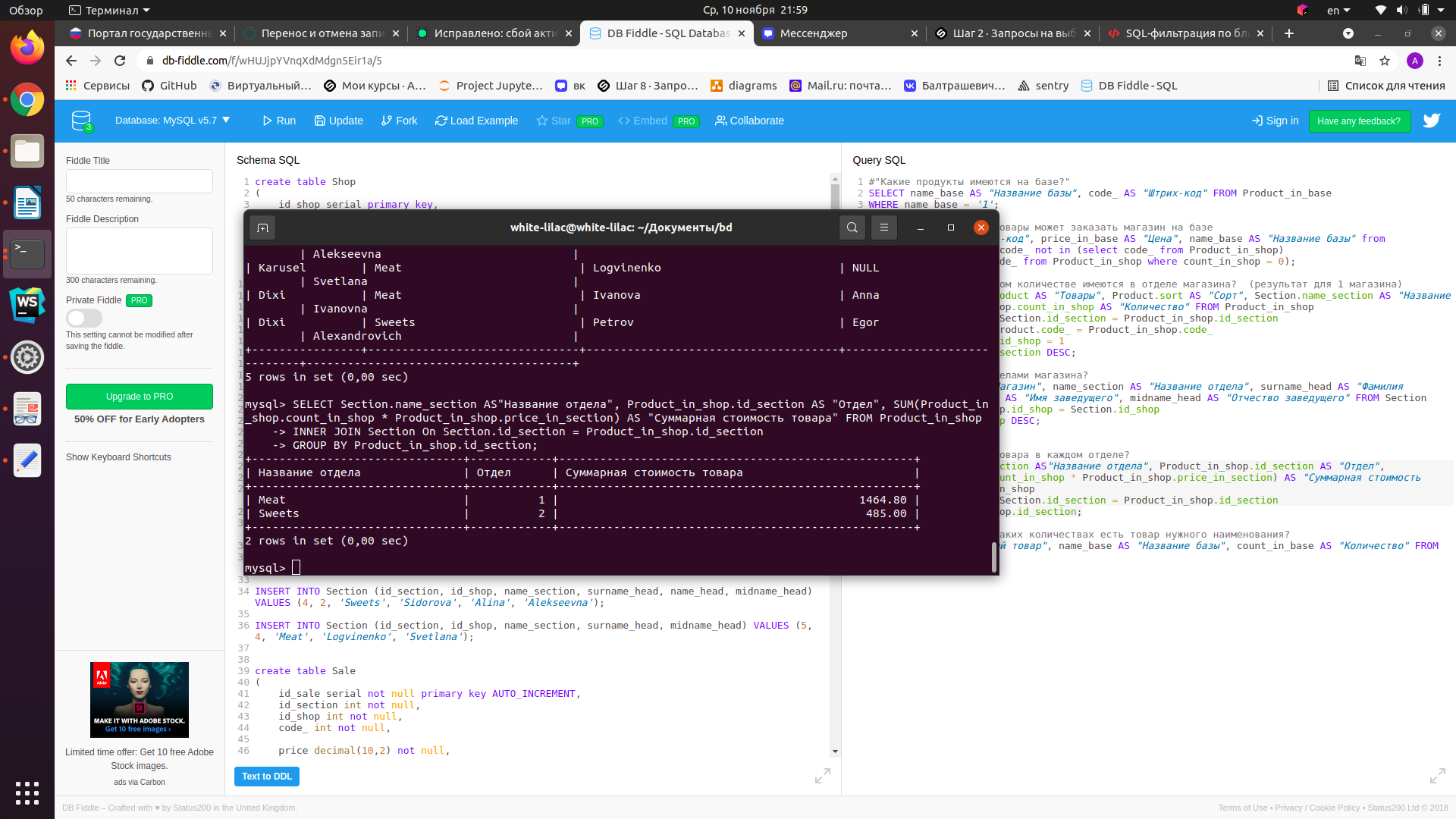


Рисунок 13 – Выборка суммарной стоимости товара в каждом отделе

Запрос «На каких базах, и в каких количествах есть товар нужного наименования?». Для выборки был взят товар «111». Результат представлен на рисунке 14.

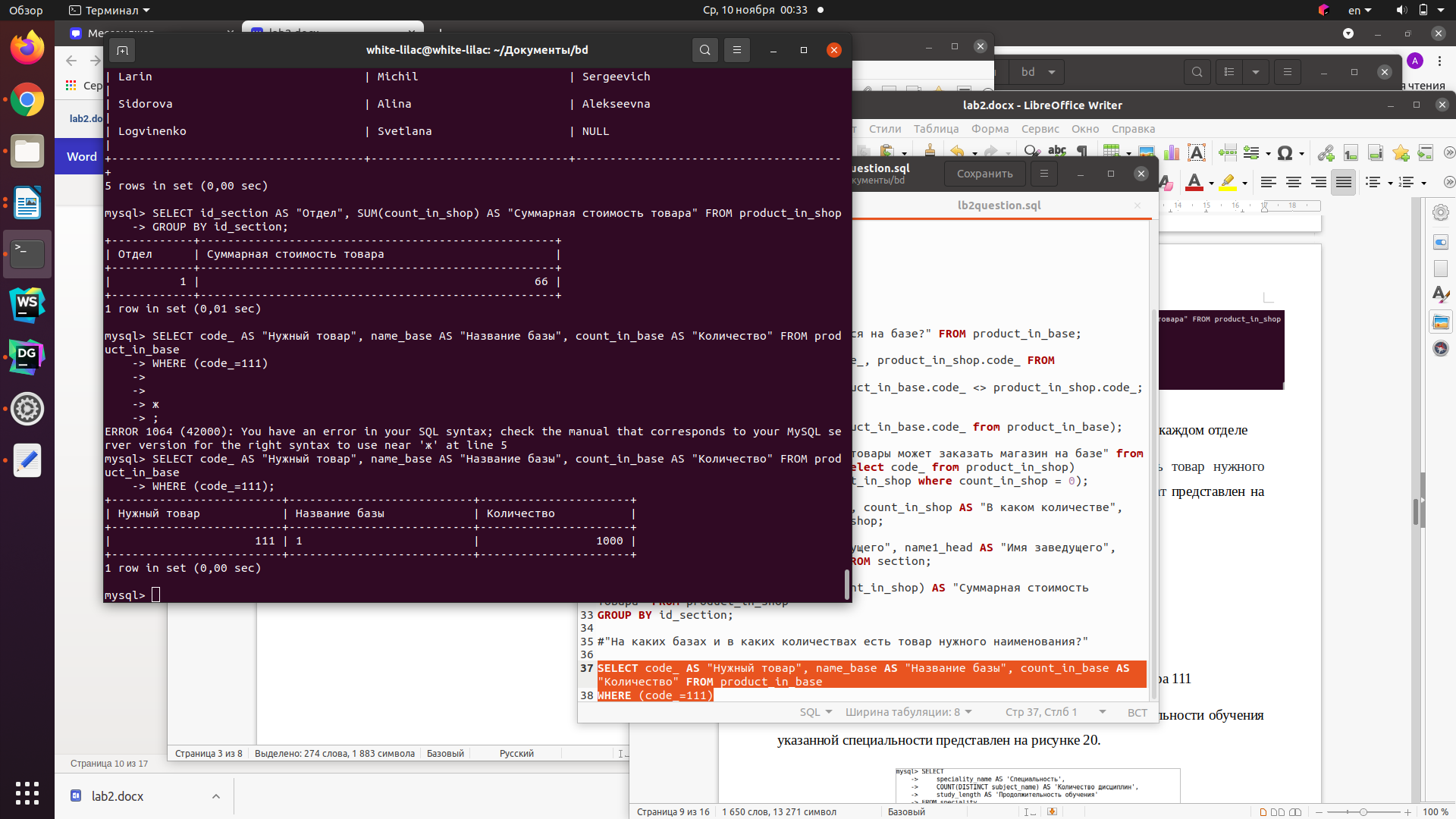


Рисунок 14 – Выборка баз и количество товара «111»

Исходный код представлен в приложении А. Ссылки на pull request с данным отчётом и на исходный код на www.db-fiddle.com представлены в приложении Б и В соответственно.

## Выводы.

В ходе работы был изучен синтаксис MySQL. Была реализована база данных, описанная в предыдущей лабораторной работе.

# Приложение А Исходный код

Название файла: lb2.sql

create table Shop

(

id\_shop int not null primary key AUTO\_INCREMENT,

class enum ('1', '2', '3'),

name\_shop varchar(30)

);

create table Base

(

name\_base varchar(20) not null primary key

);

create table Product

(

code\_ int not null primary key AUTO\_INCREMENT,

name\_product varchar(15),

sort varchar(15)

);

create table Section

(

id\_section int not null primary key AUTO\_INCREMENT,

id\_shop int not null,

name\_section varchar(20),

surname\_head varchar(20),

name\_head varchar(15),

midname\_head varchar(15) null,

FOREIGN KEY (id\_shop) REFERENCES Shop(id\_shop)

);

create table Sale

(

id\_sale serial not null primary key,

id\_section int not null,

id\_shop int not null,

code\_ int not null,

price decimal(10,2) not null,

count\_sale int unsigned,

FOREIGN KEY (id\_section) REFERENCES Section(id\_section),

FOREIGN KEY (id\_shop) REFERENCES Shop(id\_shop),

FOREIGN KEY (code\_) REFERENCES Product(code\_)

);

create table Buy

(

id\_buy serial not null,

id\_shop int not null,

id\_section int not null,

name\_base varchar(20),

code\_ int not null,

price\_buy decimal(10,2),

count\_buy int unsigned,

data\_buy date,

FOREIGN KEY (id\_shop) REFERENCES Shop(id\_shop),

FOREIGN KEY (id\_section) REFERENCES Section(id\_section),

FOREIGN KEY (name\_base) REFERENCES Base(name\_base),

FOREIGN KEY (code\_) REFERENCES Product(code\_)

);

create table Product\_in\_shop

(

id\_product\_in\_shop serial not null primary key,

id\_shop int not null,

id\_section int not null,

code\_ int not null,

price\_in\_section decimal(10,2),

count\_in\_shop int unsigned,

FOREIGN KEY (id\_shop) REFERENCES Shop(id\_shop),

FOREIGN KEY (id\_section) REFERENCES Section(id\_section),

FOREIGN KEY (code\_) REFERENCES Product(code\_)

);

create table Product\_in\_base

(

id\_product\_in\_base serial not null primary key,

name\_base varchar(20),

code\_ int not null,

price\_in\_base decimal(10,2),

count\_in\_base int unsigned,

FOREIGN KEY (name\_base) REFERENCES Base(name\_base),

FOREIGN KEY (code\_) REFERENCES Product(code\_)

);

INSERT INTO Shop (class, name\_shop) VALUES (1, 'Dixi');

INSERT INTO Shop (class, name\_shop) VALUES (1, 'Pyaterochka');

INSERT INTO Shop (class, name\_shop) VALUES (2, 'Perecresrok');

INSERT INTO Shop (class, name\_shop) VALUES (2, 'Karusel');

INSERT INTO Shop (class, name\_shop) VALUES (3, 'Real');

INSERT INTO Base (name\_base) VALUES ('1');

INSERT INTO Base (name\_base) VALUES ('2');

INSERT INTO Base (name\_base) VALUES ('3');

INSERT INTO Base (name\_base) VALUES ('4');

INSERT INTO Base (name\_base) VALUES ('5');

INSERT INTO Product (code\_, name\_product, sort) VALUES (111, 'Chicken', 'SORT1');

INSERT INTO Product (code\_, name\_product, sort) VALUES (211, 'Lastochka', 'SORT2');

INSERT INTO Product (code\_, name\_product, sort) VALUES (121, 'Pork', 'SORT1');

INSERT INTO Product (code\_, name\_product, sort) VALUES (311, 'Tomato', 'SORT3');

INSERT INTO Product (code\_, name\_product, sort) VALUES (331, 'Carrot', 'SORT3');

INSERT INTO Product (code\_, name\_product, sort) VALUES (112, 'Turkey', 'SORT1');

INSERT INTO Product (code\_, name\_product, sort) VALUES (113, 'Chicken', 'SORT3');

INSERT INTO Product (code\_, name\_product, sort) VALUES (212, 'Step', 'SORT2');

INSERT INTO Product (code\_, name\_product, sort) VALUES (213, 'Rafaello', 'SORT1');

INSERT INTO Product (code\_, name\_product, sort) VALUES (200, 'Romashka', 'SORT3');

INSERT INTO Product (code\_, name\_product, sort) VALUES (220, '35', 'SORT2');

INSERT INTO Product (code\_, name\_product, sort) VALUES (100, 'Turkey', 'SORT3');

INSERT INTO Section (id\_section, id\_shop, name\_section, surname\_head, name\_head, midname\_head) VALUES (1, 1, 'Meat', 'Ivanova', 'Anna', 'Ivanovna');

INSERT INTO Section (id\_section, id\_shop, name\_section, surname\_head, name\_head, midname\_head) VALUES (2, 1, 'Sweets', 'Petrov', 'Egor', 'Alexandrovich');

INSERT INTO Section (id\_section, id\_shop, name\_section, surname\_head, name\_head, midname\_head) VALUES (3, 2, 'Meat', 'Larin', 'Michil', 'Sergeevich');

INSERT INTO Section (id\_section, id\_shop, name\_section, surname\_head, name\_head, midname\_head) VALUES (4, 2, 'Sweets', 'Sidorova', 'Alina', 'Alekseevna');

INSERT INTO Section (id\_section, id\_shop, name\_section, surname\_head, midname\_head) VALUES (5, 4, 'Meat', 'Logvinenko', 'Svetlana');

INSERT INTO Sale (id\_section, id\_shop, code\_, price, count\_sale) VALUES (1, 1, 112, 10.9, 2);

INSERT INTO Sale (id\_section, id\_shop, code\_, price, count\_sale) VALUES (1, 1, 100, 100, 3);

INSERT INTO Sale (id\_section, id\_shop, code\_, price, count\_sale) VALUES (1, 1, 111, 120.88, 1);

INSERT INTO Sale (id\_section, id\_shop, code\_, price, count\_sale) VALUES (2, 1, 200, 53.5, 1);

INSERT INTO Sale (id\_section, id\_shop, code\_, price, count\_sale) VALUES (2, 1, 220, 49.1, 2);

INSERT INTO Buy (id\_buy, id\_shop, id\_section, name\_base, code\_, price\_buy, count\_buy, data\_buy) VALUES (1, 1, 1, '1', 111, 10.8, 2, '2021-01-08');

INSERT INTO Buy (id\_buy, id\_shop, id\_section, name\_base, code\_, price\_buy, count\_buy, data\_buy) VALUES (2, 2, 1, '1', 112, 22.8, 1, '2021-02-18');

INSERT INTO Buy (id\_buy, id\_shop, id\_section, name\_base, code\_, price\_buy, count\_buy, data\_buy) VALUES (3, 3, 1, '1', 113, 33.3, 2, '2021-03-05');

INSERT INTO Buy (id\_buy, id\_shop, id\_section, name\_base, code\_, price\_buy, count\_buy, data\_buy) VALUES (4, 4, 1, '1', 100, 44.78, 4, '2021-04-12');

INSERT INTO Buy (id\_buy, id\_shop, id\_section, name\_base, code\_, price\_buy, count\_buy, data\_buy) VALUES (5, 5, 1, '2', 200, 51.22, 1, '2021-05-29');

INSERT INTO Product\_in\_shop (id\_product\_in\_shop, id\_shop, id\_section, code\_, price\_in\_section, count\_in\_shop) VALUES (1, 1, 1, 111, 10.8, 20);

INSERT INTO Product\_in\_shop (id\_product\_in\_shop, id\_shop, id\_section, code\_, price\_in\_section, count\_in\_shop) VALUES (2, 1, 1, 112, 22.8, 10);

INSERT INTO Product\_in\_shop (id\_product\_in\_shop, id\_shop, id\_section, code\_, price\_in\_section, count\_in\_shop) VALUES (3, 2, 1, 113, 30.8, 15);

INSERT INTO Product\_in\_shop (id\_product\_in\_shop, id\_shop, id\_section, code\_, price\_in\_section, count\_in\_shop) VALUES (4, 2, 1, 112, 22.8, 11);

INSERT INTO Product\_in\_shop (id\_product\_in\_shop, id\_shop, id\_section, code\_, price\_in\_section, count\_in\_shop) VALUES (5, 3, 1, 113, 30.8, 10);

INSERT INTO Product\_in\_shop (id\_product\_in\_shop, id\_shop, id\_section, code\_, price\_in\_section, count\_in\_shop) VALUES (6, 1, 2, 212, 21.8, 15);

INSERT INTO Product\_in\_shop (id\_product\_in\_shop, id\_shop, id\_section, code\_, price\_in\_section, count\_in\_shop) VALUES (7, 1, 2, 213, 15.8, 10);

INSERT INTO Product\_in\_base (id\_product\_in\_base, name\_base, code\_, price\_in\_base, count\_in\_base) VALUES (1, '1', 111, 5.8, 1000);

INSERT INTO Product\_in\_base (id\_product\_in\_base, name\_base, code\_, price\_in\_base, count\_in\_base) VALUES (2, '1', 211, 25.8, 2000);

INSERT INTO Product\_in\_base (id\_product\_in\_base, name\_base, code\_, price\_in\_base, count\_in\_base) VALUES (3, '1', 311, 35.8, 1500);

INSERT INTO Product\_in\_base (id\_product\_in\_base, name\_base, code\_, price\_in\_base, count\_in\_base) VALUES (4, '1', 212, 3.8, 200);

INSERT INTO Product\_in\_base (id\_product\_in\_base, name\_base, code\_, price\_in\_base, count\_in\_base) VALUES (5, '1', 213, 21.8, 1100);

INSERT INTO Product\_in\_base (id\_product\_in\_base, name\_base, code\_, price\_in\_base, count\_in\_base) VALUES (6, '2', 212, 3.8, 200);

INSERT INTO Product\_in\_base (id\_product\_in\_base, name\_base, code\_, price\_in\_base, count\_in\_base) VALUES (7, '2', 213, 21.8, 1100);

#"Какие продукты имеются на базе?"

SELECT name\_base AS "Название базы", code\_ AS "Штрих-код" FROM Product\_in\_base

WHERE name\_base = '1';

#Какие отсутствующие товары может заказать магазин на базе

select code\_ AS "Штрих-код", price\_in\_base AS "Цена", name\_base AS "Название базы" from Product\_in\_base where code\_ not in (select code\_ from Product\_in\_shop)

or code\_ in (select code\_ from Product\_in\_shop where count\_in\_shop = 0);

#Какие товары, и в каком количестве имеются в отделе магазина? (результат для 1 магазина)

SELECT Product.name\_product AS "Товары", Product.sort AS "Сорт", Section.name\_section AS "Название отдела", Product\_in\_shop.count\_in\_shop AS "Количество" FROM Product\_in\_shop

INNER JOIN Section ON Section.id\_section = Product\_in\_shop.id\_section

LEFT JOIN Product ON Product.code\_ = Product\_in\_shop.code\_

WHERE Product\_in\_shop.id\_shop = 1

ORDER BY Section.name\_section DESC;

#Список заведующих отделами магазина?

SELECT name\_shop AS "Магазин", name\_section AS "Название отдела", surname\_head AS "Фамилия заведущего", name\_head AS "Имя заведущего", midname\_head AS "Отчество заведущего" FROM Section

INNER JOIN Shop ON Shop.id\_shop = Section.id\_shop

ORDER BY Shop.name\_shop DESC;

#Суммарная стоимость товара в каждом отделе?

SELECT Section.name\_section AS"Название отдела", Product\_in\_shop.id\_section AS "Отдел", SUM(Product\_in\_shop.count\_in\_shop \* Product\_in\_shop.price\_in\_section) AS "Суммарная стоимость товара" FROM Product\_in\_shop

INNER JOIN Section On Section.id\_section = Product\_in\_shop.id\_section

GROUP BY Product\_in\_shop.id\_section;

#На каких базах, и в каких количествах есть товар нужного наименования?

SELECT code\_ AS "Нужный товар", name\_base AS "Название базы", count\_in\_base AS "Количество" FROM Product\_in\_base

WHERE (code\_=111);

# Приложение Б Необходимые Ссылки

https://github.com/moevm/sql-2021-9383/pull/17

# Приложение В

# Необходимые Ссылки

https://www.db-fiddle.com/f/wHUJjpYVnqXdMdgn5Eir1a/7