

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет  
"Высшая школа экономики»**

Московский институт электроники и математики им. А. Н. Тихонова НИУ  
ВШЭ Департамент компьютерной инженерии

**Курс: Базы данных**

**Отчет по Лабораторной работе №2**

**Вариант №13**

**Студент: Юлдашов Алишер Сойибжонович**

**Москва 2021**

### Задания:

Один из запросов надо написать двумя способами и объяснить, какой из вариантов будет работать быстрее и почему.

Создать упорядоченные списки:

1. товаров с указанием их суммарного количества и веса;
  2. поставщиков и товаров, которые они поставляют;
  3. поставщиков, от которых не было поставок;
  4. товаров, остаток которых равен количеству поступившего товара и которых нет в текущих заказах.
- Проверить, достаточно ли на складе товара для выполнения конкретного заказа

### Добавление дополнительных полей для выполнения задания

```
INSERT INTO SuppliersCustomers VALUES (10, 'Redic', 'Moscow, St.4', '89266852819')
```

```
INSERT INTO SuppliersCustomers VALUES (11, 'Reipc', 'Moscopow, St.4', '89266862819')
```

```
INSERT INTO SuppliersCustomers VALUES (12, 'Roipc', 'Moocopow, St.4', '89066802819')
```

```
INSERT INTO Goods VALUES ('df', 'paper', 'kg', 10)
```

```
INSERT INTO Goods VALUES ('tf', 'paers', 'kg', 13)
```

```
INSERT INTO GoodsOnStorage VALUES (12501, 10, '2020-01-02', 'df', 12, '4', 80, 10)
```

```
INSERT INTO Goods VALUES ('kf', 'saper', 'sg', 11)
```

```
INSERT INTO GoodsOnStorage VALUES (12502, 11, '2020-01-03', 'kf', 12, '4', 90, 10)
```

```
INSERT INTO GoodsOnStorage VALUES (12553, 110, '2020-01-03', 'kf', 12, '4', 30, 30)
```

```
INSERT INTO Orders VALUES (12499, 10, 'df', 15, '2020-01-02',  
'2020-01-03')
```

```
INSERT INTO Orders VALUES (1259, 11, 'kf', 4, '2020-01-03', '2020-01-04')
```

```
INSERT INTO Orders VALUES (12499, 11, 'kf', 15, '2020-01-03',  
'2020-01-04')
```

### ЗАДАНИЕ 1

```
SELECT d.remainderOfGoods*productunitweight, d.remainderOfGoods, g.articleNumber, g.name
```

```
FROM Goods g, goodsonstorage d
```

```
WHERE (g.articleNumber = d.articleNumberOfGoods)
```

### ЗАДАНИЕ 2

```
SELECT goods, customer
```

```
FROM Orders
```

```
JOIN Goodsonstorage f ON (Orders.goods = f.articleNumberOfGoods)
```

### ЗАДАНИЕ 3

1. Коррелированный подзапрос:

```
SELECT *
```

```
FROM SuppliersCustomers c
```

```
WHERE NOT EXISTS (SELECT * FROM GoodsOnStorage g WHERE g.provider = c.id)
```

2. Некоррелированный подзапрос:

```
SELECT *  
FROM SuppliersCustomers c  
WHERE c.id NOT IN (SELECT provider FROM GoodsOnStorage)
```

Коррелированный медленнее, так как он пробегает по каждой строке. Некоррелированный выполняется один раз.

#### ЗАДАНИЕ 4

```
SELECT *  
  
FROM Goods g  
  
WHERE (g.articleNumber IN (SELECT articleNumberOfGoods FROM GoodsOnStorage g WHERE  
g.quantityOfReceivedProduct = g.remainderOfGoods)) and (g.articleNumber NOT IN (SELECT goods FROM  
Orders))
```

#### ЗАДАНИЕ 5

```
SELECT c.quantity, e.remainderOfGoods,  
  
CASE  
    WHEN c.quantity < e.remainderOfGoods THEN 'ENOUGH'  
    ELSE 'NOT ENOUGH'  
  
END  
  
FROM orders c, goodsonstorage e  
  
WHERE c.goods = e.articleNumberOfGoods
```