

В данной работе приведены решения трех задач на работу с реляционными базами данных вплоть до JOIN-запросов.

### Задача 1

**Создать:**

База данных **test**, содержащая следующие таблицы

Таблица **person** – поля: persID, lastname, firstname

Таблица **ordertab** – поля: persID, orderID, orderdate

Таблица **product** – поля: orderID, productID, productname, price

Поля **persID** и **orderID** являются внешними ключами

### Решение:

1. Отобразим существующие базы данных (чтобы удостовериться, что базы данных **test** не существует)

```
SHOW databases;
```

2. Создадим базу данных **test**

```
CREATE database test;
```

3. Проверим обновленный список баз данных, чтобы убедиться, что база данных с именем **test** успешно создана

```
SHOW databases;
```

4. Укажем системе, что будем работать с базой данных **test**

```
USE test;
```

5. Создаем таблицу **person**

```
CREATE table person;
```

6. Создадим поля persID, lastname и firstname

```
CREATE TABLE person (  
    persID INT NOT NULL,  
    firstname VARCHAR(10) NULL,  
    lastname VARCHAR(10) NULL,  
    PRIMARY KEY (persID)  
)
```

7. Заполним поля **persID**, **lastname** и **firstname** данными

```
INSERT INTO person VALUES (1,"Anton","Petrov"), (2,"Sergey","Ivanov"), (3,"Palpat","Sid"),  
(4,"Artag","Avorn"), (5,"John","Smith"), (6,"Art","Khan"), (7,"Konstantin","Frolov");
```

8. Проверим правильность заполнения таблицы **person** путем вывода всех ее полей

```
SELECT * FROM person;
```

```
mysql> SELECT * FROM person;  
+-----+-----+-----+  
| persID | firstname | lastname |  
+-----+-----+-----+  
| 1 | Anton | Petrov |  
| 2 | Sergey | Ivanov |  
| 3 | Palpat | Sid |  
| 4 | Artag | Avorn |  
| 5 | John | Smith |  
| 6 | Art | Khan |  
| 7 | Konstantin | Frolov |  
+-----+-----+-----+  
7 rows in set (0.00 sec)  
  
mysql>
```

9. Создаем таблицу **ordertab** и назначаем поле **persID** внешним ключом, соответствующим **persID** таблицы **person**

```
CREATE TABLE ordertab (  
    orderID INT NOT NULL,  
    persID INT NULL,  
    orderdate date NULL,  
    PRIMARY KEY (orderID),  
    FOREIGN KEY (persID) REFERENCES person (persID)  
);
```

10. Заполняем поля таблицы **ordertab**

```
INSERT INTO ordertab VALUES (1,2,"2014-08-11"), (2,5,"2015-10-21"),(3,1,"2005-11-15"), (4,1,"2013-  
03-08"), (5,4,"2013-03-08"), (6,5,"2013-03-08"), (7,1,"2013-03-08");
```

11. Проверим правильность заполнения таблицы **ordertab** путем вывода всех ее полей

```
SELECT * FROM ordertab;
```

```
mysql> SELECT * FROM ordertab;
```

orderID	persID	orderdate
1	2	2014-08-11
2	5	2015-10-21
3	1	2005-11-15
4	1	2013-03-08
5	4	2013-03-08
6	5	2013-03-08
7	1	2013-03-08

```
7 rows in set (0.00 sec)

mysql>
```

12. Создаем таблицу **product**, заполняем ее поля и назначаем поле **orderID** внешним ключом, соответствующим **orderID** таблицы **ordertab**

```
mysql> SELECT * FROM product;
```

productID	orderID	productname	price
1	1	apple	10
2	3	ORANGE	15
3	2	ORANGE	15
4	5	GARLIC	9
5	6	AVOCADO	17
6	5	LIME	21
7	4	LIME	21

```
7 rows in set (0.00 sec)

mysql>
```

## Задача 2

Написать SQL-запрос отображающий фамилии всех покупателей имена которых заканчиваются на «ov»

### Решение:

```
SELECT * FROM person WHERE lastname LIKE "%ov";
```

```
mysql> SELECT * FROM person WHERE lastname LIKE "%ov";
+-----+-----+-----+
| persID | firstname | lastname |
+-----+-----+-----+
|      1 | Anton     | Petrov   |
|      2 | Sergey    | Ivanov   |
|      7 | Konstantin | Frolov   |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

### Задача 3

Написать SQL-запрос, позволяющий найти все продукты, которые вошли в список покупок за 8 марта 2013г. В результате вывести неповторяющиеся названия продуктов и их цену

### Решение:

```
SHOW databases;           // отобразим все базы данных

USE test;                  // укажем Системе, что будем работать в базе данных test

SELECT product.productname AS "Product Name",    // #
product.price AS "Price",                          // #
ordertab.orderdate AS "Order Date",               // #
COUNT(product.productname) AS "Number of Product Name", // ##
SUM(product.price) AS "Total price of Product Name" // ###

FROM ordertab           // для INNER JOIN не имеет значения порядок таблиц

INNER JOIN product      // объединяем ordertab и product

ON ordertab.orderID = product.orderID             // по полю orderID (ключ)

WHERE ordertab.orderdate IN("2013-03-08")         // отобразив только поля для 8 Марта 2013 г.

GROUP BY product.productname, product.price, ordertab.orderdate; // #####
```

#

Отобразим в сводной таблице название поля productname из таблицы product как Product Name

Отобразим в сводной таблице название поля price из таблицы product как Price

Отобразим в сводной таблице название поля ordertab из таблицы orderdate как Order Date

##

Подсчитаем количество значений для поля productname из таблицы product и отобразим в сводной таблице для поля с названием Number of Product Name

###

Вычислим сумму всех значений в поле price таблицы product и отобразим в сводной таблице для поля с названием Total price of Product Name

####

Группируем товары по названию, цене и дате заказа – чтобы Система не выводила для этих полей только одно значение

```
mysql> SELECT product.productname AS "Product Name",
-> product.price AS "Price",
-> ordertab.orderdate AS "Order Date",
-> COUNT(product.productname) AS "Number of Product Name",
-> SUM(product.price) AS "Total price of Product Name"
-> FROM ordertab
-> INNER JOIN product
-> ON ordertab.orderID = product.orderID
-> WHERE ordertab.orderdate IN("2013-03-08")
-> GROUP BY product.productname, product.price, ordertab.orderdate;
```

Product Name	Price	Order Date	Number of Product Name	Total price of Product Name
LIME	21	2013-03-08	2	42
GARLIC	9	2013-03-08	1	9
AVOCADO	17	2013-03-08	1	17

3 rows in set (0.00 sec)

```
mysql>
```