

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 соответствует 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, предпочтая 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.

mysql> SELECT * FROM Shop;		
id_shop	class	name_shop
1	1	Dixi
2	1	Pyaterochka
3	2	Perecresrok
4	2	Karusel
5	3	Real

Рисунок 1 – Таблица “Магазин”

mysql> SELECT * FROM Section;					
id_section	id_shop	name_section	surname_head	name_head	midname_head
1	1	Meat	Ivanova	Anna	Ivanovna
2	1	Sweets	Petrov	Egor	Alexandrovich
3	2	Meat	Larin	Michil	Sergeevich
4	2	Sweets	Sidorova	Alina	Alekseevna
5	4	Meat	Logvinenko	NULL	Svetlana

Рисунок 2 – Таблица “Отдел”

```

mysql> SELECT * FROM Sale;
+-----+-----+-----+-----+-----+
| id_sale | id_section | id_shop | code_ | price   | count_sale |
+-----+-----+-----+-----+-----+
|      1 |          1 |        1 | 112 | 10.90 |         2 |
|      2 |          1 |        1 | 100 | 100.00 |         3 |
|      3 |          1 |        1 | 111 | 120.88 |         1 |
|      4 |          2 |        1 | 200 | 53.50 |         1 |
|      5 |          2 |        1 | 220 | 49.10 |         2 |
+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)

```

Рисунок 3 – Таблица “Продажа”

```

mysql> SELECT * FROM Buy;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_buy | id_shop | id_section | name_base | code_ | price_buy | count_buy | data_buy   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      1 |        1 |        1 |    1 | 111 | 10.80 |         2 | 2021-01-08 |
|      2 |        2 |        1 |    1 | 112 | 22.80 |         1 | 2021-02-18 |
|      3 |        3 |        1 |    1 | 113 | 33.30 |         2 | 2021-03-05 |
|      4 |        4 |        1 |    1 | 114 | 44.78 |         4 | 2021-04-12 |
|      5 |        5 |        1 |    1 | 115 | 51.22 |         1 | 2021-05-29 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)

mysql> []

```

Рисунок 4 – Таблица “Закупка”

```

mysql> SELECT * FROM Product;
+-----+-----+-----+
| code_ | name_product | sort   |
+-----+-----+-----+
| 111 | Chicken     | SORT1 |
| 112 | Turkey       | SORT1 |
| 113 | Chicken     | SORT3 |
| 121 | Pork          | SORT1 |
| 211 | Lastochka   | SORT2 |
| 212 | Step          | SORT2 |
| 213 | Rafaello     | SORT1 |
| 311 | Tomato       | SORT3 |
| 331 | Carrot       | SORT3 |
+-----+-----+-----+
9 rows in set (0,00 sec)

```

Рисунок 5 – Таблица “Товар”

```

mysql> SELECT * FROM Product_in_shop;
+-----+-----+-----+-----+-----+-----+
| id_product_in_shop | id_shop | id_section | code_ | price_in_section | count_in_shop |
+-----+-----+-----+-----+-----+-----+
|      1 |        1 |        1 | 111 | 10.80 |        20 |
|      2 |        1 |        1 | 112 | 22.80 |        10 |
|      3 |        2 |        1 | 113 | 30.80 |        15 |
|      4 |        2 |        1 | 112 | 22.80 |        11 |
|      5 |        3 |        1 | 113 | 30.80 |        10 |
|      6 |        1 |        2 | 212 | 21.80 |        15 |
|      7 |        1 |        2 | 213 | 15.80 |        10 |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0,00 sec)

```

Рисунок 6 – Таблица “Товар в магазине”

```

mysql> SELECT * FROM Base;
+-----+
| name_base |
+-----+
| 1          |
| 2          |
| 3          |
| 4          |
| 5          |
+-----+
5 rows in set (0,00 sec)

```

Рисунок 7 – Таблица “База”

```

mysql> SELECT * FROM Product_in_base;
+-----+-----+-----+-----+-----+
| id_product_in_base | name_base | code_ | price_in_base | count_in_base |
+-----+-----+-----+-----+-----+
| 1     | 1       | 111   | 5.80  | 1000    |
| 2     | 1       | 211   | 25.80 | 2000    |
| 3     | 1       | 311   | 35.80 | 1500    |
| 4     | 1       | 411   | 3.80  | 200     |
| 5     | 1       | 511   | 21.80 | 1100    |
| 6     | 2       | 411   | 3.80  | 200     |
| 7     | 2       | 511   | 21.80 | 1100    |
+-----+-----+-----+-----+-----+
7 rows in set (0,00 sec)

```

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).

```

mysql> SELECT name_base AS "Название базы", code_ AS "Штрих-код" FROM Product_in_base
-> WHERE name_base = '1';
+-----+-----+
| Название базы | Штрих-код |
+-----+-----+
| 1             | 111      |
| 1             | 211      |
| 1             | 311      |
| 1             | 411      |
| 1             | 511      |
+-----+-----+
5 rows in set (0,00 sec)

```

Рисунок 9 – Выборка товаров, имеющихся в магазине

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

```

mysql> 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);
+-----+-----+-----+
| Штрих-код | Цена   | Название базы |
+-----+-----+-----+
|     211   | 25.80 | 1
|     311   | 35.80 | 1
|     411   | 3.80  | 1
|     511   | 21.80 | 1
|     411   | 3.80  | 2
|     511   | 21.80 | 2
+-----+-----+-----+
6 rows in set (0,00 sec)

```

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).

```

mysql> 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;
+-----+-----+-----+-----+
| Товары | Сорт   | Название отдела | Количество |
+-----+-----+-----+-----+
| Step   | SORT2  | Sweets          | 15          |
| Rafaello | SORT1  | Sweets          | 10          |
| Chicken | SORT1  | Meat            | 20          |
| Turkey  | SORT1  | Meat            | 10          |
+-----+-----+-----+-----+
4 rows in set (0,00 sec)

```

Рисунок 11 – Выборка товаров и их количества

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

```

mysql> 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;
+-----+-----+-----+-----+-----+
| Магазин | Название отдела | Фамилия заведущего | Имя заведущего | Отчество заведущего |
+-----+-----+-----+-----+-----+
| Pyaterochka | Meat           | Larin           | Michil         | Sergeevich
| Pyaterochka | Sweets          | Sidorova        | Alina          | Alekseevna
| Karusel     | Meat           | Logvinenko     | NULL           | Svetlana
| Dixi        | Meat           | Ivanova         | Anna           | Ivanovna
| Dixi        | Sweets          | Petrov          | Egor           | Alexandrovich
+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)

```

Рисунок 12 – Выборка заведующих отделами магазина

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

```

mysql> 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;
+-----+-----+-----+
| Название отдела | Отдел | Суммарная стоимость товара |
+-----+-----+-----+
| Meat | 1 | 1464.80 |
| Sweets | 2 | 485.00 |
+-----+-----+
2 rows in set (0,00 sec)

```

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.

```

mysql> SELECT code_ AS "Нужный товар", name_base AS "Название базы", count_in_base AS "Количество" FROM product_in_base
-> WHERE (code_=111);
+-----+-----+-----+
| Нужный товар | Название базы | Количество |
+-----+-----+-----+
| 111 | 1 | 1000 |
+-----+-----+
1 row in set (0,00 sec)

```

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 заведующего", 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>