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***»**

**Вариант № 89**

**на тему:** «проектирование базы данных “Детские издания ”»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| Студент гр. АВТ-819, АВТФ |  | *Доцент* |
| *Ванин Константин* |  | *Трошина Г. В.* |
| « » 2021 г. |  | « » 2021 г. |
| (подпись) |  | (подпись) |

Новосибирск 2021

Оглавление

[*Задание* 3](#_Toc82113294)

[Ход работы 5](#_Toc82113295)

[*Структурное представление связей между таблицами* 5](#_Toc82113296)

[*Содержание таблиц базы данных* 5](#_Toc82113297)

[Руководство пользователя 10](#_Toc82113298)

[*Список, отсортированный по тиражу* 13](#_Toc82113299)

[*Список, отсортированный в алфавитном порядке по стоимости* 14](#_Toc82113300)

[*Самое дорогое, самое дешевое, средняя стоимость издания.* 16](#_Toc82113301)

[*Детские печатные издания с ценой свыше 50 р. (и любая другая сумма, т.е. предусмотреть ввод цены с клавиатуры).* 18](#_Toc82113302)

[*Все детские печатные издания, чей тираж попадает в заданные пределы (ввод интервала).* 19](#_Toc82113303)

[*Для заданного издательства (выбор) все) детские печатные издания.* 20](#_Toc82113304)

[*Все детски печатные издания, чья стоимость находится в заданных пределах (ввод интервала) для заданного издательства.* 20](#_Toc82113305)

[*Доля дешевых детских печатных изданий (меньше заданного, ввод ограничения) от общего числа детских печатных изданий.* 21](#_Toc82113306)

[Доля детских печатных изданий, проданной за определенной период (ввод). 22](#_Toc82113307)

[*Все виды детских печатных изданий, поступивших из заданного издательства (ввод издательства), чья стоимость больше заданной (ввод стоимости.)* 23](#_Toc82113308)

[Все детские печатные издания, поступившие от заданного издательства, чья стоимость больше, чем средняя стоимость заданного детского печатного издания (выбор наименования), поступившего из заданной страны (выбор страны). 23](#_Toc82113309)

[*Доля дешевых детских печатных изданий (чья стоимость меньше заданной, ввод стоимости), поступивших от заданного издательства и в целом.* 25](#_Toc82113310)

[*Средняя стоимость детских печатных изданий, проданных за определённый промежуток времени (ввод интервала).* 26](#_Toc82113311)

[*Все детские печатные издания, чья стоимость выше, чем средняя стоимость детских печатных изданий заданного издательства.* 27](#_Toc82113312)

[*Детские печатные издания, которые лучше всего раскупают, указать параметры (стоимость, издательство и т.д.).* 29](#_Toc82113313)

[**Создание таблицы логов:** 29](#_Toc82113314)

[Создание роли оператора, пользователя и администратора 30](#_Toc82113315)

[Выводы: 33](#_Toc82113316)

[Список использованной литературы 34](#_Toc82113317)

Введение

## *Задание*

Основная база данных «Издания». 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 базы данных. Реализовать следующие типы запросов с использованием нескольких таблиц:

* Для каждого детского печатного издания указать сведения о нём (наименование, издательство, цена, тираж и т.п.).
* Получить список, отсортированный: в алфавитном порядке по наименованию, по тиражу, по стоимости.
* Найти самое дорогое издание, самое дешевое, среднюю стоимость.
* Найти детские печатные издания с ценой свыше 50 р. (и любая другая сумма, т.е. предусмотреть ввод цены с клавиатуры).
* Найти все детские печатные издания, чей тираж попадает в заданные пределы (ввод интервала).
* Для заданного издательства (выбор) найти все детские печатные издания.
* Найти все детски печатные издания, чья стоимость находится в заданных пределах (ввод интервала) для заданного издательства.
* Найти долю дешевых детских печатных изданий (меньше заданного, ввод ограничения) от общего числа детских печатных изданий.
* Найти долю детских печатных изданий, проданной за определенной период (ввод).
* Найти все виды детских печатных изданий, поступивших из заданного издательства (ввод издательства), чья стоимость больше заданной (ввод стоимости.)
* Найти все детские печатные издания, поступившие от заданного издательства, чья стоимость больше, чем средняя стоимость заданного детского печатного издания (выбор наименования), поступившего из заданной страны (выбор страны).
* Найти долю дешевых детских печатных изданий (чья стоимость меньше заданной, ввод стоимости), поступивших от заданного издательства и в целом.
* Найти среднюю стоимость детских печатных изданий, проданных за определённый промежуток времени (ввод интервала).
* Найти все детские печатные издания, чья стоимость выше, чем средняя стоимость детских печатных изданий заданного издательства.
* Найти детские печатные издания, которые лучше всего раскупают, указать параметры (стоимость, издательство и т.д.).

# Ход работы

## *Структурное представление связей между таблицами*

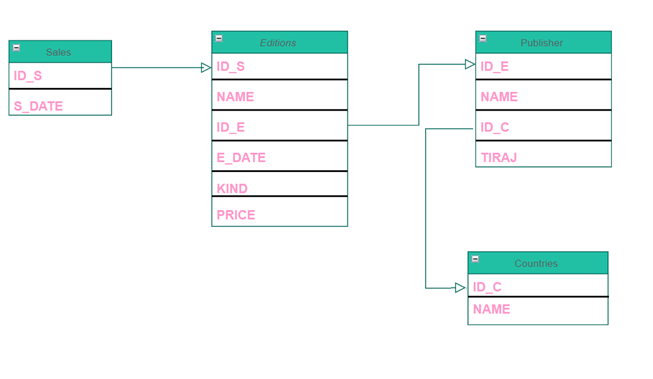


Рисунок 1. Структура базы данных, связи между таблицами

## *Содержание таблиц базы данных*

* Таблица Editions (Издания):

|  |  |  |  |
| --- | --- | --- | --- |
|  | *ID\_S* | − | Идентификационный номер издания, |
|  | *NAME* | − | Наименование, |
|  | *ID\_E* | − | Идентификатор издательство, |
|  | *E\_DATE* | − | Дата печати, |
|  | *KIND* | − | Возрастная категория (6+, 12+, 16+), |
|  | *PRICE* | − | Стоимость |

* Таблица Publisher (Издательство):

|  |  |  |  |
| --- | --- | --- | --- |
|  | *ID\_E* | − | Идентификационный номер издательства, |
|  | *NAME* | − | Название издательства, |
|  | *ID\_C* | − | Идентификатор страны. |
|  | *TIRAJ* |  | Тираж |

* Таблица Countries (Страны):

|  |  |  |  |
| --- | --- | --- | --- |
|  | *ID\_С* | − | Идентификационный номер страны, |
|  | *NAME* | − | Название страны. |

* Таблица Sales (Продажи):

|  |  |  |  |
| --- | --- | --- | --- |
|  | *ID\_S* | − | Идентификационный номер издания, |
|  | *S\_DATE* | − | Дата продажи. |
|  |  |  |  |

Таблицы – справочники:

* Countries
* Sales

Таблицы, использующие справочники:

* Editions
* Publisher

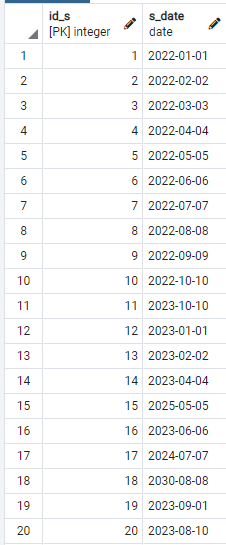
**Содержание таблиц баз данных**

* Таблица-справочник Countries:



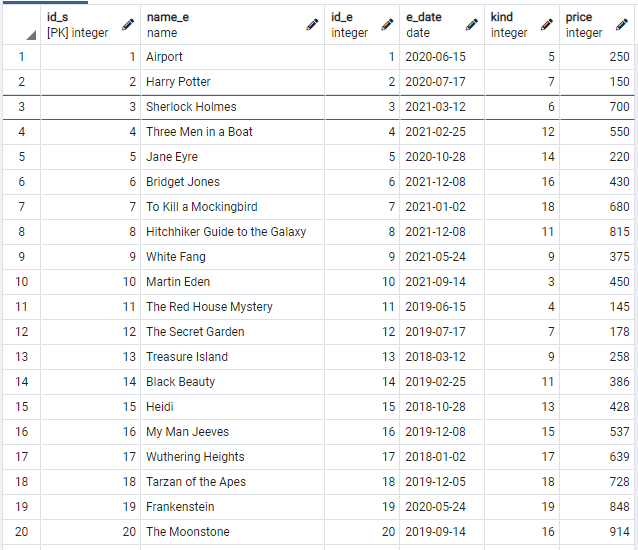
*Рисунок 2 – содержание таблицы-справочника* Countries*.*

* Таблица-справочник Sales (Продажи):



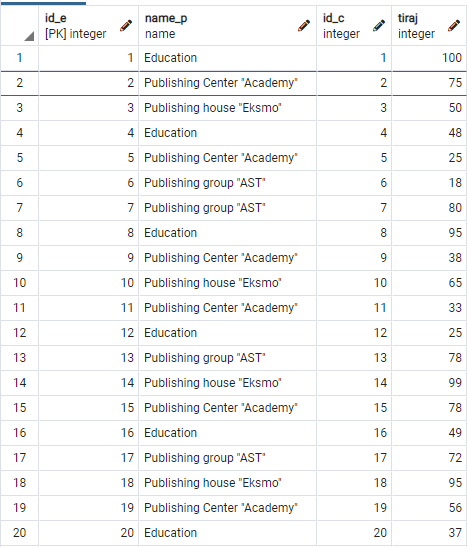
*Рисунок 3 – содержание таблицы-справочника* Sales*.*

Таблица Editions (Издания):



*Рисунок 4 – содержание таблицы-справочника* Editions*.*

* Таблица Publisher:



*Рисунок 5 – содержание таблицы-справочника* Publisher*.*

# Руководство пользователя

all\_info() - Для каждого детского печатного издания указать сведения о нём. Входные параметры: нет.

all\_info\_p() - Получить список, отсортированный: в алфавитном порядке по наименованию. Входные параметры: нет.

all\_info\_p\_t() - Получить список, отсортированный: *по тиражу.* Входные параметры: нет.

avg\_b\_price(date,date) - Найти среднюю стоимость детских печатных изданий, проданных за определённый промежуток времени. Входные параметры: промежуток дат.

avg\_c\_price(name) - Найти все детские печатные издания, чья стоимость выше, чем средняя стоимость детских печатных изданий заданного издательства. Входные параметры: Имя.

avg\_price() -  *Средняя стоимость.* Входные параметры: Нет.

best\_sell() - Детские печатные издания, которые лучше всего раскупают. Входные параметры: Нет.

ed\_country\_all(name,name) - все детские печатные издания, поступившие от заданного издательства, чья стоимость больше, чем средняя стоимость заданного детского печатного издания , поступившего из заданной страны. Входные параметры: выбор наименования и выбор страны.

find\_b\_p(name,integer) - Все виды детских печатных изданий, поступивших из заданного издательства , чья стоимость больше заданной. Входные параметры: ввод издательства и ввод стоимости.

find\_p\_pub(integer,integer,name) - Все детские печатные издания, чья стоимость находится в заданных пределах для заданного издательства. Входные параметры: ввод интервала и издательства.

find\_price(integer,integer) - Детские печатные издания с любой ценой. Входные параметры: ввод интервала цены.

find\_publisher(name) - Для заданного издательства найти все детские печатные издания.

Входные параметры: выбор издательства .

find\_tiraj(integer,integer) - Все детские печатные издания, чей тираж попадает в заданные пределы. Входные параметры: ввод интервала.

max\_price() – Максимальная стоимость. Входные параметры: Нет.

min\_price() – Минимальная стоимость. Входные параметры: Нет.

part\_cheap(integer) - Доля дешевых детских печатных изданий (меньше заданного), от общего числа детских печатных изданий. Входные параметры: ввод ограничения.

part\_cheap\_p(integer,name) – Доля дешевых детских печатных изданий (чья стоимость меньше заданной), поступивших от заданного издательства. . Входные параметры: Ввод стоимости.

part\_sale\_date(date,date) - Доля детских печатных изданий, проданной за определенной период . Входные параметры: промежуток дат.

pop\_all() - Самые популярные детские печатные издания. Входные параметры: Нет.

Для каждого детского печатного издания указать сведения о нём (наименование, издательство, цена, тираж, дата печати, дата продажи, страна.).

CREATE or replace FUNCTION all\_info() RETURNS table (name\_e name, publisher name, price integer, tiraj integer, date\_print date, date\_sell date, countries name)

AS

$$

begin

RETURN QUERY

SELECT

editions.name\_e, publisher.name\_p, editions.price, publisher.tiraj, editions.e\_date, sales.s\_date, countries.name

FROM editions

JOIN sales ON editions.id\_s = sales.id\_s

JOIN publisher ON editions.id\_e = publisher.id\_e

JOIN countries ON publisher.id\_c = countries.id\_c

ORDER BY editions.name\_e DESC;

end;

$$

language 'plpgsql';

select \*from all\_info();

**Тексты запросов и примеры работы**

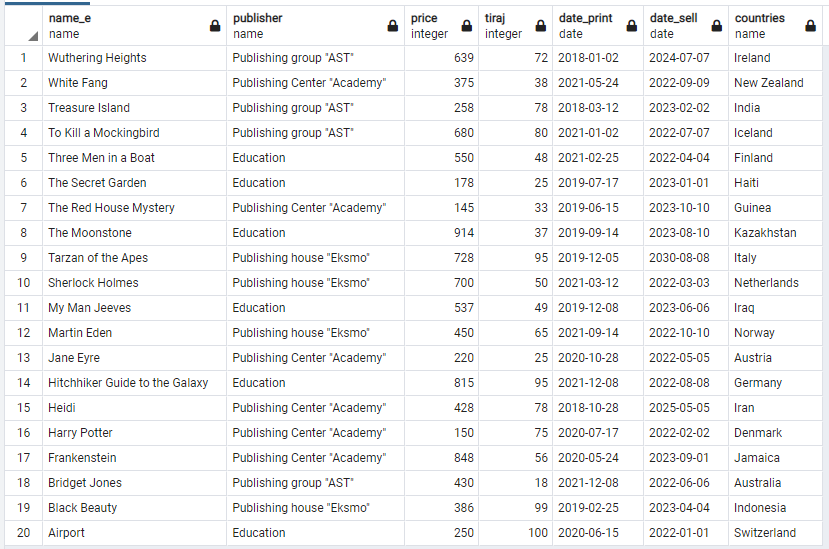


Рисунок 7. Функция all\_info()

Получить список, отсортированный: в алфавитном порядке по наименованию, по тиражу, по стоимости.

## *Список, отсортированный по тиражу*

CREATE or replace FUNCTION all\_info\_t() RETURNS table (name\_e name, publisher name, price integer, tiraj integer, date\_print date, date\_sell date, countries name)

AS

$$

begin

RETURN QUERY

SELECT

editions.name\_e, publisher.name\_p, editions.price, publisher.tiraj, editions.e\_date, sales.s\_date, countries.name

FROM editions

JOIN sales ON editions.id\_s = sales.id\_s

JOIN publisher ON editions.id\_e = publisher.id\_e

JOIN countries ON publisher.id\_c = countries.id\_c

ORDER BY publisher.tiraj DESC;

end;

$$

language 'plpgsql';

select \*from all\_info\_t();

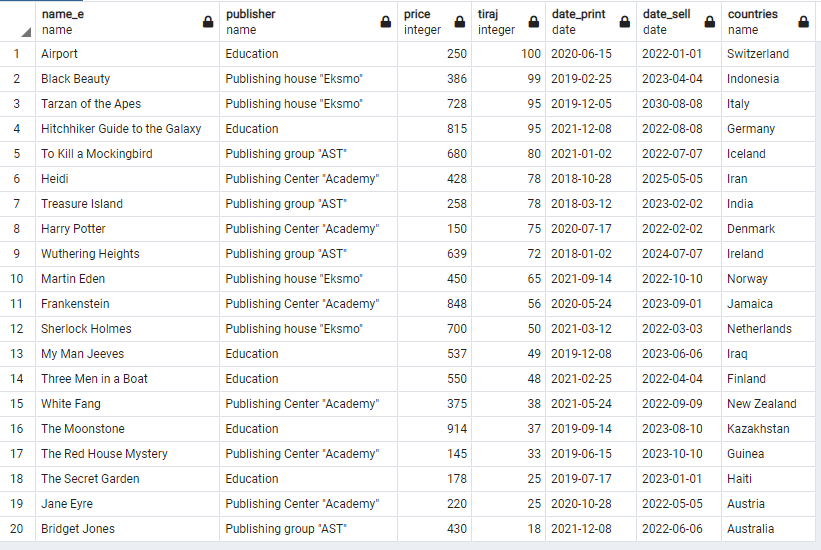


Рисунок 8. Функция all\_info\_t()

## *Список, отсортированный в алфавитном порядке по стоимости*

CREATE or replace FUNCTION all\_info\_p() RETURNS table (name\_e name, publisher name, price integer, tiraj integer, date\_print date, date\_sell date, countries name)

AS

$$

begin

RETURN QUERY

SELECT

editions.name\_e, publisher.name\_p, editions.price, publisher.tiraj, editions.e\_date, sales.s\_date, countries.name

FROM editions

JOIN sales ON editions.id\_s = sales.id\_s

JOIN publisher ON editions.id\_e = publisher.id\_e

JOIN countries ON publisher.id\_c = countries.id\_c

ORDER BY editions.price DESC;

end;

$$

language 'plpgsql';

select \*from all\_info\_p();

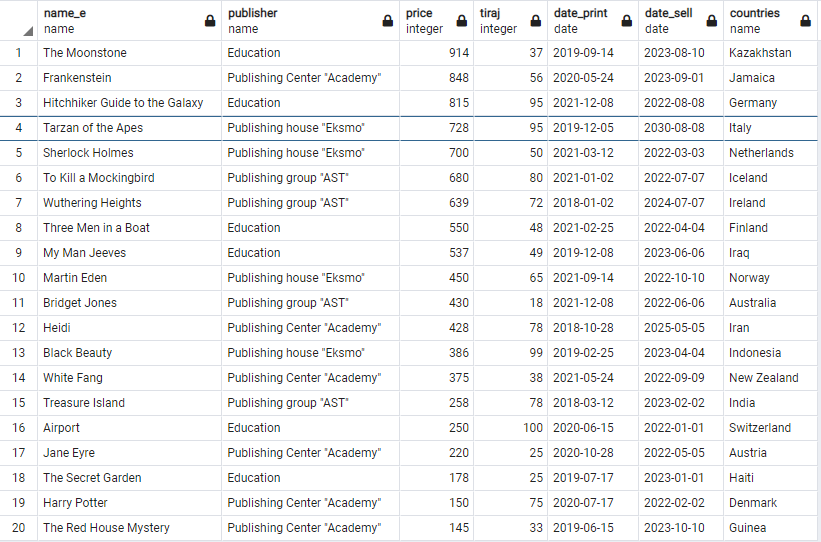


Рисунок 9. Функция all\_info\_p()

## *Самое дорогое, самое дешевое, средняя стоимость издания.*

*CREATE FUNCTION max\_price() RETURNS table (name\_e name, price integer, publisher name) AS*

*$$*

*begin*

*RETURN QUERY*

*SELECT editions.name\_e, editions.price, publisher.name\_p*

*FROM editions, publisher*

*WHERE editions.id\_s = publisher.id\_e*

*AND editions.price = (SELECT MAX(editions.price) from editions);*

*end;*

*$$*

*language 'plpgsql';*

select \*from *max\_price()*;

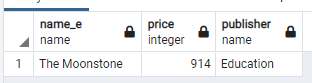


Рисунок 10. Функция *max\_price()*

*CREATE FUNCTION min\_price() RETURNS table (name\_e name, price integer, publisher name) AS*

*$$*

*begin*

*RETURN QUERY*

*SELECT editions.name\_e, editions.price, publisher.name\_p*

*FROM editions, publisher*

*WHERE editions.id\_s = publisher.id\_e*

*AND editions.price = (SELECT MIN(editions.price) from editions);*

*end;*

*$$*

*language 'plpgsql';*

select \*from min\_price();

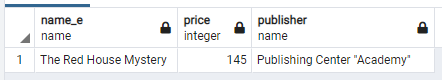


Рисунок 11. Функция min\_price()

*CREATE FUNCTION avg\_price() RETURNS table (price numeric) AS*

*$$*

*begin*

*RETURN QUERY*

*SELECT AVG(editions.price) from editions;*

*end;*

*$$*

*language 'plpgsql';*

select \*from avg\_price()

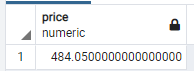


Рисунок 12. Функция avg\_price()

## *Детские печатные издания с ценой свыше 50 р. (и любая другая сумма, т.е. предусмотреть ввод цены с клавиатуры).*

*CREATE FUNCTION find\_price(int, int) RETURNS table (name\_e name, price integer, publisher name) AS*

*$$*

*begin*

*RETURN QUERY*

*SELECT editions.name\_e, editions.price, publisher.name\_p*

*FROM editions*

*JOIN publisher ON editions.id\_s = publisher.id\_e*

*WHERE editions.price BETWEEN $1 AND $2;*

*end;*

*$$*

*language 'plpgsql';*

select \*from find\_price(50,500)



Рисунок 13. Функция find\_price ()

## *Все детские печатные издания, чей тираж попадает в заданные пределы (ввод интервала).*

*CREATE FUNCTION find\_tiraj (int, int) RETURNS table (name\_e name, tiraj integer, publisher name) AS*

*$$*

*begin*

*RETURN QUERY*

*SELECT editions.name\_e, publisher.tiraj, publisher.name\_p*

*FROM editions*

*JOIN publisher ON editions.id\_s = publisher.id\_e*

*WHERE publisher.tiraj BETWEEN $1 AND $2;*

*end;*

*$$*

*language 'plpgsql';*

select \*from find\_tiraj (50,70)

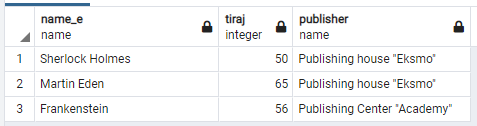


Рисунок 14. Функция find\_price ()

## *Для заданного издательства (выбор) все) детские печатные издания.*

*CREATE FUNCTION find\_publisher(name) RETURNS table (name\_e name, publisher name, price integer) AS*

*$$*

*begin*

*RETURN QUERY*

*SELECT editions.name\_e, publisher.name\_p, editions.price*

*FROM editions*

*JOIN publisher ON editions.id\_s = publisher.id\_e*

*WHERE publisher.name\_p = $1;*

*end;*

*$$*

*language 'plpgsql';*

select \*from find\_publisher('Education')

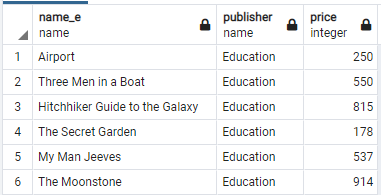


Рисунок 15. Функция find\_ *publisher* ()

## *Все детски печатные издания, чья стоимость находится в заданных пределах (ввод интервала) для заданного издательства.*

*CREATE FUNCTION find\_p\_pub(int, int, name) RETURNS table (name\_e name, price integer, publisher name) AS*

*$$*

*begin*

*RETURN QUERY*

*SELECT editions.name\_e, editions.price, publisher.name\_p*

*FROM editions*

*JOIN publisher ON editions.id\_s = publisher.id\_e*

*WHERE ((editions.price BETWEEN $1 AND $2) AND (publisher.name\_p = $3));*

*end;*

*$$*

*language 'plpgsql';*

select \*from find\_p\_pub(50,1000,'Education');

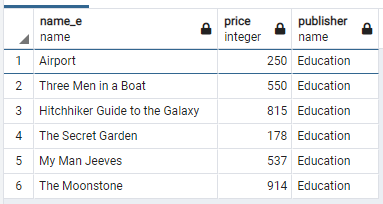


Рисунок 16. Функция find\_p\_pub()

## *Доля дешевых детских печатных изданий (меньше заданного, ввод ограничения) от общего числа детских печатных изданий.*

*CREATE FUNCTION part\_cheap(integer) RETURNS float AS*

*$$*

*Declare a float;*

*begin*

*SELECT (SELECT COUNT(editions.price)\*100*

*FROM editions WHERE editions.price <= $1)/*

*(SELECT COUNT(editions.price) FROM editions) INTO a;*

*RETURN a;*

*end;*

*$$*

*language 'plpgsql';*

select \*from *part\_cheap* (1000);

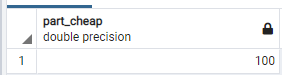


Рисунок 17. Функция part\_cheap()

## Доля детских печатных изданий, проданной за определенной период (ввод).

CREATE FUNCTION part\_sale\_date (date, date) RETURNS float AS

$$

Declare a float;

begin

SELECT (SELECT COUNT(sales.s\_date)\*100

FROM sales JOIN editions ON sales.id\_s = editions.id\_s

WHERE sales.s\_date BETWEEN $1 AND $2)/

(SELECT COUNT(sales.s\_date) FROM sales) INTO a;

RETURN a;

end;

$$

language 'plpgsql';

select \*from part\_sale\_date('2022-04-01', '2022-11-12');

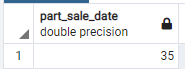


Рисунок 18. Функция part\_sale\_date()

## *Все виды детских печатных изданий, поступивших из заданного издательства (ввод издательства), чья стоимость больше заданной (ввод стоимости.)*

*CREATE FUNCTION find\_b\_p (name, integer) RETURNS table (name\_e name, publisher name, price integer) AS*

*$$*

*begin*

*RETURN QUERY*

*SELECT editions.name\_e, publisher.name\_p, editions.price*

*FROM editions*

*JOIN publisher ON editions.id\_s = publisher.id\_e*

*WHERE publisher.name\_p = $1 AND publisher.price => $2 ;*

*end;*

*$$*

*language 'plpgsql';*

select \*from find\_b\_p ('Publishing house "Eksmo"', 15);

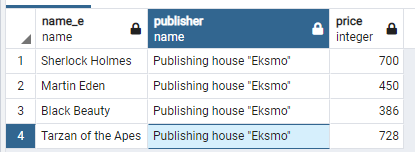


Рисунок 19. Функция find\_b\_p()

## Все детские печатные издания, поступившие от заданного издательства, чья стоимость больше, чем средняя стоимость заданного детского печатного издания (выбор наименования), поступившего из заданной страны (выбор страны).

*CREATE OR REPLACE FUNCTION ed\_country\_all (name, name) RETURNS table(name\_e name, publisher name, price integer) AS*

$$

Begin

RETURN QUERY

SELECT editions.name\_e, publisher.name\_p, editions.price

FROM editions, publisher

WHERE (editions.id\_s = publisher.id\_e

AND publisher.name\_p = $1

AND editions.price >

(SELECT avg(editions.price) FROM editions, publisher, countries

WHERE (editions.id\_s = publisher.id\_e

AND publisher.id\_e= countries.id\_c

AND countries.name = $2)));

Return;

end;

$$

language 'plpgsql';

select \*from ed\_country\_all ('Education', 'Denmark');

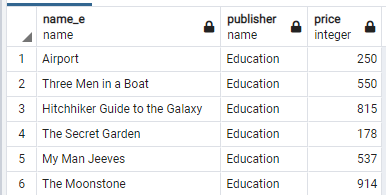


Рисунок 20. Функция ed\_country\_all()

## *Доля дешевых детских печатных изданий (чья стоимость меньше заданной, ввод стоимости), поступивших от заданного издательства и в целом.*

* От заданного издательства

CREATE OR REPLACE FUNCTION part\_cheap\_p(int, name) RETURNS float AS

$$

declare a float;

begin

SELECT (SELECT COUNT(editions.price)\*100

FROM editions JOIN publisher ON editions.id\_s = publisher.id\_e

WHERE editions.price <= $1

AND publisher.name\_p = $2)/

(SELECT COUNT(editions.price) FROM editions) INTO a;

RETURN a;

end;

$$

language 'plpgsql';

select \*from part\_cheap\_p('1000', 'Education');

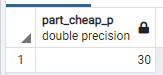


Рисунок 21. Функция ed\_country\_all()

* в целом

CREATE FUNCTION part\_cheap\_all(integer) RETURNS float AS

$$

Declare a float;

begin

SELECT (SELECT COUNT(editions.price)\*100

FROM editions WHERE editions.price <= $1)/

(SELECT COUNT(editions.price) FROM editions) INTO a;

RETURN a;

end;

$$

language 'plpgsql';

select \*from part\_cheap\_all('500');

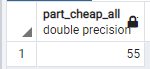


Рисунок 22. Функция part\_cheap\_all()

## *Средняя стоимость детских печатных изданий, проданных за определённый промежуток времени (ввод интервала).*

*CREATE OR REPLACE FUNCTION avg\_b\_price(date, date) RETURNS float AS*

*$$*

*declare a float;*

*begin*

*SELECT avg(*editions*.price)*

*FROM* editions *JOIN Sales ON* editions*.id\_s = Sales.id\_s*

*WHERE Sales.s\_date BETWEEN $1 AND $2 INTO a;*

*RETURN a;*

*end;*

*$$*

*language 'plpgsql';*

select \*from avg\_b\_price('2021-01-01','2022-01-01');

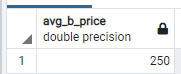


Рисунок 23. Функция avg\_b\_price()

## *Все детские печатные издания, чья стоимость выше, чем средняя стоимость детских печатных изданий заданного издательства.*

*CREATE OR REPLACE FUNCTION avg\_c\_price(name) RETURNS table (name\_e name, price integer) AS*

*$$*

*Begin*

*RETURN QUERY*

*SELECT editions.name\_e, editions.price*

*FROM editions*

*WHERE editions.price > (SELECT avg(editions.price)*

*FROM editions JOIN publisher ON editions.id\_s = publisher.id\_e*

*WHERE publisher.name\_p = $1)*

*ORDER BY editions.price;*

*end;*

*$$*

*language 'plpgsql';*

select \*from avg\_c\_price('Publishing Center "Academy"');



Рисунок 24. Функция avg\_c\_price()

* Находим самые популярные детские печатные издания

*CREATE FUNCTION pop\_all() RETURNS table (name\_e name, popularity bigint) AS*

*$$*

*Begin*

*RETURN QUERY*

*SELECT editions.name\_e, COUNT(Sales.s\_date) AS pop*

*FROM Sales JOIN editions ON Sales.id\_s = editions.id\_s*

*GROUP BY editions.name\_e*

*ORDER BY pop DESC*

*LIMIT 1;*

*end;*

*$$*

*language 'plpgsql';*

*select \*from pop\_all();*

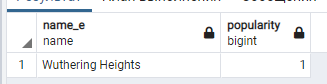


Рисунок 25. Функция *pop\_all* ()

## *Детские печатные издания, которые лучше всего раскупают, указать параметры (стоимость, издательство и т.д.).*

CREATE OR REPLACE FUNCTION best\_sell() RETURNS table (name\_e name, price integer, publisher name) AS

$$

Begin

RETURN QUERY

SELECT editions.name\_e, editions.price, publisher.name\_p

FROM editions

JOIN publisher ON editions.id\_s = publisher.id\_e

WHERE editions.name\_e = (SELECT pop\_all.name\_e from pop\_all());

end;

$$

language 'plpgsql';

*select \*from* best\_sell *();*

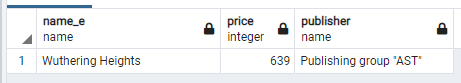


Рисунок 26. Функция best\_sell ()

# **Создание таблицы логов:**

create table logs(operation char(6) not null, change\_time timestamp, users text, table\_name name, strings text not null);

Создание функции записи в лог add\_to\_logs():

create or replace function add\_to\_logs() returns trigger as $$

begin

if(tg\_op = 'DELETE') then insert into logs select 'delete', now(), user, TG\_RELNAME, OLD;

return OLD;

elseif(tg\_op = 'UPDATE') then insert into logs select 'update', now(), user, TG\_RELNAME, NEW;

return NEW;

elseif(tg\_op = 'INSERT') then insert into logs select 'insert', now(), user, TG\_RELNAME, NEW;

return NEW;

end if;

return null;

end;

$$ language 'plpgsql';

Создание триггеров для таблиц:

create trigger log\_trigger after insert or update or delete on publisher

for each row execute procedure add\_to\_logs();

create trigger log\_trigger after insert or update or delete on countries

for each row execute procedure add\_to\_logs();

create trigger log\_trigger after insert or update or delete on sales

for each row execute procedure add\_to\_logs();

create trigger log\_trigger after insert or update or delete on editions

for each row execute procedure add\_to\_logs();

Создание роли оператора, пользователя и администратора:

*Create role operator with password '12345' nocreatedb nocreaterole login;*

*Create role user1 with password '12345' nocreatedb nocreaterole login;*

*Create role administrator with password '12345' nocreatedb createrole login;*

Оператор может просматривать, удалять, обновлять и добавлять записи только в справочники:

grant insert, delete, update, select

on table Countries,Sales

to operator;

Пользователь может просматривать, удалять, обновлять и добавлять записи в таблицах, кроме справочников:

grant insert, delete, update, select, connect

on table Editions, Publisher

to user1;

Администратор может совершать любые действия над всеми таблицами:

grant all privileges

on table Countries,Sales, Editions, Publisher

to administrator;

update Countries set name=' **Zimbabwe**' where id\_c=1;

select \*from Countries;

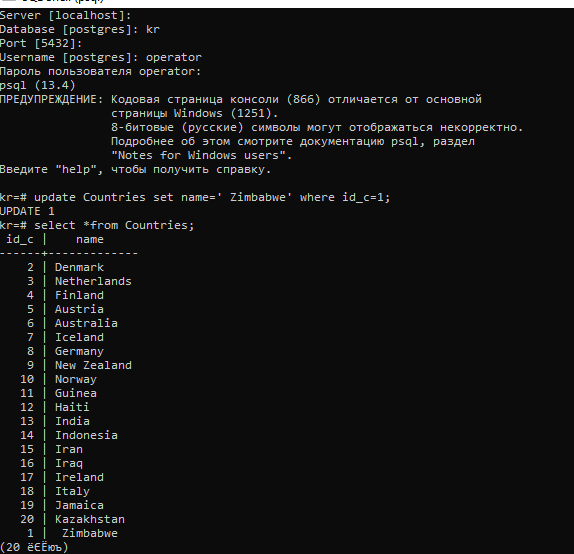


Рисунок 27. – Изменение оператором таблицы-справочника и его просмотр.

update Publisher set name\_p=' **VOT**' where id\_c=1;

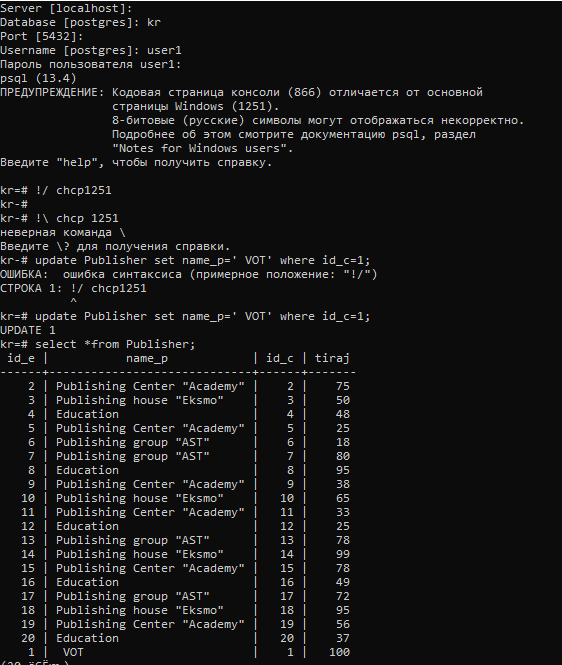


Рисунок 28. – Изменение пользователем таблицы и его просмотр.

select \*from logs;

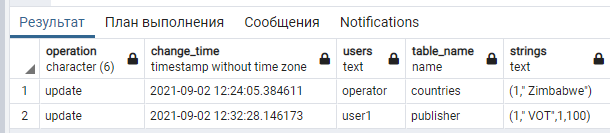


Рисунок 29. – просмотр таблицы логов администратором.

# Выводы:

В результате выполнения курсовой работы была построена база данных в СУБД PostgreSQL с таблицами, в соответствии с вариантом, а именно база данных «Детские издания» с таблицами «Издания», «Издательство», «Продажи», «Страны». Для данной базы данных была представлена структура таблиц и связей между ними. Были выполнены все поставленные задачи: написаны функции для выполнения запросов с использованием процедурного языка plpgsql, а также приведены примеры результатов выполнения функций.

Для реализации курсовой работы были применены знания и навыки разработки БД, выполнения запросов, написания функций, полученные в ходе выполнения лабораторных работ по курсу и изучения документации PostgreSQL.

# Список использованной литературы

1. Документация по PostgreSQL. Postgres Professional [электронный ресурс] – Открытый – URL: <https://postgrespro.ru/docs/postgrespro/9.5/tutorial-join> (дата обращения 25.08.2021)
2. 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
3. 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. - Загл. с экрана – (Дата обращения: 01.09.2021 )
4. Документация PostgreSQL и Postgres Pro // PostgresPro// [электронный ресурс]// URL: https://postgrespro.ru/docs/ (дата обращения: 02.09.2021).
5. 3.PostgreSQL Tutorial - Learn PostgreSQL from Scratch // PostgreSQL Tutorial// [электронный ресурс]// URL: http://www.postgresqltutorial.com/ (дата обращения: 20.12.2020).