МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра вычислительной техники

**

**КУРСОВАЯ РАБОТА**

по дисциплине: «Базы данных»

на тему: «Разработка базы данных «Авиасалон»

Вариант 2

Выполнил:Проверила:

Студент гр. АВТ-917 к.т.н., доцент

Борткевич К.А.Трошина Г.В.

«\_\_\_» \_\_\_\_\_\_ 2022г.«\_\_\_» \_\_\_\_\_\_ 2022 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

Новосибирск

2022

# **ЗАДАНИЕ**

Тема: «Разработка базы данных «Авиасалон». Сформировать несколько таблиц: «Ремонт», «Поставщик запасных частей» и др. 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 базы данных. Реализовать следующие типы запросов с использованием нескольких таблиц:

1. Для каждого вида самолета указать сведения о нем (тип, год выпуска, срок эксплуатации, место изготовление, цена, скорость, вместимость)
2. Для каждого вида самолета выдать список, отсортированный: по году выпуска, в алфавитном порядке все типы, по стоимости, по сроку эксплуатации.
3. Найти долю «старых» самолетов (срок эксплуатации больше заданного, ввод с клавиатуры) от общего числа самолетов.
4. Найти самый дорогой самолет (по каждому типу), самый дешевый, среднюю стоимость.
5. Найти самолеты с ценой свыше 5000 т.р. (и любая другая сумма, т.е. предусмотреть ввод цены с клавиатуры).
6. Найти все самолеты, поступившие из заданной страны (ввод страны), чья скорость больше заданной (ввод скорости).
7. Найти количество самолетов на авиасалоне (по каждому типу и в целом) с заданным сроком эксплуатации (ввод срока).
8. Найти долю самолетов, поступивших из заданной страны (выбор страны) от общего числа самолетов.
9. Найти все самолеты заданной вместимости (ввод) для заданного года выпуска (ввод).
10. Найти количество самолетов заданного года выпуска (ввод), чья стоимость больше, чем средняя стоимость самолетов, поступивших из заданной страны (выбор страны) и заданными ограничениями по скорости (ввод наибольшего и наименьшего значения).
11. Определить количество ремонтируемых самолетов, соотношение подлежащих ремонту самолетов и годных к эксплуатации самолетов (в процентах).
12. Найти вид запасных частей, поступающих для ремонта самолетов отсортировать по наименованиям, по стоимости, по поставщику
13. Найти поставщиков заданного вида запасных частей (выбор) для заданного вида самолетов.

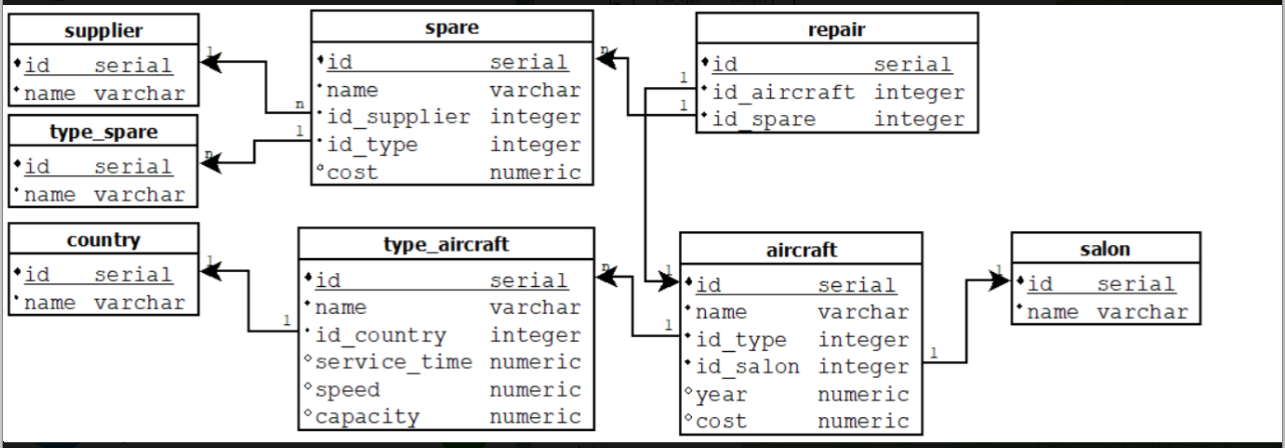
# **ВВЕДЕНИЕ**

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных.

На сегодняшний день любая современная организация не может обойтись без базы данных. Актуальность данной работы связана с существующей необходимостью в хранении, обработке и защите больших объемов финансово-экономической информации. Наиболее эффективным способом решения этой проблемы является автоматизация предметной области средствами СУБД.

Целью работы является построение функционирующей базу данных, используя PostgreSQL, а также улучшение и закрепление практических навыков по работе с языком запросов SQL.

# **ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СВЯЗЕЙ МЕЖДУ ТАБЛИЦАМИ**



*Рисунок 1 – Графическое представление связей между таблицами*

# **СТРУКТУРЫ ТАБЛИЦ БАЗЫ ДАННЫХ**

Таблица-справочник salon содержит информацию о салонах:

Таблица 1 – Структура salon

|  |  |  |  |
| --- | --- | --- | --- |
| Salon | | | |
| Название поля | Тип данных | Ограничения | Комментарий |
| id | serial | primary key, not null | идентификатор |
| name | varchar(40) | - | название страны |

Таблица-справочник aircraft type содержит информацию о видах самолетах:

Таблица 2 – Структура aircraft type

|  |  |  |  |
| --- | --- | --- | --- |
| aircraft type | | | |
| Название поля | Тип данных | Ограничения | Комментарий |
| id | serial | primary key, not null | идентификатор вида самолета |
| name | varchar | - | название вида самолета |
| id\_countre | integer | foreign key, not null | идентификатор страны |
| service\_time | numeric | - | вр |
| speed | numeric | - | скорость самолета |
| capacity | numeric | - | вместимость самолета |

Таблица aircraft содержит информацию о самолетах:

Таблица 3 – Структура aircraft

|  |  |  |  |
| --- | --- | --- | --- |
| Aircraft | | | |
| Название поля | Тип данных | Ограничения | Комментарий |
| id | serial | primary key, not null | идентификатор производителя |
| name | varchar (40) | - | имя производителя |
| id\_type | integer | foreign key, not null | Идентификатор типа самолета |
| id\_salon | integer | foreign key, not null | идентификатор салона |
| year | numeric | - | год выпуска самолета |
| cost | numeric | - | стоимость самолета |

Таблица-справочник country содержит информацию о странах:

Таблица 4 – Структура country

|  |  |  |  |
| --- | --- | --- | --- |
| Country | | | |
| Название поля | Тип данных | Ограничения | Комментарий |
| id | serial | primary key, not null | идентификатор страны |
| name | varchar (40) | - | название страны |

Таблица spare содержит информацию о запчастях:

Таблица 5 – Структура spare

|  |  |  |  |
| --- | --- | --- | --- |
| Spare | | | |
| Название поля | Тип данных | Ограничения | Комментарий |
| id | serial | primary key, not null | идентификатор запчасти |
| name | varchar (40) | - | наименование запчасти |
| id\_supplier | integer | foreign key, not null | идентификатор производителя |
| id\_type | integer | foreign key, not null | идентификатор вида запчасти |
| cost | numeric | - | стоимость запчасти |

Таблица repair содержит информацию о ремонте:

Таблица 6 – Структура repair

|  |  |  |  |
| --- | --- | --- | --- |
| Repair | | | |
| Название поля | Тип данных | Ограничения | Комментарий |
| id | serial | primary key, not null | идентификатор ремонтной роботы |
| id\_aircraft | integer | foreign key, not null | идентификатор самолета |
| id\_spare | integer | foreign key, not null | идентификатор запчасти |

Таблица type\_spare содержит информацию о видах запчастей:

Таблица 7 – Структура type\_spare

|  |  |  |  |
| --- | --- | --- | --- |
| type\_spare | | | |
| Название поля | Тип данных | Ограничения | Комментарий |
| id | serial | primary key, not null | идентификатор вида запчасти |
| name | integer | foreign key, not null | наименование вида запчасти |

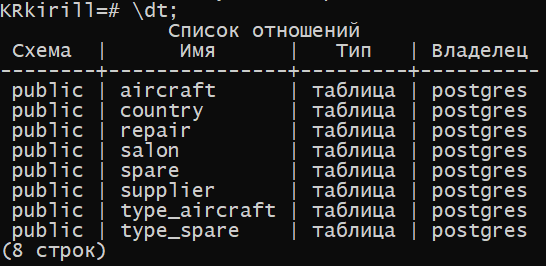
Таблица supplier содержит информацию о поставщиках запчастей:

Таблица 8 – Структура supplier

|  |  |  |  |
| --- | --- | --- | --- |
| Supplier | | | |
| Название поля | Тип данных | Ограничения | Комментарий |
| id | serial | primary key, not null | идентификатор поставщика |
| name | integer | foreign key, not null | наименование поставщика |

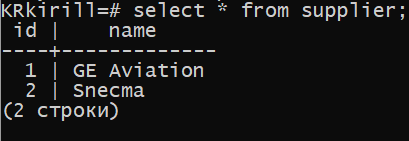
# **СОДЕРЖАНИЕ ТАБЛИЦ БАЗ ДАННЫХ**

Список отношений базы данных:



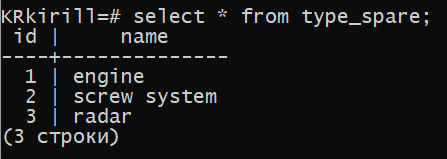
*Рисунок 2 – Таблицы базы данных*

Содержание таблицы-справочника supplier:



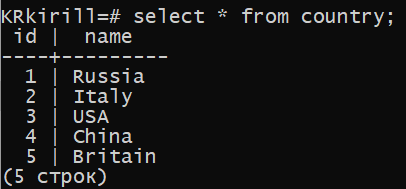
*Рисунок 3 – Содержание таблицы-справочника* supplier

Содержание таблицы-справочника type\_spare:



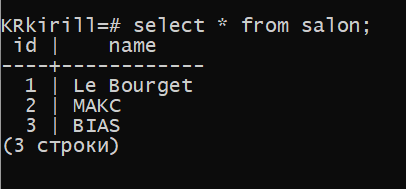
*Рисунок 4 – Содержание таблицы-справочника* type\_spare

Содержание таблицы-справочника country:



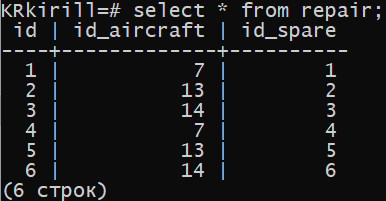
*Рисунок 5 – Содержание таблицы-справочника* country

Содержание таблицы-справочника salon:



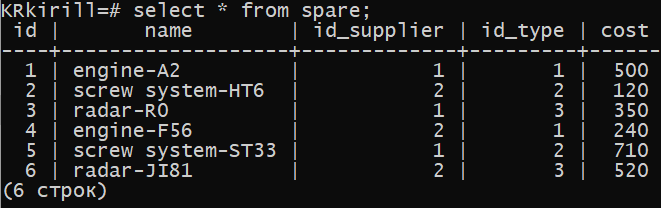
*Рисунок 6 – Содержание таблицы-справочника* salon

Содержание таблицы repair:



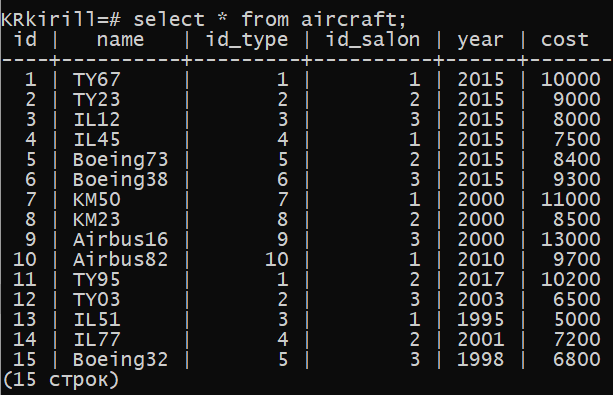
*Рисунок 7 – Содержание таблицы* repair

Содержание таблицы spare:



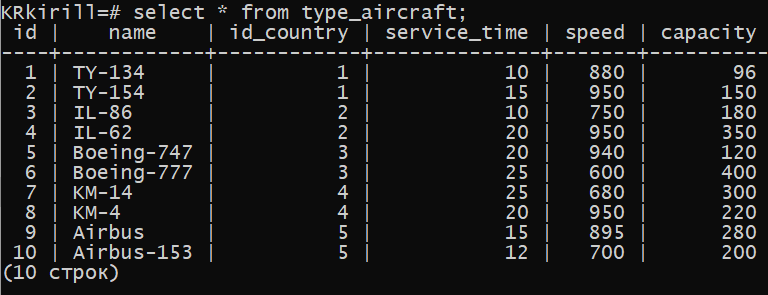
*Рисунок 8 – Содержание таблицы* spare

Содержание таблицы aircraft:



*Рисунок 9 – Содержание таблицы* aircraft

Содержание таблицы type\_aircraft:



*Рисунок 10 – Содержание таблицы* type\_aircraft

# **аыСОДЕРЖАНИЕ ТАБЛИЦ БАЗ ДАННЫХ**

# **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

*Таблица 9 – Описание функций*

|  |  |  |  |
| --- | --- | --- | --- |
| Номер задания, для которого написана функция | Наименование функции | Аргумент функции | Описание функции |
| 1,2 | aircraft\_type\_info() | - | Выводит сведения по видам самолета (сортировка по наименованию вида) |
| 2 | aircraft\_type\_order\_by\_year() | - | Выводит сведения по видам самолета (сортировка по году выпуска) |
| 2 | aircraft\_type\_order\_by\_cost() | - | Выводит сведения по видам самолета (сортировка по стоимости) |
| 2 | aircraft\_type\_order\_by\_service\_time() | - | Выводит сведения по видам самолета (сортировка по сроку эксплуатации) |
| 3 | part\_old\_aircraft(max\_st int) | max\_st – срок эксплуатации | Выводит долю старых самолетов (срок эксплуатации больше заданного) |
| 4 | cost\_type\_aircraft(in\_type varchar(40)) | in\_type – тип самолета | По заданному типу выводит самый дорогой самолет и его стоимость, самый дешевый и его стоимость, среднюю стоимость данного типа самолетов |
| 5 | expensive\_aircraft(in\_cost int) | in\_cost – минимальная стоимость самолета | Выводит самолеты (цена которых больше заданной) и их стоимость |
| 6 | fast\_aircraft\_from\_country(in\_country varchar(40), min\_speed int) | in\_country – наименование страны  min\_speed – минимальная скорость | Выводит все самолеты, поступившие из заданной страны, со скоростью больше заданной |
| 7 | aircraft\_quantity\_ta(int) | срок эксплуатации | Выводит наименование типа самолета, наименование салона, количество самолетов каждого типа с заданным сроком эксплуатации |
| 7 | aircraft\_quantity\_st(int) | срок эксплуатации | Выводит наименование салона и количество самолетов с заданным сроком эксплуатации |
| 8 | part\_aircraft\_from\_country(c varchar(40)) | с – наименование страны | Выводит наименование и количество самолетов, поступивших из заданной страны, и их долю от общего числа самолетов |
| 9 | aircraft\_capacity\_year(c int,y int) | с – наименование страны  у – год выпуска самолета | Выводит наименование самолета, его вместительность и год выпуска для заданных – вместительности и года выпуска |
| 10 | aircraft\_capacity\_year(in\_year int,in\_country varchar(40),min\_speed int, max\_speed int) | in\_year – год выпуска  in\_country – наименование страны  min\_speed – минимальная скорость  max\_speed – максимальная скорость | Выводит самолеты с заданным годом выпуска (наименование самолета, год выпуска, стоимость, средняя стоимость по заданным условиям, страну, скорость, количество самолетов) , стоимость которых больше, чем средняя стоимость самолетов, поступивших из заданной страны и с заданными ограничениями по скорости |
| 11 | repair() | - | Выводит количество: ремонтируемых самолетов, самолетов, подлежащих ремонту (сломанные), самолетов, готовых к эксплуатации(готовые) и долю(в процентах) сломанных к готовым |
| 12 | spare\_by\_name() | - | Выводит вид запасных частей, поступающих для ремонта самолетов отсортированных по наименованиям |
| 12 | spare\_by\_cost() | - | Выводит вид запасных частей, поступающих для ремонта самолетов, отсортированных по стоимости |
| 12 | spare\_by\_supplier() | - | Выводит вид запасных частей, поступающих для ремонта самолетов отсортированных по поставщику |
| 13 | supplier(spare\_t varchar(40), aircraft\_t varchar(40)) | spare\_t – тип запчасти  aircraft\_t – тип самолета | Выводит поставщиков заданного вида запасных частей (выбор) для заданного вида самолетов. |

# **ТЕКСТЫ ЗАПРОСОВ И ПРИМЕРЫ РАБОТЫ**

Создадим базу данных с названием KRkirill;

Create database KRkirill;

Роли

Создадим следующие роли: оператор БД, пользователь БД, администратор БД.

Оператор БД (пополнение справочников).

**Текст запроса:**

CREATE ROLE role\_operator WITH password '0001' LOGIN;

GRANT INSERT, UPDATE, SELECT ON supplier, type\_spare, country, salon TO role\_operator;

Пользователь БД (основная работа с БД, с ограничениями для

некоторого вида операций).

**Текст запроса:**

CREATE ROLE role\_user WITH password '0002' LOGIN;

GRANT SELECT ON ALL TABLES IN SCHEMA public TO role\_user;

GRANT INSERT ON ALL TABLES IN SCHEMA public TO role\_user;

GRANT UPDATE ON ALL TABLES IN SCHEMA public TO role\_user;

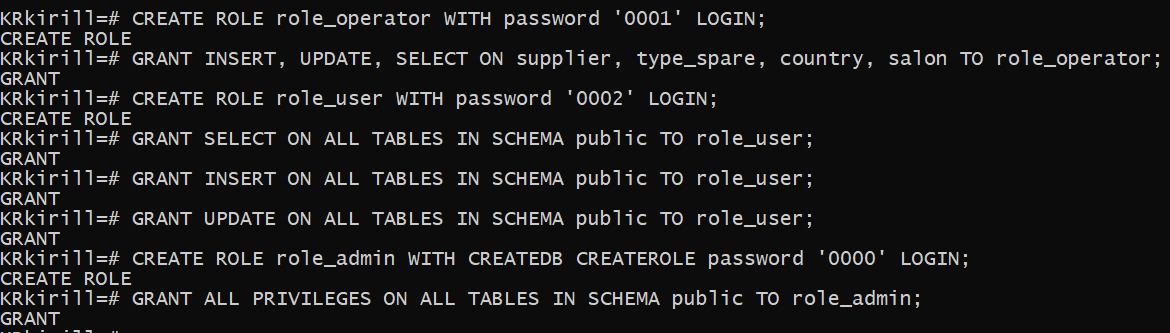
Администратор БД (любые изменения БД).

**Текст запроса:**

CREATE ROLE role\_admin WITH CREATEDB CREATEROLE password '0000' LOGIN;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO role\_admin;

**Исполнение запроса:**



*Рисунок 11 – Создание ролей*

Проверим как работают роли. Например, оператор может пополнять (INSERT, UPDATE, т.к. для UPDATE нужны права на SELECT, добавим права и на эту операцию) только справочники.

**Текст запроса:**

\c KRkirill role\_operator;

INSERT INTO country(id,name)

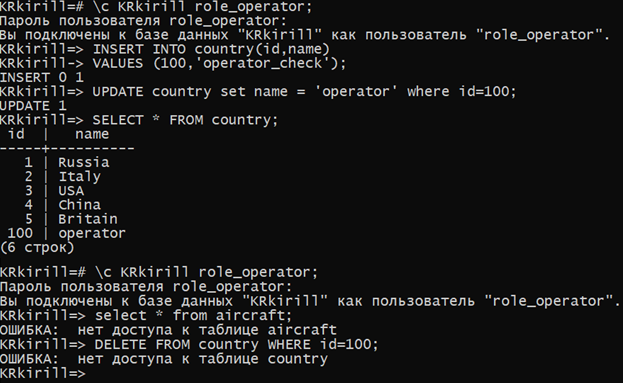
VALUES (100,'operator\_check');

UPDATE country set name = 'operator' where id=100;

SELECT \* FROM country;

\c KRkirill postgres;

DELETE FROM country WHERE id=100;

  
*Рисунок 12 – Работа в БД под ролью оператора и его ограничения*

\c KRkirill role\_user;

INSERT INTO country(id,name)

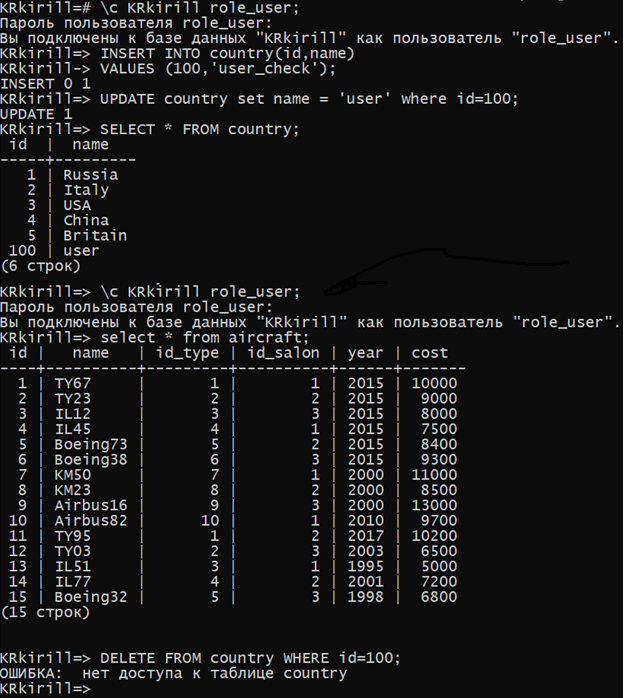
VALUES (100,'user\_check');

UPDATE country set name = 'user' where id=100;

SELECT \* FROM country;

\c KRkirill postgres;

DELETE FROM country WHERE id = 100;

  
*Рисунок 13 – Работа в БД под ролью пользователя и его ограничения*

\c KRkirill role\_admin;

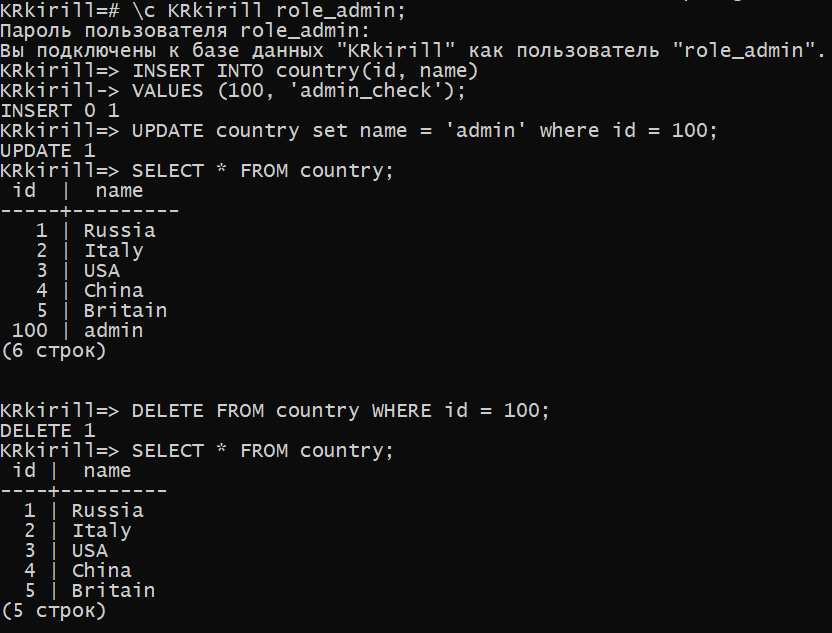
INSERT INTO country(id, name)

VALUES (100, 'admin\_check');

UPDATE country set name = 'admin' where id = 100;

SELECT \* FROM country;

DELETE FROM country WHERE id = 100;



*Рисунок 14 – Работа в БД под ролью администратора*

1. Для каждого вида самолета указать сведения о нем (тип, год выпуска, срок эксплуатации, место изготовление, цена, скорость, вместимость)

**Текст запроса:**

create or replace function aircraft\_type\_info()

returns table (type\_name varchar(40),year numeric,service\_time numeric,country varchar(40),cost numeric,speed numeric,capacity numeric)

as $$ begin

return query

select type\_aircraft.name as type\_name,

aircraft.year,

type\_aircraft.service\_time,

country.name as country,

aircraft.cost,

type\_aircraft.speed,

type\_aircraft.capacity

from type\_aircraft

join aircraft on type\_aircraft.id = aircraft.id\_type

join country on type\_aircraft.id\_country = country.id

order by type\_name;

end $$

language plpgsql;

select \* from aircraft\_type\_info();

**Исполнение запроса:**

  
*Рисунок 15 – Сведения для каждого вида самолета отсортированные по имени самолета*

2. Для каждого вида самолета выдать список, отсортированный: по году выпуска.

**Текст запроса:**

create or replace function aircraft\_type\_order\_by\_year()

returns table (type\_name varchar(40),year numeric,service\_time numeric,country varchar(40),cost numeric,speed numeric,capacity numeric)

as $$ begin

return query

select type\_aircraft.name as type\_name,

aircraft.year,

type\_aircraft.service\_time,

country.name as country,

aircraft.cost,

type\_aircraft.speed,

type\_aircraft.capacity

from type\_aircraft

join aircraft on type\_aircraft.id = aircraft.id\_type

join country on type\_aircraft.id\_country = country.id

order by year;

end $$

language plpgsql;

select \* from aircraft\_type\_order\_by\_year();  
  
**Исполнение запроса:***Рисунок 16 – Сведения для каждого вида самолета., отсортированные по году выпуска*

2. Для каждого вида самолета выдать список, отсортированный: по стоимости.

**Текст запроса:**

create or replace function aircraft\_type\_order\_by\_cost()

returns table (type\_name varchar(40),year numeric,service\_time numeric,country varchar(40),cost numeric,speed numeric,capacity numeric)

as $$ begin

return query

select type\_aircraft.name as type\_name,

aircraft.year,

type\_aircraft.service\_time,

country.name as country,

aircraft.cost,

type\_aircraft.speed,

type\_aircraft.capacity

from type\_aircraft

join aircraft on type\_aircraft.id = aircraft.id\_type

join country on type\_aircraft.id\_country = country.id

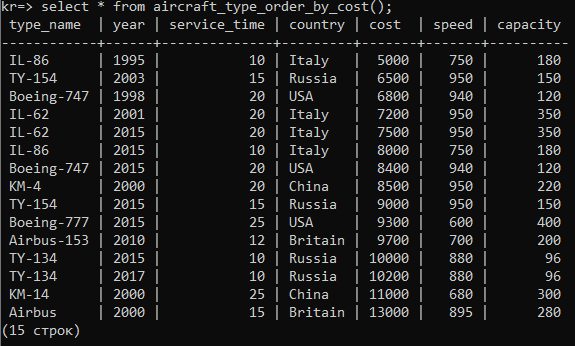
order by cost;

end $$

language plpgsql;

select \* from aircraft\_type\_order\_by\_cost();

**Исполнение запроса:**

  
*Рисунок 17 – Сведения для каждого вида самолета, отсортированные по стоимости*

2. Для каждого вида самолета выдать список, отсортированный: по сроку эксплуатации.

**Текст запроса:**

create or replace function aircraft\_type\_order\_by\_service\_time()

returns table (type\_name varchar(40),year numeric,service\_time numeric,country varchar(40),cost numeric,speed numeric,capacity numeric)

as $$ begin

return query

select type\_aircraft.name as type\_name,

aircraft.year,

type\_aircraft.service\_time,

country.name as country,

aircraft.cost,

type\_aircraft.speed,

type\_aircraft.capacity

from type\_aircraft

join aircraft on type\_aircraft.id = aircraft.id\_type

join country on type\_aircraft.id\_country = country.id

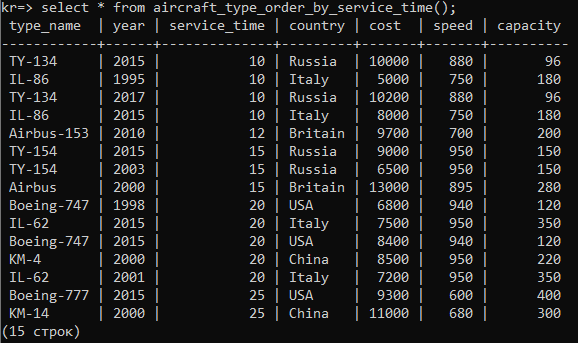
order by service\_time;

end $$

language plpgsql;

select \* from aircraft\_type\_order\_by\_service\_time();

**Исполнение запроса:**

  
*Рисунок 18 – Сведения для каждого вида самолета, отсортированные по времени эксплуатации*

3. Найти долю «старых» самолетов (срок эксплуатации больше заданного, ввод с клавиатуры) от общего числа самолетов.

**Текст запроса:**

create or replace function part\_old\_aircraft(max\_st int)

returns table (aircraft varchar(40) ,count bigint, part float) as $$

declare all\_count int;

declare part\_count int;

BEGIN

all\_count = (select count(id) from aircraft);

part\_count = (select count(aircraft.id)

from aircraft join type\_aircraft on type\_aircraft.id=aircraft.id\_type

and type\_aircraft.service\_time > max\_st);

return query (select aircraft.name, count (aircraft.id) as count\_id,

part\_count/all\_count::bigint::float as part

from type\_aircraft

join aircraft on aircraft.id\_type = type\_aircraft.id

and type\_aircraft.service\_time > max\_st

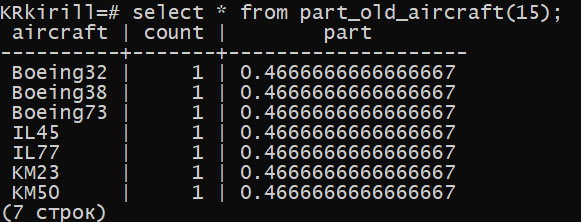
group by aircraft.name

order by aircraft.name);

END; $$ language plpgsql;

select \* from part\_old\_aircraft(15);

**Исполнение запроса:**

  
*Рисунок 19 –* Доля старых самолетов

4. Найти самый дорогой самолет (по каждому типу), самый дешевый, среднюю стоимость.

**Текст запроса:**

create or replace function cost\_type\_aircraft(in\_type varchar(40))

returns table (most\_expensive\_aircraft varchar(40), max\_cost bigint,cheapest\_aircraft varchar(40), min\_cost bigint, avg\_cost bigint) as $$

declare max\_cost bigint;

declare min\_cost bigint;

declare avg\_cost bigint;

declare max\_air varchar(40);

declare min\_air varchar(40);

BEGIN

max\_cost = (select max(aircraft.cost) from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

and type\_aircraft.name = in\_type

);

min\_cost = (select min(aircraft.cost) from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

and type\_aircraft.name = in\_type

);

avg\_cost = (select avg(aircraft.cost) from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

and type\_aircraft.name = in\_type

);

max\_air = (select aircraft.name from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

and type\_aircraft.name = in\_type

where aircraft.cost = max\_cost

);

min\_air = (select aircraft.name from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

and type\_aircraft.name = in\_type

where aircraft.cost = min\_cost

);

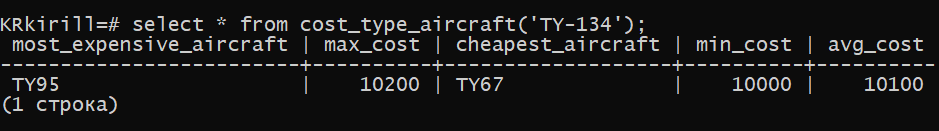
return query (select max\_air, max\_cost, min\_air, min\_cost, avg\_cost

);

END; $$ language plpgsql;

select \* from cost\_type\_aircraft('TY-134');

**Исполнение запроса:**

  
*Рисунок 20 – По заданному типу выводит самый дорогой самолет, минимальную стоимость самолета этого тип и среднюю стоимость*

5. Найти самолеты с ценой свыше 5000 т.р. (и любая другая сумма, т.е. предусмотреть ввод цены с клавиатуры).

**Текст запроса:**

create or replace function expensive\_aircraft(in\_cost int)

returns table (aircraft varchar (40), cost numeric) as $$

BEGIN

return query (select aircraft.name, aircraft.cost

from aircraft

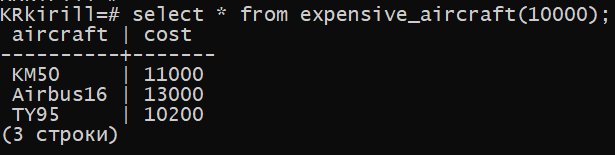
where aircraft.cost > in\_cost

);

END; $$ language plpgsql;

select \* from expensive\_aircraft(10000);

**Исполнение запроса:**

  
*Рисунок 21 – Поиск самолета (цена которого больше заданной) и его стоимость*

6. Найти все самолеты, поступившие из заданной страны (ввод страны), чья скорость больше заданной (ввод скорости).

**Текст запроса:**

create or replace function fast\_aircraft\_from\_country(in\_country varchar(40), min\_speed int)

returns table (aircraft varchar(40), country varchar(40), speed numeric) as $$

BEGIN

return query (select aircraft.name, country.name, type\_aircraft.speed

from type\_aircraft

join aircraft on aircraft.id\_type = type\_aircraft.id

join country on country.id = type\_aircraft.id\_country

and country.name = in\_country

where type\_aircraft.speed > min\_speed

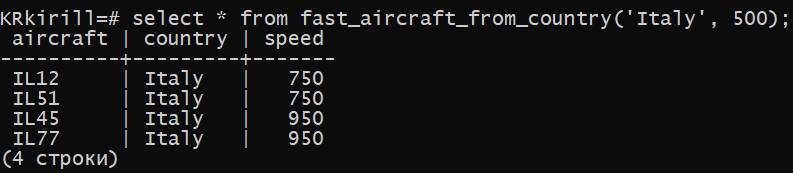
order by type\_aircraft.speed

);

END; $$ language plpgsql;

select \* from fast\_aircraft\_from\_country('Italy', 500);

**Исполнение запроса:**

  
*Рисунок 22 – Поиск самолета, поступивших из заданной страны, со скоростью больше заданной*

7. Найти количество самолетов на авиасалоне (по каждому типу и в целом) с заданным сроком эксплуатации (ввод срока).

**Текст запроса:**

**ПО ТИПУ**

create or replace function aircraft\_quantity\_ta(int)

returns table (aircraftType varchar(40), salonName varchar(40), count bigint)

as $$

BEGIN

return query (

select type\_aircraft.name, salon.name, count(aircraft.id)

from aircraft, type\_aircraft, salon

where aircraft.id\_type = type\_aircraft.id

and aircraft.id\_salon = salon.id

and type\_aircraft.service\_time = 10

group by (type\_aircraft.name, salon.name)

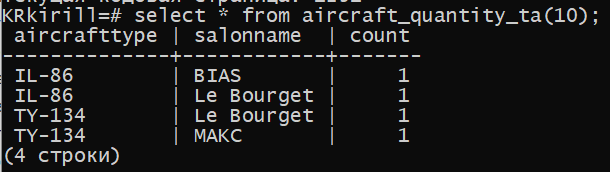
);

END;

$$ language plpgsql;

select \* from aircraft\_quantity\_ta(10);

**Исполнение запроса:**

  
*Рисунок 23 – Поиск типа самолета, с выводом наименование салона и количеством самолетов каждого типа с заданным сроком эксплуатации*

7. Найти количество самолетов на авиасалоне (по каждому типу и в целом) с заданным сроком эксплуатации (ввод срока).

**Текст запроса:**

общий вывод

create or replace function aircraft\_quantity\_st(int)

returns table (salonName varchar(40), count bigint)

as $$

BEGIN

return query (

select salon.name, count(aircraft.id)

from aircraft, type\_aircraft, salon

where aircraft.id\_type = type\_aircraft.id

and aircraft.id\_salon = salon.id

and type\_aircraft.service\_time = $1

group by (salon.name)

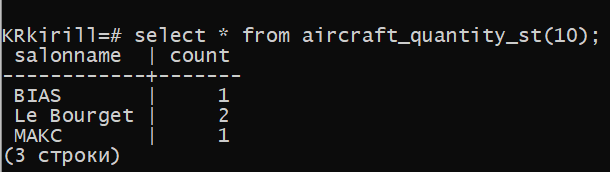
);

END;

$$ language plpgsql;

select \* from aircraft\_quantity\_st(10);

**Исполнение запроса:**

  
*Рисунок 24 – Поиск* наименование салона и количество самолетов с заданным сроком эксплуатации

8. Найти долю самолетов, поступивших из заданной страны (выбор страны) от общего числа самолетов.

**Текст запроса:**

create or replace function part\_aircraft\_from\_country(c varchar(40))

returns table (aircraft varchar(40) ,count bigint, part float) as $$

declare all\_count int;

BEGIN

all\_count = (select count(id) from aircraft);

return query (select aircraft.name, count (aircraft.id) as count\_id,

count(aircraft.id)/all\_count::bigint::float as part

from type\_aircraft

left join aircraft on aircraft.id\_type = type\_aircraft.id

left join country on country.id = type\_aircraft.id\_country

and country.name = c

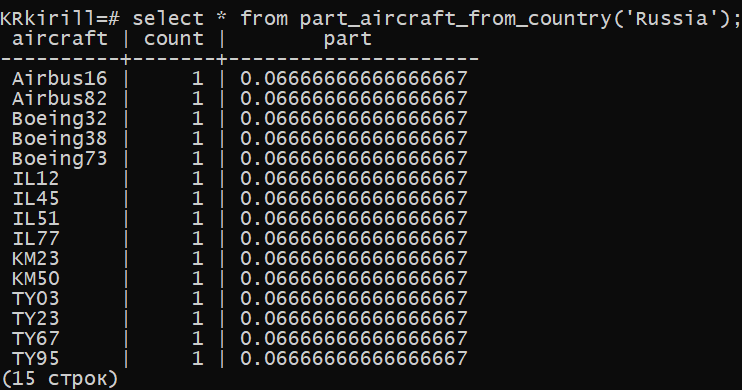
group by aircraft.name

order by aircraft.name);

END; $$ language plpgsql;

select \* from part\_aircraft\_from\_country('Russia');

**Исполнение запроса:**

  
*Рисунок 25 – Поиск по заданной страны, наименование и количество самолетов*

9. Найти все самолеты заданной вместимости (ввод) для заданного года выпуска (ввод).

**Текст запроса:**

create or replace function aircraft\_capacity\_year(c int,y int)

returns table (aircraft varchar(40) ,capacity numeric, year numeric) as $$

BEGIN

return query (select aircraft.name, type\_aircraft.capacity, aircraft.year

from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

and type\_aircraft.capacity = c

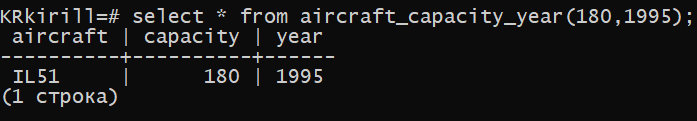
where aircraft.year = y

group by aircraft.name,type\_aircraft.capacity,aircraft.year

order by aircraft.name);

END; $$ language plpgsql;

select \* from aircraft\_capacity\_year(180,1995);

**Исполнение запроса:** *Рисунок 26 – Поиск по вместительности и году выпуска*

10. Найти количество самолетов заданного года выпуска (ввод), чья стоимость больше, чем средняя стоимость самолетов, поступивших из заданной страны (выбор страны) и заданными ограничениями по скорости (ввод наибольшего и наименьшего значения).

**Текст запроса:**

create or replace function aircraft\_capacity\_year(in\_year int,in\_country varchar(40),min\_speed int, max\_speed int)

returns table (aircraft varchar(40), year numeric, cost numeric,avg\_cost bigint,country varchar(40), speed numeric,count bigint) as $$

declare avg\_cost bigint;

BEGIN

avg\_cost = (select avg(aircraft.cost)

from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type and type\_aircraft.speed < max\_speed and type\_aircraft.speed > min\_speed

join country on country.id = type\_aircraft.id\_country and country.name = in\_country

);

return query (select aircraft.name, aircraft.year, aircraft.cost,avg\_cost, country.name, type\_aircraft.speed, count(aircraft.id)

from type\_aircraft

join aircraft on type\_aircraft.id = aircraft.id\_type

and aircraft.cost > avg\_cost

and aircraft.year = in\_year

join country on country.id = type\_aircraft.id\_country and country.name = in\_country

where type\_aircraft.speed <= max\_speed and type\_aircraft.speed >= min\_speed

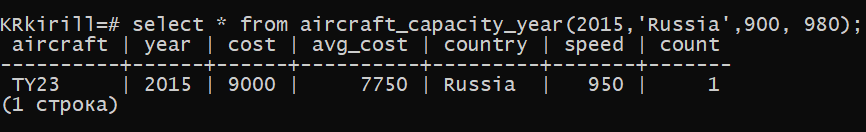
group by aircraft.name, aircraft.year, aircraft.cost,avg\_cost, country.name, type\_aircraft.speed

order by aircraft.name);

END; $$ language plpgsql;

select \* from aircraft\_capacity\_year(2015,'Russia',900, 980);

**Исполнение запроса:**

  
*Рисунок 27 – Поиск самолета с заданным годом выпуска*

11. Определить количество ремонтируемых самолетов, соотношение подлежащих ремонту самолетов и годных к эксплуатации самолетов (в процентах).

**Текст запроса:**

create or replace function repair()

returns table (repair\_aircraft bigint, broken\_aircraft bigint, ready\_aircraft bigint, perc\_broken\_to\_ready float) as $$

declare broken\_count bigint;

declare ready\_count bigint;

BEGIN

broken\_count = (select count(aircraft.id) from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

where (aircraft.year + type\_aircraft.service\_time) > 2022);

ready\_count = (select count(aircraft.id) from aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

where (aircraft.year + type\_aircraft.service\_time) <= 2022);

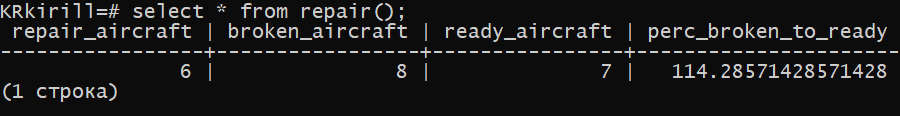
return query (select count(repair.id),broken\_count,ready\_count, broken\_count::bigint::float /ready\_count::bigint::float\*100

from repair);

END; $$ language plpgsql;

select \* from repair();

**Исполнение запроса:**

  
Рисунок 28 – Поиск ремонтируемых самолетов

12. Найти вид запасных частей, поступающих для ремонта самолетов отсортировать по наименованиям, по стоимости, по поставщику

**Текст запроса:**

**Сортировка по наименованию**

create or replace function spare\_by\_name()

returns table (spare varchar(40), type varchar(40), cost numeric, supplier varchar(40)) as $$

BEGIN

return query (select spare.name, type\_spare.name, spare.cost, supplier.name

from spare

join type\_spare on type\_spare.id = spare.id\_type

join supplier on supplier.id = spare.id\_supplier

group by spare.name, type\_spare.name, spare.cost, supplier.name

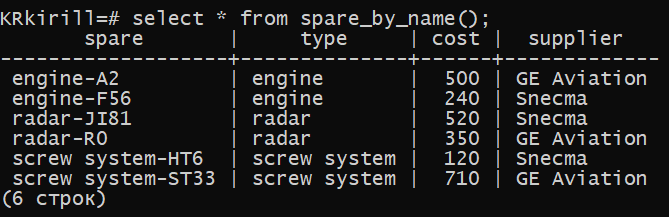
order by spare.name

);

END; $$ language plpgsql;

select \* from spare\_by\_name();

**Исполнение запроса:**

  
*Рисунок 29 –Поиск по производителям запасных частей*

**Текст запроса:**

**Сортировка по стоимости**

create or replace function spare\_by\_cost()

returns table (spare varchar(40), type varchar(40), cost numeric, supplier varchar(40)) as $$

BEGIN

return query (select spare.name, type\_spare.name, spare.cost, supplier.name

from spare

join type\_spare on type\_spare.id = spare.id\_type

join supplier on supplier.id = spare.id\_supplier

group by spare.name, type\_spare.name, spare.cost, supplier.name

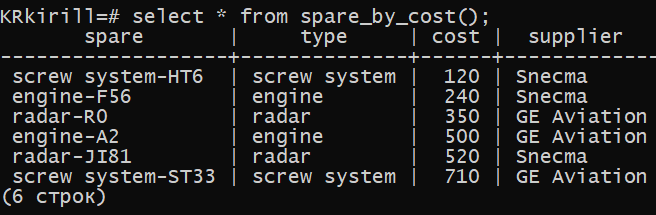
order by spare.cost

);

END; $$ language plpgsql;

select \* from spare\_by\_cost();

**Исполнение запроса:**

  
*Рисунок 30 – Поиск запчастей по стоимости*

**Текст запроса:**

**Сортировка по поставщику**

create or replace function spare\_by\_supplier()

returns table (spare varchar(40), type varchar(40), cost numeric, supplier varchar(40)) as $$

BEGIN

return query (select spare.name, type\_spare.name, spare.cost, supplier.name

from spare

join type\_spare on type\_spare.id = spare.id\_type

join supplier on supplier.id = spare.id\_supplier

group by spare.name, type\_spare.name, spare.cost, supplier.name

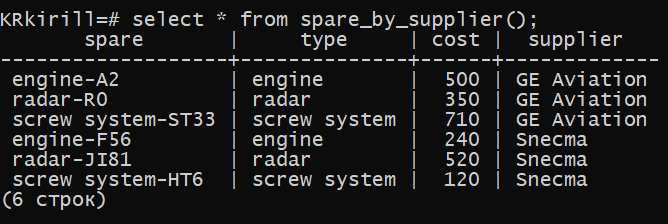
order by supplier.name

);

END; $$ language plpgsql;

select \* from spare\_by\_supplier();

**Исполнение запроса:**

  
*Рисунок 31 – Поиск запчастей по производителю*

13. Найти поставщиков заданного вида запасных частей (выбор) для заданного вида самолетов.

**Текст запроса:**

create or replace function supplier(spare\_t varchar(40), aircraft\_t varchar(40))

returns table (supplier varchar(40), spare\_type varchar(40), aircraft\_type varchar(40)) as $$

BEGIN

return query (select supplier.name, type\_spare.name,type\_aircraft.name

from spare

join type\_spare on type\_spare.id = spare.id\_type

and type\_spare.name = spare\_t

join supplier on supplier.id = spare.id\_supplier

join repair on repair.id\_spare = spare.id

join aircraft on aircraft.id = repair.id\_aircraft

join type\_aircraft on type\_aircraft.id = aircraft.id\_type

and type\_aircraft.name = aircraft\_t

group by supplier.name, type\_spare.name,type\_aircraft.name

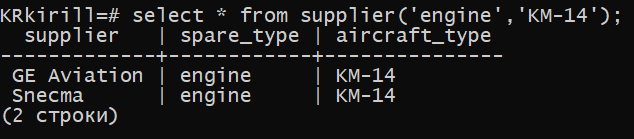
order by supplier.name

);

END; $$ language plpgsql;

select \* from supplier('engine','KM-14');

**Исполнение запроса:**

  
*Рисунок 32 – Поиск типа самолетов с заданным производителем запчасти*

# **АНАЛИЗ РЕЗУЛЬТАТОВ И ВЫВОДЫ**

В ходе выполнения курсовой работы была создана и заполнена база данных KRkirill. В базе данных были созданы таблицы, связанных между собой. Созданные таблицы соответствуют требованиям выданного задания.

Были созданы функции для выполнения различных запросов, соответствующих заданию. Эти функции были протестированы на работоспособность и корректность выдаваемого результата.

Также для базы данных были созданы роли – роль оператора, пользователя и администратора. Они различаются правами доступа к базе данных и функциям.

Для выполнения работы применялись навыки построения баз данных, написания запросов и функций, полученных в ходе выполнения лабораторных работ по дисциплине «Базы данных».

# **СПИСОК ЛИТЕРАТУРЫ**

1. Трошина Г. В. Базы данных: Электронное учебное пособие / Г. В. Трошина. –: ИДО НГТУ, 2006.