

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»

---

Кафедра вычислительной техники

---



**НГТУ  
НЭТИ**

**Курсовая работа  
По дисциплине  
«Базы данных»**

Тема: Разработка базы данных «Снаряжение для горного туризма»

Студент Павлюк А.С. (вариант 163)

Группа: АВТ-008

Преподаватель: Трошина Галина Васильевна

Новосибирск, 2022

## Оглавление

Задание .....	3
Введение.....	5
Графическое представление связей между таблицами.....	6
Структура таблиц базы данных .....	6
Содержание таблиц базы данных .....	7
Руководство пользователя .....	10
Тексты запросов и примеры работы.....	11
Анализ результатов и выводы.....	33
Список использованных источников .....	34

## Задание

Сформировать несколько таблиц. Предусмотреть ввод данных, редактирование, просмотр данных. Обязательные требования к базе данных: наличие таблиц-справочников и таблиц, использующих справочники; предусмотреть следующие роли: оператор базы данных, пользователь базы данных, администратор БД. Реализовать следующие типы запросов с использованием нескольких таблиц:

- Для каждого снаряжения для горного туризма указать сведения о нем (наименование, дату выпуска, поставщик, цена, дата продажи и т.п.).
- Получить список, отсортированный: по дате выпуска, в алфавитном порядке по поставщику, по стоимости, по дате продажи.
- Найти самое дорогое снаряжение для горного туризма, самое дешевое, среднюю стоимость.
- Найти снаряжение для горного туризма с ценой в заданных пределах (предусмотреть ввод цены с клавиатуры).
- Найти все снаряжение для горного туризма заданного производителя (выбор производителя).
- Найти долю дешевого снаряжения для горного туризма (меньше заданного, ввод ограничения) от общего числа снаряжения для горного туризма.
- Найти все снаряжение для горного туризма с заданной датой выпуска (ввод даты).
- Найти все снаряжение для горного туризма, чья дата продажи находится в заданных пределах (ввод интервала) для заданного производителя (ввод производителя) и в целом.
- Найти долю снаряжения для горного туризма, проданного за определенный период (ввод периода) от общего времени продажи.
- Найти самое популярное снаряжение для горного туризма (продано наибольшее количество).
- Найти все снаряжение для горного туризма, поступившие от заданного поставщика (выбор поставщика), чья стоимость больше, чем средняя стоимость снаряжения для горного туризма, поступившего из заданной страны (выбор страны).
- Найти долю дорогого снаряжения для горного туризма (чья стоимость больше заданной стоимости), поступившего от заданного поставщика (выбор поставщика) и в целом.
- Найти среднюю стоимость снаряжения для горного туризма, проданного за определенный промежуток времени (ввод интервала).

- Найти все снаряжение для горного туризма, чья стоимость выше, чем средняя стоимость снаряжения для горного туризма заданного производителя (выбор производителя).
- Определить долю регулярных поставок снаряжения для горного туризма от общего числа снаряжения для горного туризма.
- Найти объем продаж снаряжения для горного туризма за месяц, за квартал, за год; за этот же период: найти среднюю цену, самое дорогое снаряжение для горного туризма, самое дешевое снаряжение для горного туризма.

## Введение

В качестве инструмента для проектирования базы данных выбрана одна из самых популярных СУБД PostgreSQL.

СУБД отличается высокой надёжностью и хорошей производительностью. PostgreSQL поддерживает транзакции (ACID). При этом система расширяемая — можно создавать свои типы данных и индексов, а также расширять поведение при помощи языков программирования.

В рамках выполнения курсовой работы необходимо разработать структуру базы данных «Снаряжение для горного туризма», реализовать типовые запросы с использованием нескольких таблиц, оценить полученные результаты.

Цель работы – обобщить и структурировать знания, полученные в рамках дисциплины «Базы данных», научиться применять современные технологии создания баз данных.

## Графическое представление связей между таблицами

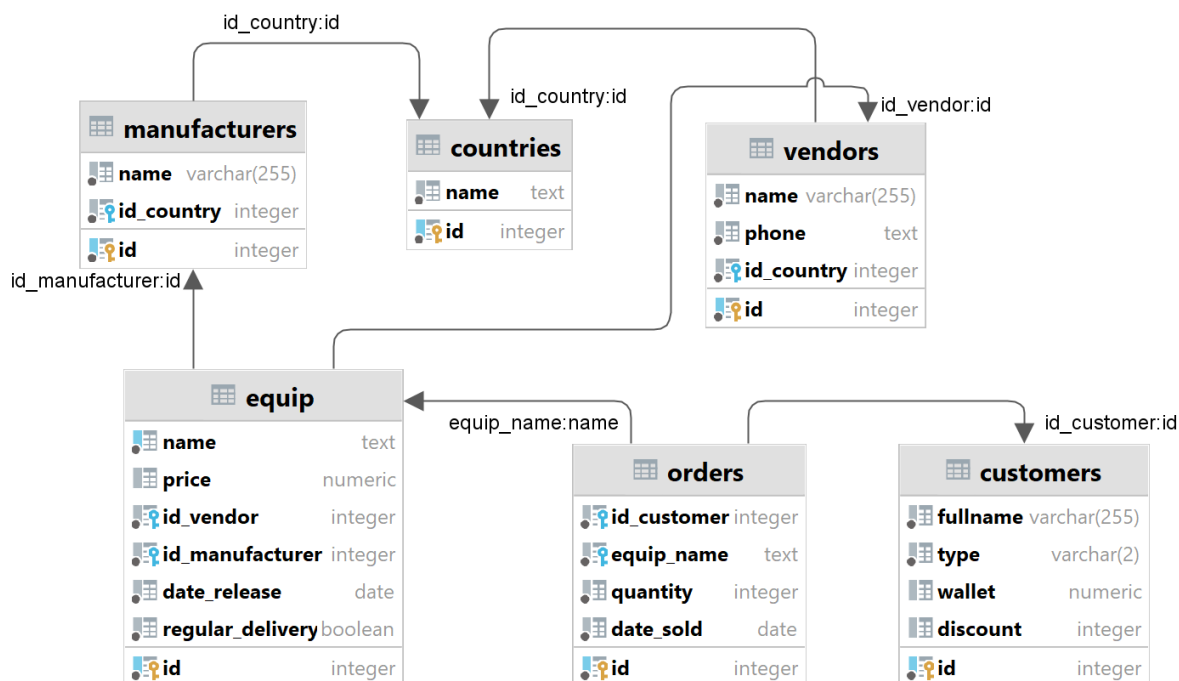


Рисунок 1. Структура базы данных "Снаряжение для горного туризма"

### Структура таблиц базы данных

Таблица-справочник **countries** – список стран:

- id – идентификатор страны
- name – название страны

Таблица-справочник **manufacturers** – список производителей:

- id – идентификатор производителя
- name – название компании производителя
- id\_country – идентификатор страны производителя

Таблица-справочник **vendors** – список поставщиков:

- id – идентификатор поставщика
- name – название компании поставщика
- phone – телефон компании поставщика
- id\_country – идентификатор страны поставщика

Таблица **customers** – информация о покупателях:

- id – идентификатор покупателя
- fullname – имя покупателя
- type – тип покупателя
- wallet – количество денег на счету покупателя
- discount – скидка для покупателя

Таблица **equip** – информация о снаряжении для горного туризма:

- id – идентификатор снаряжения
- price – цена снаряжения
- id\_vendor – идентификатор поставщика
- id\_manufacturer – идентификатор производителя
- date\_release – дата привоза снаряжения
- regular\_delivery – флаг регулярных доставок

Таблица **orders** – информация о заказах:

- id – идентификатор заказа
- id\_customer – идентификатор покупателя
- equip\_name – название снаряжения
- quantity – количество снаряжения
- date\_sold – дата продажи

Содержание таблиц базы данных

- база данных tour\_equip

```
tour_equip=# \dt
```

Список отношений			
Схема	Имя	Тип	Владелец
public	countries	таблица	nstu
public	customers	таблица	nstu
public	equip	таблица	nstu
public	manufacturers	таблица	nstu
public	orders	таблица	nstu
public	vendors	таблица	nstu

(6 строк)

Рисунок 2. Список таблиц базы данных

- countries

```
tour_equip=# select * from countries;
```

id	name
1	Россия
2	Китай
3	США
4	Канада
5	Япония
6	Беларусь
7	Тайвань
8	Нидерланды

(8 строк)

Рисунок 3. Содержание таблицы стран

- manufacturers

```
tour equip=# select * from manufacturers;
id |          name          | id_country
---+-----+-----+
 1 | Alpinestars            |          7
 2 | Acerbis                |          3
 3 | Daytona                |          5
 4 | MadBull                |          7
 5 | INFLAME                |          4
 6 | BLAUER                 |          7
 7 | Hein Gericke           |          4
 8 | IXS Motorcycle Fashion |          8
 9 | SPYKE                  |          1
10 | Fox Head Inc           |          7
(10 строк)
```

Рисунок 4. Содержание таблицы производителей

- vendors

```
tour equip=# select * from vendors;
id |      name      |      phone      | id_country
---+-----+-----+
 1 | 000 РЕД СКИЛЛ | 89124216754 |          7
 2 | Nrg Sport     | 1238967   |          8
 3 | Deporte Shop  | 9998877   |          7
 4 | Фабрика 6F   | 89538892931 |          1
 5 | Sixteen Tons | 2121212   |          6
 6 | 000 Зубрава  | 8120735   |          8
 7 | BEST ORIGINAL | 89133334455 |          6
 8 | ПО Кобра     | 1234567   |          7
 9 | Pro Equipment | 5555555   |          1
10 | HighMount    | 6589292   |          2
(10 строк)
```

Рисунок 5. Содержание таблицы поставщиков

- customers

```
tour equip=# select * from customers;
id |      fullname      | type | wallet | discount
---+-----+-----+
 2 | Петров А.К.        | ФЛ   | 10200 |          0
 3 | Овиченко П.Ф.      | ФЛ   |   400 |          0
 4 | Наминов Р.Д.       | ФЛ   |  7032 |          0
 5 | Фратонов К.Л.      | ФЛ   |  1000 |          0
 6 | ИП Иванова А.У.    | ЮЛ   | 230000 |         10
 7 | Иванов С.А.        | ФЛ   |  21040 |          0
 8 | Кулякова Р.И.      | ФЛ   |   6700 |          0
 9 | Сидоров А.А.       | ФЛ   |  10000 |          0
10 | 000 Азимут         | ЮЛ   | 142000 |         10
11 | Пригнак А.К.       | ФЛ   |  29030 |          0
12 | Уров Г.Н.          | ФЛ   | 100000 |          0
13 | Аникин Б.Ю.        | ФЛ   |   8350 |          0
14 | 000 Компания БКС   | ЮЛ   | 534000 |         10
(13 строк)
```

Рисунок 6. Содержание таблицы покупателей



- equip

```
tour Equip=# select * from equip order by id;
```

id	name	price	id_vendor	id_manufacturer	date_release	regular_delivery
1	Рюкзак	7304	4	2	2022-01-20	t
2	Палатка	4275	4	10	2022-01-21	t
3	Спальный мешок	8368	3	9	2022-02-11	f
4	Сидушка	1940	7	3	2022-02-12	t
5	Палки trekkingовые	9082	5	6	2022-03-22	f
6	Маленький рюкзак для городов и радиалок	5602	5	7	2022-03-23	f
7	Штаны	9943	7	6	2022-04-13	t
8	Термобелье	486	5	5	2022-04-14	f
9	Флисовый батник	8176	3	6	2022-05-24	f
10	Дождевик	8104	4	6	2022-05-25	t
11	Панама	3544	6	3	2022-06-15	f
12	Защитные перчатки	9821	5	5	2022-06-16	f
13	Белье	1096	3	8	2022-07-26	f
14	Tреккинговые ботинки	8017	4	5	2022-07-27	t
15	Кроссовки trekkingовые	3825	4	2	2022-08-17	t
16	сланци для бивака	4982	5	2	2022-08-18	f
17	Солнцезащитные очки	8606	7	4	2022-09-28	t
18	Солнцезащитный крем	1294	5	7	2022-09-29	f
19	Средства личной гигиены	5433	2	2	2022-10-19	f
20	Набор одноразовой посуды	8858	2	6	2022-10-10	f
21	Фонарик	5752	2	9	2022-11-20	f
22	Личная аптечка	4476	4	1	2022-11-21	t
23	Герметичные пакеты	1216	7	8	2022-12-31	t
24	Газовая горелка	145	7	3	2022-12-30	t

(24 строки)

Рисунок 7. Содержание таблицы снаряжения

- orders

id	id_customer	equip_name	quantity	date_sold
1	7	Рюкзак	3	2022-03-21
2	7	Палатка	5	2022-03-22
3	10	Спальный мешок	4	2022-04-23
4	8	Сидушка	4	2022-04-24
5	10	Палки trekkingовые	3	2022-04-15
6	9	Маленький рюкзак для городов и радиалок	5	2022-04-16
7	11	Штаны	2	2022-05-17
8	12	Термобелье	4	2022-05-18
9	13	Флисовый батник	5	2022-05-31
10	10	Дождевик	2	2022-06-30
11	8	Панама	5	2022-07-29
12	10	Защитные перчатки	1	2022-07-30
13	11	Белье	4	2022-08-19
14	10	Tреккинговые ботинки	4	2022-08-28
15	5	Кроссовки trekkingовые	3	2022-09-17
16	3	Солнцезащитные очки	1	2022-10-25
17	9	Солнцезащитный крем	4	2022-10-24
18	14	Средства личной гигиены	2	2022-10-23
19	9	Набор одноразовой посуды	2	2022-10-12
20	3	Фонарик	3	2022-11-29
21	4	Личная аптечка	2	2022-11-30
22	7	Герметичные пакеты	5	2023-01-01
23	7	Газовая горелка	5	2022-12-31
24	2	Рюкзак	3	2023-01-13
25	11	Палатка	5	2023-01-12
26	12	Спальный мешок	1	2023-02-14
27	7	Сидушка	1	2023-02-15
28	11	Палки trekkingовые	2	2023-03-27
29	10	Маленький рюкзак для городов и радиалок	4	2023-03-20
30	3	Штаны	3	2023-03-21
31	11	Дождевик	3	2023-04-22
32	14	Фонарик	2	2023-04-13
33	11	Газовая горелка	4	2023-04-14
34	8	Средства личной гигиены	4	2023-05-15
35	7	Панама	5	2023-05-16

-- Далее --

Рисунок 8. Содержание таблицы заказов

Таблица 1. Список функций

Функция	Входные параметры	Результат
get_data()	-	Таблица со сведениями о снаряжении
sort_date_release()	-	Таблица всех товаров, отсортированных по дате выпуска
sort_vendor()	-	Таблица всех товаров, отсортированных по названию поставщика
sort_price()	-	Таблица всех товаров, отсортированных по стоимости
sort_date_sold()	-	Таблица всех товаров, отсортированных по дате продажи
max_price()	-	Информация о самом дорогом снаряжении
min_price()	-	Информация о самом дешевом снаряжении
avg_price()	-	Средняя стоимость снаряжения
between_price(numeric, numeric)	Нижняя и верхняя граница цены	Таблица снаряжения с ценой в заданных пределах
get_by_vendor(varchar)	Производитель	Таблица снаряжения заданного производителя
get_less_than(numeric)	Заданная стоимость	Таблица снаряжения, чья стоимость меньше заданной
get_by_release(date)	Дата выпуска	Таблица снаряжения с заданной датой выпуска
by_date(date, date)	Нижняя и верхняя граница даты продажи	Таблица снаряжения, проданного за определенный период
by_interval(interval)	Интервал	Таблица снаряжения, чья дата продажи находится в заданном интервале в целом
by_date_and_manuf(interval, varchar)	Интервал, производитель	Таблица снаряжения заданного производителя, чья дата продажи находится в заданном интервале
most_popular()	-	Самое популярное снаряжение (больше всего продано)
by_vendor_with_price(varchar, text)	Поставщик, название страны	Таблица снаряжения, поступившее от заданного поставщика, чья стоимость больше, чем средняя стоимость снаряжения, поступившего из заданной страны
greater_than(numeric)	Заданная стоимость	Таблица снаряжения, чья стоимость больше заданной
greater_than(numeric, varchar)	Заданная стоимость,	Таблица снаряжения,

	поставщик	поступившего от заданного поставщика, чья стоимость больше заданной
avg_price_by_date(date, date)	Нижняя и верхняя граница даты продажи	Средняя стоимость снаряжения, проданного за определенный период времени
price_greater_than_avg_manufac (text)	Производитель	Таблица снаряжения, чья стоимость выше, чем средняя стоимость снаряжения заданного производителя
is_regular_delivery()	-	Таблица снаряжения с регулярными поставками
count_with_price_status(integer)	Число месяцев	Количество и средняя стоимость снаряжения, проданного за указанное число месяцев
equip_with_price_by_period (integer, text)	Число месяцев, параметр (max, min)	Самое дорогое/дешевое снаряжение, проданное за указанное число месяцев

### Тексты запросов и примеры работы

Для каждого снаряжения для горного туризма указать сведения о нем (наименование, дату выпуска, поставщик, цена, дата продажи и т.п.).

create or replace function get\_data() returns table (

id integer,

name text,

release date,

vendor varchar(255),

manufacturer varchar(255),

price numeric,

sold date

) as \$\$ BEGIN RETURN QUERY

select e.id,

e.name,

e.date\_release as release,

v.name as vendor,

m.name as manufacturer,

e.price,

o.date\_sold as sold

from equip e

join vendors v on v.id = e.id\_vendor

```

join manufacturers m on m.id = e.id_manufacturer
join orders o on o.equip_name = e.name;
END;
$$ language plpgsql;

```

tour_equip=# select * from get_data();						
id	name	release	vendor	manufacturer	price	sold
1	Рюкзак	2022-01-20	Фабрика 6F	Acerbis	7304	2022-03-21
2	Палатка	2022-01-21	Фабрика 6F	Fox Head Inc	4275	2022-03-22
3	Спальный мешок	2022-02-11	Deporte Shop	SPYKE	8368	2022-04-23
4	Сидушка	2022-02-12	BEST ORIGINAL	Daytona	1940	2022-04-24
5	Палки trekkingовые	2022-03-22	Sixteen Tons	BLAUER	9082	2022-04-15
6	Маленький рюкзак для городов и радиалок	2022-03-23	Sixteen Tons	Hein Gericke	5602	2022-04-16
7	Штаны	2022-04-13	BEST ORIGINAL	BLAUER	9943	2022-05-17
8	Термобелье	2022-04-14	Sixteen Tons	INFLAME	486	2022-05-18
9	Флисовый батник	2022-05-24	Deporte Shop	BLAUER	8176	2022-05-31
10	Дождевик	2022-05-25	Фабрика 6F	BLAUER	8104	2022-06-30
11	Панама	2022-06-15	000 Зубрава	Daytona	3544	2022-07-29
12	Защитные перчатки	2022-06-16	Sixteen Tons	INFLAME	9821	2022-07-30
13	Белье	2022-07-26	Deporte Shop	IXS Motorcycle Fashion	1096	2022-08-19
14	Trekkingовые ботинки	2022-07-27	Фабрика 6F	INFLAME	8017	2022-08-28
15	Кроссовки trekkingовые	2022-08-17	Фабрика 6F	Acerbis	3825	2022-09-17
17	Солнцезащитные очки	2022-09-28	BEST ORIGINAL	MadBull	8606	2022-10-25
18	Солнцезащитный крем	2022-09-29	Sixteen Tons	Hein Gericke	1294	2022-10-24
19	Средства личной гигиены	2022-10-19	Nrg Sport	Acerbis	5433	2022-10-23
20	Набор одноразовой посуды	2022-10-10	Nrg Sport	BLAUER	8858	2022-10-12
21	Фонарик	2022-11-20	Nrg Sport	SPYKE	5752	2022-11-29
22	Личная аптечка	2022-11-21	Фабрика 6F	Alpinestars	4476	2022-11-30
23	Герметичные пакеты	2022-12-31	BEST ORIGINAL	IXS Motorcycle Fashion	1216	2023-01-01
24	Газовая горелка	2022-12-30	BEST ORIGINAL	Daytona	145	2022-12-31
1	Рюкзак	2022-01-20	Фабрика 6F	Acerbis	7304	2023-01-13
2	Палатка	2022-01-21	Фабрика 6F	Fox Head Inc	4275	2023-01-12
3	Спальный мешок	2022-02-11	Deporte Shop	SPYKE	8368	2023-02-14
4	Сидушка	2022-02-12	BEST ORIGINAL	Daytona	1940	2023-02-15
5	Палки trekkingовые	2022-03-22	Sixteen Tons	BLAUER	9082	2023-03-27
6	Маленький рюкзак для городов и радиалок	2022-03-23	Sixteen Tons	Hein Gericke	5602	2023-03-20
7	Штаны	2022-04-13	BEST ORIGINAL	BLAUER	9943	2023-03-21
10	Дождевик	2022-05-25	Фабрика 6F	BLAUER	8104	2023-04-22
21	Фонарик	2022-11-20	Nrg Sport	SPYKE	5752	2023-04-13
24	Газовая горелка	2022-12-30	BEST ORIGINAL	Daytona	145	2023-04-14
19	Средства личной гигиены	2022-10-19	Nrg Sport	Acerbis	5433	2023-05-15
11	Панама	2022-06-15	000 Зубрава	Daytona	3544	2023-05-16

Рисунок 9. Демонстрация работы функции get\_data()

Получить список, отсортированный:

- по дате выпуска,
- в алфавитном порядке по поставщику,
- по стоимости,
- по дате продажи.
- по дате выпуска:

create or replace function sort\_date\_release() returns table (

id integer,

name text,

price numeric,

id\_vendor integer,

id\_manufacturer integer,

```

        date_release date
    ) as $$ begin return query
select *
from equip
order by date_release;
end;
$$ language plpgsql;

```

```
tour_equip=# select * from sort_date_release();
```

id	name	price	id_vendor	id_manufacturer	date_release	regular_delivery
1	Рюкзак	7304	4	2	2022-01-20	t
2	Палатка	4275	4	10	2022-01-21	t
3	Спальный мешок	8368	3	9	2022-02-11	f
4	Сидушка	1940	7	3	2022-02-12	t
5	Палки трекинговые	9082	5	6	2022-03-22	f
6	Маленький рюкзак для городов и радиалок	5602	5	7	2022-03-23	f
7	Штаны	9943	7	6	2022-04-13	t
8	Термобелье	486	5	5	2022-04-14	f
9	Флисовый батник	8176	3	6	2022-05-24	f
10	Дождевик	8104	4	6	2022-05-25	t
11	Панама	3544	6	3	2022-06-15	f
12	Защитные перчатки	9821	5	5	2022-06-16	f
13	Белье	1096	3	8	2022-07-26	f
14	Трекинговые ботинки	8017	4	5	2022-07-27	t
15	Кроссовки трекинговые	3825	4	2	2022-08-17	t
16	сланци для бивака	4982	5	2	2022-08-18	f
17	Солнцезащитные очки	8606	7	4	2022-09-28	t
18	Солнцезащитный крем	1294	5	7	2022-09-29	f
20	Набор одноразовой посуды	8858	2	6	2022-10-10	f
19	Средства личной гигиены	5433	2	2	2022-10-19	f
21	Фонарик	5752	2	9	2022-11-20	f
22	Личная аптечка	4476	4	1	2022-11-21	t
24	Газовая горелка	145	7	3	2022-12-30	t
23	Герметичные пакеты	1216	7	8	2022-12-31	t

(24 строки)

Рисунок 10. Демонстрация работы функции sort\_date\_release()

- в алфавитном порядке по поставщику:

```

create or replace function sort_vendor() returns table (
    id integer,
    name text,
    price numeric,
    vendor varchar(255)
) as $$ begin return query
select e.id,
       e.name,
       e.price,
       v.name
from equip e
       join vendors v on e.id_vendor = v.id
order by v.name;

```

end;

\$\$ language plpgsql;

```
tour Equip=# select * from sort_vendor();
```

id	name	price	vendor
7	Штаны	9943	BEST ORIGINAL
4	Сидушка	1940	BEST ORIGINAL
23	Герметичные пакеты	1216	BEST ORIGINAL
17	Солнцезащитные очки	8606	BEST ORIGINAL
24	Газовая горелка	145	BEST ORIGINAL
3	Спальный мешок	8368	Deporte Shop
9	Флисовый батник	8176	Deporte Shop
13	Белье	1096	Deporte Shop
21	Фонарик	5752	Nrg Sport
19	Средства личной гигиены	5433	Nrg Sport
20	Набор одноразовой посуды	8858	Nrg Sport
16	сланци для бивака	4982	Sixteen Tons
18	Солнцезащитный крем	1294	Sixteen Tons
12	Защитные перчатки	9821	Sixteen Tons
8	Термобелье	486	Sixteen Tons
6	Маленький рюкзак для городов и радиалок	5602	Sixteen Tons
5	Палки трекинговые	9082	Sixteen Tons
11	Панама	3544	000 Зубрава
14	Трекинговые ботинки	8017	Фабрика 6F
15	Кроссовки трекинговые	3825	Фабрика 6F
10	Дождевик	8104	Фабрика 6F
2	Палатка	4275	Фабрика 6F
22	Личная аптечка	4476	Фабрика 6F
1	Рюкзак	7304	Фабрика 6F

(24 строки)

Рисунок 11. Демонстрация работы функции sort\_vendor()

- по стоимости:

create or replace function sort\_price() returns table (

id integer,

name text,

price numeric,

id\_vendor integer,

id\_manufacturer integer,

date\_release date

) as \$\$ begin return query

select \*

from equip

order by price;

end;

\$\$ language plpgsql;

```
tour_equip=# select * from sort_price();
```

id	name	price	id_vendor	id_manufacturer	date_release	regular_delivery
24	Газовая горелка	145	7	3	2022-12-30	t
8	Термобелье	486	5	5	2022-04-14	f
13	Белье	1096	3	8	2022-07-26	f
23	Герметичные пакеты	1216	7	8	2022-12-31	t
18	Солнцезащитный крем	1294	5	7	2022-09-29	f
4	Сидушка	1940	7	3	2022-02-12	t
11	Панама	3544	6	3	2022-06-15	f
15	Кроссовки трекинговые	3825	4	2	2022-08-17	t
2	Палатка	4275	4	10	2022-01-21	t
22	Личная аптечка	4476	4	1	2022-11-21	t
16	сланци для бивака	4982	5	2	2022-08-18	f
19	Средства личной гигиены	5433	2	2	2022-10-19	f
6	Маленький рюкзак для городов и радиалок	5602	5	7	2022-03-23	f
21	Фонарик	5752	2	9	2022-11-20	f
1	Рюкзак	7304	4	2	2022-01-20	t
14	Трекинговые ботинки	8017	4	5	2022-07-27	t
10	Дождевик	8104	4	6	2022-05-25	t
9	Флисовый батник	8176	3	6	2022-05-24	f
3	Спальный мешок	8368	3	9	2022-02-11	f
17	Солнцезащитные очки	8606	7	4	2022-09-28	t
20	Набор одноразовой посуды	8858	2	6	2022-10-10	f
5	Палки трекинговые	9082	5	6	2022-03-22	f
12	Защитные перчатки	9821	5	5	2022-06-16	f
7	Штаны	9943	7	6	2022-04-13	t

(24 строки)

Рисунок 12. Демонстрация работы функции sort\_price()

- по дате продажи:

create or replace function sort\_date\_sold() returns table (

id integer,

name text,

price numeric,

release date

) as \$\$ begin return query

select e.id,

e.name,

e.price,

o.date\_sold

from equip e

join orders o on e.name = o.equip\_name

order by o.date\_sold;

end;

\$\$ language plpgsql;

```
tour Equip=# select * from sort_date_sold();
```

id	name	price	sold
1	Рюкзак	7304	2022-03-21
2	Палатка	4275	2022-03-22
5	Палки треккинговые	9082	2022-04-15
6	Маленький рюкзак для городов и радиалок	5602	2022-04-16
3	Спальный мешок	8368	2022-04-23
4	Сидушка	1940	2022-04-24
7	Штаны	9943	2022-05-17
8	Термобелье	486	2022-05-18
9	Флисовый батник	8176	2022-05-31
10	Дождевик	8104	2022-06-30
11	Панама	3544	2022-07-29
12	Защитные перчатки	9821	2022-07-30
13	Белье	1096	2022-08-19
14	Треккинговые ботинки	8017	2022-08-28
15	Кроссовки треккинговые	3825	2022-09-17
20	Набор одноразовой посуды	8858	2022-10-12
19	Средства личной гигиены	5433	2022-10-23
18	Солнцезащитный крем	1294	2022-10-24
17	Солнцезащитные очки	8606	2022-10-25
21	Фонарик	5752	2022-11-29
22	Личная аптечка	4476	2022-11-30
24	Газовая горелка	145	2022-12-31
23	Герметичные пакеты	1216	2023-01-01
2	Палатка	4275	2023-01-12
1	Рюкзак	7304	2023-01-13
3	Спальный мешок	8368	2023-02-14
4	Сидушка	1940	2023-02-15
6	Маленький рюкзак для городов и радиалок	5602	2023-03-20
7	Штаны	9943	2023-03-21
5	Палки треккинговые	9082	2023-03-27
21	Фонарик	5752	2023-04-13
24	Газовая горелка	145	2023-04-14
10	Дождевик	8104	2023-04-22

Рисунок 13. Демонстрация работы функции `sort_date_sold()`

Найти самое дорогое снаряжение для горного туризма, самое дешевое, среднюю стоимость.

- самое дорогое:

create or replace function `max_price()` returns table (

id integer,

name text,

price numeric,

id\_vendor integer,

id\_manufacturer integer,

date\_release date,

regular\_delivery boolean

) as \$\$ begin return query

select \*

from equip e

where e.price = (



```

select MAX(e.price)
from equip e
);
end;
$$ language plpgsql;

```

```

tour_equip=# select * from max_price();
 id | name  | price | id_vendor | id_manufacturer | date_release | regular_delivery 
-----+-----+-----+-----+-----+-----+-----
  7 | Штаны |  9943 |         7 |                6 | 2022-04-13   | t
(1 строка)

```

*Рисунок 14. Демонстрация работы функции max\_price()*

- самое дешевое:

```

create or replace function min_price() returns table (
    id integer,
    name text,
    price numeric,
    id_vendor integer,
    id_manufacturer integer,
    date_release date,
    regular_delivery boolean
) as $$ begin return query
select *
from equip e
where e.price = (
    select MIN(e.price)
    from equip e
);
end;
$$ language plpgsql;

```

```

tour_equip=# select * from min_price();
 id | name           | price | id_vendor | id_manufacturer | date_release | regular_delivery 
-----+-----+-----+-----+-----+-----+-----
 24 | Газовая горелка |   145 |         7 |                3 | 2022-12-30   | t
(1 строка)

```

*Рисунок 15. Демонстрация работы функции min\_price()*

- среднюю стоимость:

```
create or replace function avg_price() returns numeric language sql as $$
select AVG(e.price)
from equip e; $$;
```

```
tour_equip=# select * from avg_price();
          avg_price
-----
 5431.041666666666667
(1 строка)
```

Рисунок 16. Демонстрация работы функции avg\_price()

Найти снаряжение для горного туризма с ценой в заданных пределах (предусмотреть ввод цены с клавиатуры).

```
create or replace function between_price(min numeric, max numeric) returns table (
```

```
    id integer,
    name text,
    price numeric,
    id_vendor integer,
    id_manufacturer integer,
    date_release date,
    regular_delivery boolean
```

```
) as $$ begin return query
```

```
select *
```

```
from equip e
```

```
where e.price between min and max;
```

```
end;
```

```
$$ language plpgsql;
```

```
tour_equip=# select * from between_price(3000, 5000);
 id |          name          | price | id_vendor | id_manufacturer | date_release | regular_delivery
-----+-----+-----+-----+-----+-----+-----
 11 | Панама                 | 3544  | 6         | 3               | 2022-06-15  | f
 16 | сланцы для бивака     | 4982  | 5         | 2               | 2022-08-18  | f
 2  | Палатка               | 4275  | 4         | 10              | 2022-01-21  | t
 15 | Кроссовки трекинговые | 3825  | 4         | 2               | 2022-08-17  | t
 22 | Личная аптечка       | 4476  | 4         | 1               | 2022-11-21  | t
(5 строк)
```

Рисунок 17. Демонстрация работы функции between\_price()

Найти все снаряжение для горного туризма заданного производителя (выбор производителя).

```

create or replace function get_by_vendor(_vendor varchar(255)) returns table (
    id integer,
    name text,
    price numeric,
    vendor varchar(255)
) as $$ begin return query
select e.id,
       e.name,
       e.price,
       v.name as vendor
from equip e
       join vendors v on v.id = e.id_vendor
where upper(v.name) = upper(_vendor);
end;
$$ language plpgsql;

```

```

tour_equip=# select * from get_by_vendor('nrg sport');
 id |          name          | price | vendor
-----+-----+-----+-----
 19 | Средства личной гигиены |  5433 | Nrg Sport
 20 | Набор одноразовой посуды |  8858 | Nrg Sport
 21 | Фонарик                 |  5752 | Nrg Sport
(3 строки)

```

*Рисунок 18. Демонстрация работы функции get\_by\_vendor()*

Найти долю дешевого снаряжения для горного туризма (меньше заданного, ввод ограничения) от общего числа снаряжения для горного туризма.

```

create or replace function get_less_than(_price numeric) returns table (
    id integer,
    name text,
    price numeric,
    id_vendor integer,
    id_manufacturer integer,
    date_release date,
    regular_delivery boolean
) as $$ begin return query
select *

```

```

from equip e
where e.price < _price;
end;
$$ language plpgsql;

```

```

tour_equip=# select * from get_less_than(5500);
 id |          name          | price | id_vendor | id_manufacturer | date_release | regular_delivery
-----+-----+-----+-----+-----+-----+-----
  8 | Термобелье            |  486 |         5 |                | 2022-04-14   | f
 11 | Панама                 | 3544 |         6 |                | 2022-06-15   | f
 13 | Белье                  | 1096 |         3 |                | 2022-07-26   | f
 16 | сланцы для бивака      | 4982 |         5 |                | 2022-08-18   | f
 19 | Средства личной гигиены | 5433 |         2 |                | 2022-10-19   | f
 18 | Солнцезащитный крем    | 1294 |         5 |                | 2022-09-29   | f
  2 | Палатка                | 4275 |         4 |                | 2022-01-21   | t
  4 | Сидушка                | 1940 |         7 |                | 2022-02-12   | t
 15 | Кроссовки трекинговые  | 3825 |         4 |                | 2022-08-17   | t
 22 | Личная аптечка        | 4476 |         4 |                | 2022-11-21   | t
 23 | Герметичные пакеты     | 1216 |         7 |                | 2022-12-31   | t
 24 | Газовая горелка        |  145 |         7 |                | 2022-12-30   | t
(12 строк)

```

*Рисунок 19. Демонстрация работы функции get\_less\_than()*

Найти все снаряжение для горного туризма с заданной датой выпуска (ввод даты).

```

create or replace function get_by_release(_release date) returns table (

```

```

    id integer,
    name text,
    price numeric,
    id_vendor integer,
    id_manufacturer integer,
    date_release date,
    regular_delivery boolean
) as $$ begin return query

```

```

select *
from equip e
where e.date_release = _release;
end;
$$ language plpgsql;

```

```

tour_equip=# select * from get_by_release('2022-07-26');
 id | name  | price | id_vendor | id_manufacturer | date_release | regular_delivery
-----+-----+-----+-----+-----+-----+-----
 13 | Белье | 1096 |         3 |                | 2022-07-26   | f
(1 строка)

```

*Рисунок 20. Демонстрация работы функции get\_by\_release()*

Найти все снаряжение для горного туризма, чья дата продажи находится в заданных пределах (ввод интервала) для заданного производителя (ввод производителя) и в целом.

- для заданного производителя:

create or replace function by\_date\_and\_manuf(\_interval interval, \_manuf varchar(255)) returns table (

```
    id integer,
    name text,
    date_sold date,
    manufacturer varchar(255)
```

```
) as $$ begin return query
```

```
select e.id,
```

```
    e.name,
```

```
    o.date_sold,
```

```
    m.name as manufacturer
```

```
from equip e
```

```
    join orders o on e.name = o.equip_name
```

```
    join manufacturers m on e.id_manufacturer = m.id
```

```
where now() - o.date_sold <= _interval
```

```
    and now() - o.date_sold >= '1 days'
```

```
    and upper(m.name) = upper(_manuf);
```

```
end;
```

```
$$ language plpgsql;
```

```
tour_equip=# select * from by_date_and_manuf('12 months', 'acerbis');
 id |          name          | date_sold | manufacturer
----+-----+-----+-----
 19 | Средства личной гигиены | 2022-10-23 | Acerbis
   1 | Рюкзак                  | 2022-03-21 | Acerbis
  15 | Кроссовки трекинговые   | 2022-09-17 | Acerbis
(3 строки)
```

*Рисунок 21. Демонстрация работы функции by\_date\_and\_manuf()*

- в целом:

create or replace function by\_interval(\_interval interval) returns table (

```
    id integer,
    name text,
    date_sold date,
    vendor varchar(255)
```

```

) as $$ begin return query
select e.id,
       e.name,
       o.date_sold,
       v.name as vendor
from equip e
      join orders o on e.name = o.equip_name
      join vendors v on e.id_vendor = v.id
where now() - o.date_sold <= _interval
      and now() - o.date_sold >= '1 days';
end;
$$ language plpgsql;

```

```

tour_equip=# select * from by_interval('6 months 2 days');
 id |          name          | date_sold |      vendor
-----+-----+-----+-----
  8 | Термобелье            | 2022-05-18 | Sixteen Tons
  9 | Флисовый батник       | 2022-05-31 | Deporte Shop
 11 | Панама                | 2022-07-29 | 000 Зубрава
 12 | Защитные перчатки     | 2022-07-30 | Sixteen Tons
 13 | Белье                 | 2022-08-19 | Deporte Shop
 19 | Средства личной гигиены | 2022-10-23 | Nrg Sport
 20 | Набор одноразовой посуды | 2022-10-12 | Nrg Sport
 18 | Солнцезащитный крем   | 2022-10-24 | Sixteen Tons
  7 | Штаны                 | 2022-05-17 | BEST ORIGINAL
 10 | Дождевик              | 2022-06-30 | Фабрика 6F
 14 | Треккинговые ботинки  | 2022-08-28 | Фабрика 6F
 15 | Кроссовки треккинговые | 2022-09-17 | Фабрика 6F
 17 | Солнцезащитные очки   | 2022-10-25 | BEST ORIGINAL
(13 строк)

```

*Рисунок 22. Демонстрация работы функции by\_interval()*

Найти долю снаряжения для горного туризма, проданного за определенный период (ввод периода) от общего времени продажи.

```

create or replace function by_date(_start date, _end date) returns table (
    id integer,
    name text,
    date_sold date
) as $$ begin return query
select e.id,

```

```

e.name,
o.date_sold
from equip e
  join orders o on e.name = o.equip_name
where o.date_sold >= _start
      and o.date_sold <= _end;
end;
$$ language plpgsql;

```

```

tour_equip=# select * from by_date('2022-10-01', '2022-12-31');
 id |          name          | date_sold
-----+-----+-----
 19 | Средства личной гигиены | 2022-10-23
 20 | Набор одноразовой посуды | 2022-10-12
 21 | Фонарик                 | 2022-11-29
 18 | Солнцезащитный крем    | 2022-10-24
 17 | Солнцезащитные очки    | 2022-10-25
 22 | Личная аптечка         | 2022-11-30
 24 | Газовая горелка        | 2022-12-31
(7 строк)

```

*Рисунок 23. Демонстрация работы функции by\_date()*

Найти самое популярное снаряжение для горного туризма (продано наибольшее количество).

```

create or replace function most_popular() returns table (name text, quantity bigint) as $$ begin
return query
select e.name,
       sum(o.quantity) as quantity
from orders o
  left join equip e on o.equip_name = e.name
group by e.name
order by quantity desc
limit 1;
end;
$$ language plpgsql;

```

```

tour_equip=# select * from most_popular();
  name  | quantity
-----+-----
Панама |         10
(1 строка)

```

Рисунок 24. Демонстрация работы функции *most\_popular()*

Найти все снаряжение для горного туризма, поступившие от заданного поставщика (выбор поставщика), чья стоимость больше, чем средняя стоимость снаряжения для горного туризма, поступившего из заданной страны (выбор страны).

create or replace function by\_vendor\_with\_price(\_vendor varchar(255), \_country text) returns table (

```

    id integer,
    name text,
    avg_price numeric,
    price numeric,
    vendor varchar(255)

```

) as \$\$

declare \_avg numeric;

begin

select AVG(e.price) into \_avg

from equip e

join vendors v on v.id = e.id\_vendor

join countries c on c.id = v.id\_country

where upper(c.name) = upper(\_country);

return query

select e.id,

e.name,

round(\_avg, 2),

e.price,

v.name as vendor

from equip e

join vendors v on v.id = e.id\_vendor

where upper(v.name) = upper(\_vendor)

and e.price > \_avg;

end;



```
$$ language plpgsql;
```

```
tour_equip=# select * from by_vendor_with_price('best original', 'беларусь');
 id |          name          | avg_price | price |      vendor
-----+-----+-----+-----+-----
  7 | Штаны                  | 4828.82 | 9943 | BEST ORIGINAL
 17 | Солнцезащитные очки | 4828.82 | 8606 | BEST ORIGINAL
(2 строки)
```

Рисунок 25. Демонстрация работы функции `by_vendor_with_price()`

Найти долю дорогого снаряжения для горного туризма (чья стоимость больше заданной стоимости), поступившего от заданного поставщика (выбор поставщика) и в целом.

- от заданного поставщика

```
create or replace function greater_than(_price numeric, _vendor varchar(255)) returns table (
```

```
    id integer,
```

```
    name text,
```

```
    price numeric,
```

```
    vendor varchar(255)
```

```
) as $$ begin return query
```

```
select e.id,
```

```
    e.name,
```

```
    e.price,
```

```
    v.name
```

```
from equip e
```

```
    join vendors v on v.id = e.id_vendor
```

```
where e.price > _price
```

```
    and v.name = _vendor;
```

```
end;
```

```
$$ language plpgsql;
```

```
tour_equip=# select * from greater_than(1500, 'best original');
 id |          name          | price |      vendor
-----+-----+-----+-----
  4 | Сидушка                | 1940 | BEST ORIGINAL
  7 | Штаны                  | 9943 | BEST ORIGINAL
 17 | Солнцезащитные очки | 8606 | BEST ORIGINAL
(3 строки)
```

Рисунок 26. Демонстрация работы функции `greater_than()`

- В целом

create or replace function greater\_than(\_price numeric) returns table (

id integer,

name text,

price numeric

) as \$\$ begin return query

select e.id,

e.name,

e.price

from equip e

where e.price > \_price;

end;

\$\$ language plpgsql;

```
tour_equip=# select * from greater_than(7500);
 id |          name          | price
-----+-----+-----
   3 | Спальный мешок        |  8368
   5 | Палки треккинговые    |  9082
   9 | Флисовый батник       |  8176
  12 | Защитные перчатки     |  9821
  20 | Набор одноразовой посуды |  8858
   7 | Штаны                 |  9943
  10 | Дождевик              |  8104
  14 | Треккинговые ботинки   |  8017
  17 | Солнцезащитные очки   |  8606
(9 строк)
```

Рисунок 27. Демонстрация работы функции greater\_than()

Найти среднюю стоимость снаряжения для горного туризма, проданного за определенный промежуток времени (ввод интервала).

create or replace function avg\_price\_by\_date(\_start date, \_end date) returns real language sql as  
\$\$

select AVG(e.price)

from equip e

join orders o on e.name = o.equip\_name

where o.date\_sold >= \_start

and o.date\_sold <= \_end; \$\$;

```

tour_equip=# select * from avg_price_by_date('2022-11-11', '2023-07-20');
avg_price_by_date
-----
          5067.5625
(1 строка)

```

*Рисунок 28. Демонстрация работы функции avg\_price\_by\_date()*

Найти все снаряжение для горного туризма, чья стоимость выше, чем средняя стоимость снаряжения для горного туризма заданного производителя (выбор производителя).

create or replace function price\_greater\_than\_avg\_manufac(\_manufacturer varchar(255)) returns table (

id integer,

name text,

avg\_price numeric,

price numeric,

manufacturer varchar(255)

) as \$\$

declare \_avg numeric;

begin

select avg(e.price) into \_avg

from equip e

join manufacturers m on e.id\_manufacturer = m.id

where upper(m.name) = upper(\_manufacturer);

return query

select e.id,

e.name,

round(\_avg, 2),

e.price,

m.name as manufacturer

from equip e

join manufacturers m on e.id\_manufacturer = m.id

where e.price > \_avg;

end;

\$\$ language plpgsql;

```
tour_equip=# select * from price_greater_than_avg_manufac('madbull');
 id |          name          | avg_price | price | manufacturer
-----+-----+-----+-----+-----
  5 | Палки треккинговые    | 8606.00 | 9082 | BLAUER
 12 | Защитные перчатки     | 8606.00 | 9821 | INFLAME
 20 | Набор одноразовой посуды | 8606.00 | 8858 | BLAUER
  7 | Штаны                 | 8606.00 | 9943 | BLAUER
(4 строки)
```

Рисунок 29. Демонстрация работы функции `price_greater_than_avg_manufac()`

Определить долю регулярных поставок снаряжения для горного туризма от общего числа снаряжения для горного туризма.

create or replace function `is_regular_delivery()` returns table (

id integer,

name text,

regular\_delivery boolean

) as \$\$ begin return query

select e.id,

e.name,

e.regular\_delivery

from equip e

where e.regular\_delivery = true;

end;

\$\$ language plpgsql;

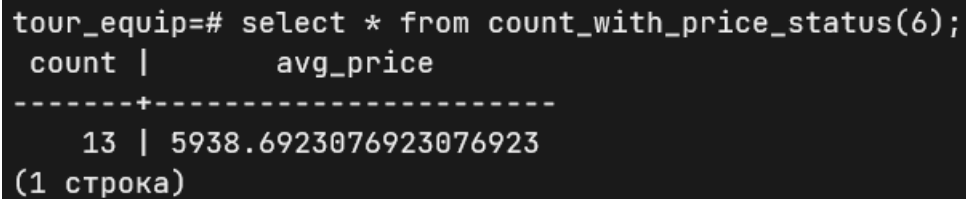
```
tour_equip=# select * from is_regular_delivery();
 id |          name          | regular_delivery
-----+-----+-----
  1 | Рюкзак                 | t
  2 | Палатка                | t
  4 | Сидушка                | t
  7 | Штаны                  | t
 10 | Дождевик               | t
 14 | Треккинговые ботинки   | t
 15 | Кроссовки треккинговые | t
 17 | Солнцезащитные очки   | t
 22 | Личная аптечка        | t
 23 | Герметичные пакеты     | t
 24 | Газовая горелка        | t
(11 строк)
```

Рисунок 30. Демонстрация работы функции `is_regular_delivery()`

Найти объем продаж снаряжения для горного туризма за месяц, за квартал, за год; за этот же период: найти среднюю цену, самое дорогое снаряжение для горного туризма, самое дешевое снаряжение для горного туризма.

- объем продаж и средняя стоимость

```
create or replace function count_with_price_status(_period integer) returns table (  
    count bigint,  
    avg_price numeric  
) as $$ begin return query  
select COUNT(e.id) as volume,  
    AVG(e.price) as avg_price  
from equip e  
    join orders o on e.name = o.equip_name  
where now() - o.date_sold <= _period * interval '1 month'  
    and now() - o.date_sold >= '1 days';  
end;  
$$ language plpgsql;
```



```
tour_equip=# select * from count_with_price_status(6);  
count |      avg_price  
-----+-----  
      13 | 5938.6923076923076923  
(1 строка)
```

Рисунок 31. Демонстрация работы функции count\_with\_price\_status();

- самое дорогое и самое дешевое

```
create or replace function equip_with_price_by_period(_period integer, _property text) returns  
table (  
    id integer,  
    name text,  
    price numeric,  
    date_sold date  
) as $$  
declare _max numeric;  
declare _min numeric;  
begin  
select MAX(e.price) into _max  
from equip e
```

```

    join orders o on e.name = o.equip_name
where now() - o.date_sold <= _period * interval '1 month'
    and now() - o.date_sold >= '1 days';
select MIN(e.price) into _min
from equip e
    join orders o on e.name = o.equip_name
where now() - o.date_sold <= _period * interval '1 month'
    and now() - o.date_sold >= '1 days';
if (upper(_property) like upper('max')) then return query
select e.id,
    e.name,
    e.price,
    o.date_sold
from equip e
    join orders o on e.name = o.equip_name
where e.price = _max
    and now() - o.date_sold <= _period * interval '1 month'
    and now() - o.date_sold >= '1 days';
end if;
if (upper(_property) like upper('min')) then return query
select e.id,
    e.name,
    e.price,
    o.date_sold
from equip e
    join orders o on e.name = o.equip_name
where e.price = _min
    and now() - o.date_sold <= _period * interval '1 month'
    and now() - o.date_sold >= '1 days';
end if;
end;
$$ language plpgsql;

```

```

tour_equip=# select * from equip_with_price_by_period(3, 'max');
 id |          name          | price | date_sold
-----+-----+-----+-----
 20 | Набор одноразовой посуды | 8858  | 2022-10-12
(1 строка)

tour_equip=# select * from equip_with_price_by_period(3, 'min');
 id | name  | price | date_sold
-----+-----+-----+-----
 13 | Белье | 1096  | 2022-08-19
(1 строка)

```

Рисунок 32. Демонстрация работы функции `equip_with_price_by_period()`

Создание ролей «оператор», «пользователь», «администратор».

«администратор» - `CREATE ROLE admin WITH PASSWORD 'admin' LOGIN NOSUPERUSER CREATEROLE CREATEDB;`

Для администратора предоставлена возможность изменять целостность любых таблиц.

`GRANT SELECT, INSERT, UPDATE, DELETE ON countries, customers, equip, manufacturers, orders, vendors TO admin;`

```

PS C:\Users\sasha> psql -U admin -d tour_equip
Пароль пользователя admin:
psql (14.4)
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной
                  страницы Windows (1251).
                  8-битовые (русские) символы могут отображаться некорректно.
                  Подробнее об этом смотрите документацию psql, раздел
                  "Notes for Windows users".
Введите "help", чтобы получить справку.

tour_equip=> \! chcp 1251
Текущая кодовая страница: 1251
tour_equip=> select * from countries;
 id | name
-----+-----
  1 | Россия
  2 | Китай
  3 | США
  4 | Канада
  5 | Япония
  6 | Беларусь
  7 | Тайвань
  8 | Нидерланды
(8 строк)

tour_equip=> insert into countries (name) values ('Франция');
INSERT 0 1
tour_equip=> update countries set name = 'Финляндия' where id = 9;
UPDATE 1
tour_equip=> delete from countries where name like 'Финляндия';
DELETE 1
tour_equip=> |

```

Рисунок 33. Демонстрация операций в роли админа

«оператор» - CREATE ROLE operator WITH PASSWORD 'operator' LOGIN;

Оператору запрещается удалять данные из таблиц. Разрешен просмотр, пополнение и обновление таблиц и таблиц-справочников.

GRANT SELECT, UPDATE, INSERT ON countries, customers, equip, manufacturers, orders, vendors TO operator;

```
PS C:\Users\sasha> psql -U operator -d tour equip
Пароль пользователя operator:
psql (14.4)
Введите "help", чтобы получить справку.

tour equip=> select * from vendors;
 id |   name   |   phone   | id_country
-----+-----+-----+-----
  1 | 000 РЕД СКИЛЛ | 89124216754 |      7
  2 | Nrg Sport | 1238967 |      8
  3 | Deporte Shop | 9998877 |      7
  4 | Фабрика 6F | 89538892931 |      1
  5 | Sixteen Tons | 2121212 |      6
  6 | 000 Зубрава | 8120735 |      8
  7 | BEST ORIGINAL | 89133334455 |      6
  8 | ПО Кобра | 1234567 |      7
  9 | Pro Equipment | 5555555 |      1
 10 | HighMount | 6589292 |      2
(10 строк)

tour equip=> insert into vendors(name, phone, id_country) values ('000 ТЕСТ', '111223', 1);
INSERT 0 1
tour equip=> delete from vendors where id = 11;
ОШИБКА: нет доступа к таблице vendors
tour equip=> |
```

Рисунок 34. Демонстрация команд в роли оператора

«пользователь» - CREATE ROLE user\_db WITH PASSWORD 'user' LOGIN;

Пользователю базы данных разрешен просмотр, обновление, вставка и удаление данных из таблиц.

GRANT SELECT, INSERT, UPDATE, DELETE ON countries, customers, equip, manufacturers, orders, vendors TO user\_db;

```
PS C:\Users\sasha> psql -U user_db -d tour equip
Пароль пользователя user_db:
psql (14.4)
Введите "help", чтобы получить справку.

tour equip=> delete from vendors where id = 11;
DELETE 1
tour equip=> |
```

Рисунок 35. Демонстрация команд в роли пользователя



## Анализ результатов и выводы

В ходе выполнения работы была разработана структура базы данных «Снаряжение для горного туризма» с использованием СУБД PostgreSQL, реализованы типовые запросы с использованием нескольких таблиц, которые были превращены в функции на языке plpgsql для простоты воспроизведения. В базе данных присутствуют таблицы-справочники, а также таблицы, использующие справочники. Графически представлена связь между таблицами. Были предусмотрены роли – оператор, администратор, пользователь.

В результате работы были обобщены и структурированы знания, полученные в рамках дисциплины «Базы данных» и применены современные технологии создания баз данных.

## Список использованных источников

1. Базы данных: методические указания по выполнению курсовой работы для 3 курса факультета автоматики и вычислительной техники по направлению "Информатика и вычислительная техника" / Новосиб. гос. техн. ун-т; [сост.: В. А. Астапчук, В. Б. Панова]. - Новосибирск, 2012. - 60, [1] с.: ил., табл. - Режим доступа: [http://elibrary.nstu.ru/source?bib\\_id=vtls000173910](http://elibrary.nstu.ru/source?bib_id=vtls000173910) (Дата обращения: 01.10.2022)
2. Трошина Г. В. Базы данных [Электронный ресурс]: электронный учебно-методический комплекс / Г. В. Трошина; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: [http://elibrary.nstu.ru/source?bib\\_id=vtls000152242](http://elibrary.nstu.ru/source?bib_id=vtls000152242). - Загл. с экрана – (Дата обращения: 17.10.2022)