МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра вычислительной техники

**Курсовая работа**

По дисциплине: «База данных»

Тема: «Разработка базы данных “Детские издания”»

Вариант 89

|  |  |
| --- | --- |
| Факультет: АВТФ  Группа: АВТ-943  Студент: Фокин Е.К. | Преподаватель: Трошина Г. В. |

Новосибирск 2021 г.

**Оглавление**

[Задание 3](#_Toc89375820)

[Введение 5](#_Toc89375821)

[Цель работы 6](#_Toc89375822)

[1. Графическое представление связей между таблицами 6](#_Toc89375823)

[2. Структура таблиц базы данных 7](#_Toc89375824)

[3. Содержание таблиц баз данных 9](#_Toc89375825)

[4. Руководство пользователя 16](#_Toc89375826)

[5. Тексты запросов и примеры работы 18](#_Toc89375827)

[**5.1** **Создание запросов** 18](#_Toc89375828)

[**5.2** **Создание функций** 22](#_Toc89375829)

[**5.3** **Добавление ролей** 31](#_Toc89375830)

[**5.4** **Проверка прав ролей** 33](#_Toc89375831)

[Анализ результатов и выводы 38](#_Toc89375832)

[Список литературы 39](#_Toc89375833)

# Задание

Тема: «Разработка базы данных «Детские 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 БД. Реализовать следующие типы запросов с использованием нескольких таблиц:

* Для каждого детского печатного издания указать сведения о нем (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 детские 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 определенный 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 определенный 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.).

# Введение

Базы данных — это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия.

Базы данных стали очень широко распространенной технологией, что потребовало, в свою очередь, большего числа специалистов, способных проектировать их и обслуживать. В ходе эволюции теории и практики баз данных стандартом стала реляционная модель данных, а в рамках этой модели сформировался и специализированный язык программирования, позволяющий выполнять все необходимые операции с данными — Structured Query Language (SQL). Таким образом, важным компонентом квалификации специалиста в области баз данных является владение языком SQL.

Данная работа посвящена разработке базы данных на языке PostgreSQL, формирование таблиц. В рамках данной работы будут реализованы таблицы справочники и таблицы, использующие справочники. Будут реализованы роли пользователей, такие как: администратор, оператор, пользователь базы данных.

Будут рассматриваться создание многотабличных запросов, подзапросов и функций.

# Цель работы

Закрепление навыков проектирования и создания баз данных, а также работы с *PostgreSQL*.

# Графическое представление связей между таблицами

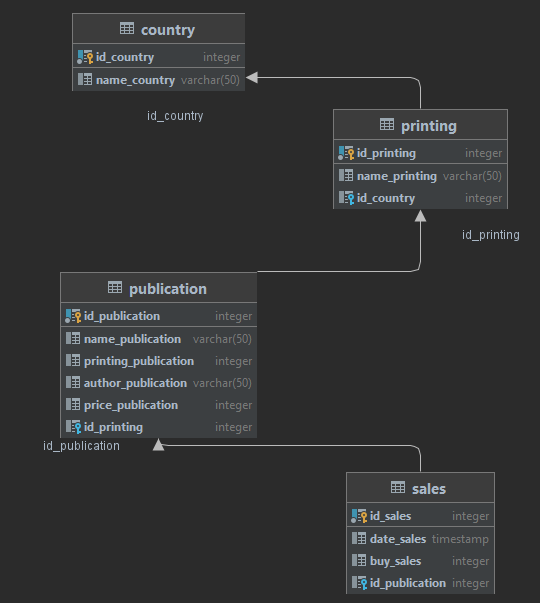


Рисунок 1 – Графическое представление связей между таблицами.

# Структура таблиц базы данных

Таблица *country* является таблицей-справочником. В ней хранится информация о странах, которые выпускают детские издания

Таблица 1. Типы данных и смысловое содержание полей таблицы *country*.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Ограничения** | **Пояснения** |
| id\_country | integer | PRIMARY KEY | Идентификатор страны |
| name\_country | varchar | – | Название страны |

Таблица *printing* является таблицей. В ней хранится информация о изданиях.

Таблица 2. Типы данных и смысловое содержание полей таблицы *printing*.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Ограничения** | **Пояснения** |
| id\_printing | integer | PRIMARY KEY | Идентификатор издателя |
| name\_printing | varchar | – | Название издателя |
| id\_country | integer | REFERENCES(id\_country) | Идентификатор страны |

Таблица *publication* является таблицей. В ней хранится информация о публикациях детских печатных изданий.

Таблица 3. Типы данных и смысловое содержание полей таблицы *publication.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Ограничения** | **Пояснения** |
| id\_publication | integer | PRIMARY KEY | Идентификатор публикации |
| name\_publication | varchar | – | Название публикации |
| printing\_publication | integer | – | Кол-во выпущенных книг(тираж) |
| author\_publication | varchar | – | Автор публикации |
| price\_publication | integer | – | Стоимость публикации |
| id\_printing | integer | REFERENCES(id\_printing) | Идентификатор издателя |

Таблица *sales* использует таблицу. В ней хранится информация о поставщиках.

Таблица 4. Типы данных и смысловое содержание полей таблицы *sales.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Ограничения** | **Пояснения** |
| id\_sales | integer | PRIMARY KEY | Идентификатор покупки |
| buy\_sales | integer | – | Кол-во купленных детских печатных изданий |
| date\_sales | timestamp | – | Дата покупки детских печатных изданий |
| id\_publication | integer | REFERENCES(id\_publication) | Идентификатор публикации |

# Содержание таблиц баз данных

Для начало создаём базу данных *children\_publication* при помощи команды CREATE DATABASE <название> (рисунок 2).



Рисунок 2 – Создание базы данных.

После этого создадим таблицу *country,* использующую таблицы-справочники. Создаём её при помощи команды CREATE TABLE <название> (<поля таблицы>) (рисунок 3).

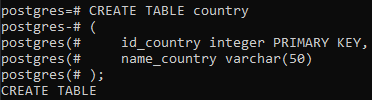


Рисунок 3 – Создание таблицы *country.*

Создаём таблицу *printing*, использующую таблицы-справочники *country* (рисунок 4).

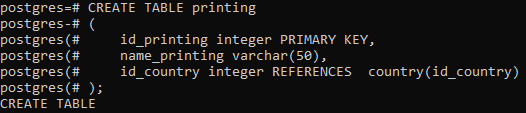


Рисунок 4 – Создание таблицы *printing*.

Создаём таблицу *publication,* использующую таблицу-справочник *printing* (рисунок 5).

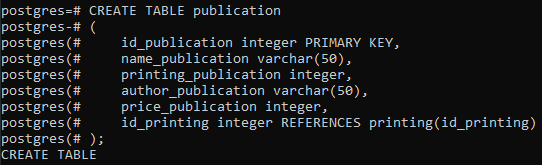


Рисунок 5 – Создание таблицы *publication*.

Создаём таблицу *sales*, использующую таблицы-справочники *publication* (рисунок 6).

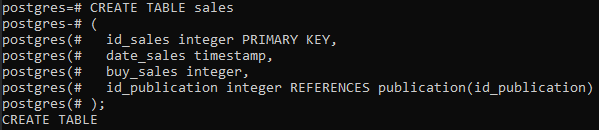


Рисунок 6 – Создание таблицы *sales*.

Демонстрация добавленных таблиц (рисунки 7-10).

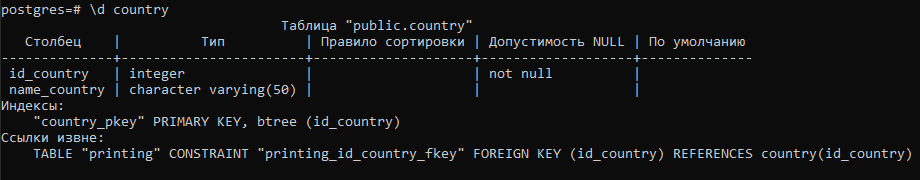


Рисунок 7 – Таблица *country*.

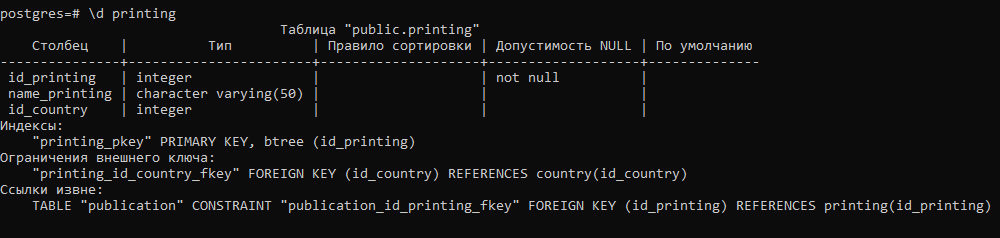


Рисунок 8 – Таблица *printing*.

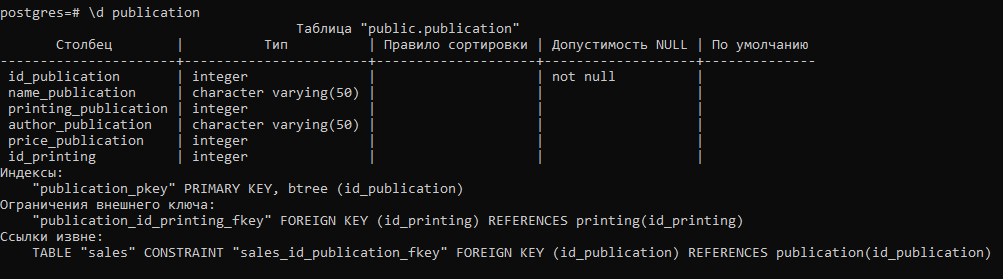


Рисунок 9 – Таблица *publication*.

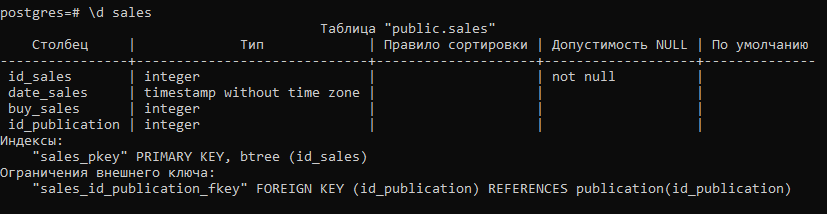


Рисунок 10 – Таблица *sales*.

Перейдём к заполнению таблиц. Заполнение таблиц происходит при помощи команды INSERT INTO <название\_таблицы> (поля\_для\_заполнения) VALUES (значения\_полей). Просмотр содержимого при помощи реализуется при помощи команды SELECT \* FROM <название\_таблицы>.

Заполним таблицу *country*, в которой хранится информация о странах (рисунок 11). И сразу посмотрим содержимое таблицы (рисунок 12).

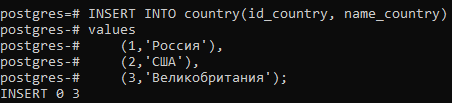


Рисунок 11 – Заполнение таблицы *country*.

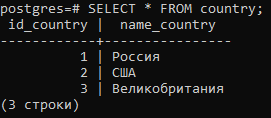


Рисунок 12 – Содержание таблицы *country*.

Заполним таблицу *printing*, в которой хранится информация о издателей детских печатных изданий (рисунок 13). Затем выведем содержание данной таблицы (рисунок 14).

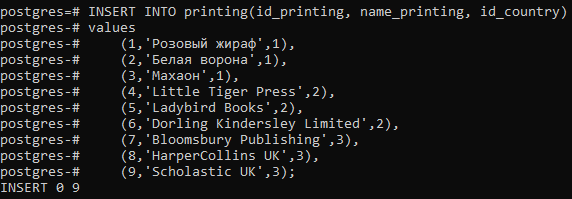


Рисунок 13 – Заполнение таблицы *printing*.

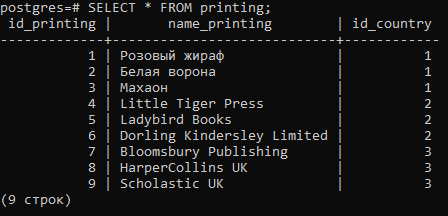


Рисунок 14 – Содержание таблицы *printing*.

Заполним таблицу *publication,* в которой хранится информация о публикациях детских печатных изданий (рисунок 15). Затем выведем содержимое на экран (рисунок 16).

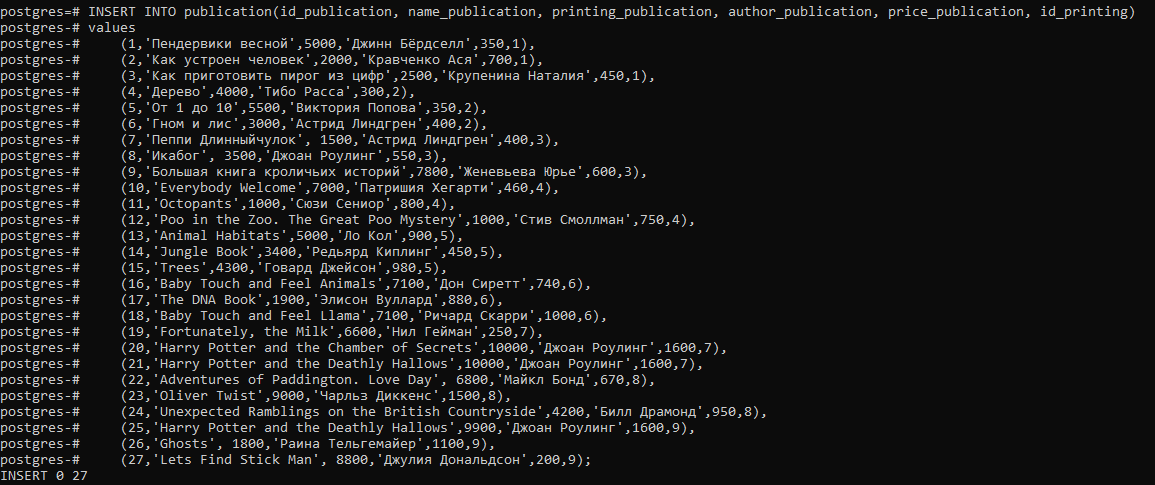


Рисунок 15 – Заполнение таблицы *publication*.

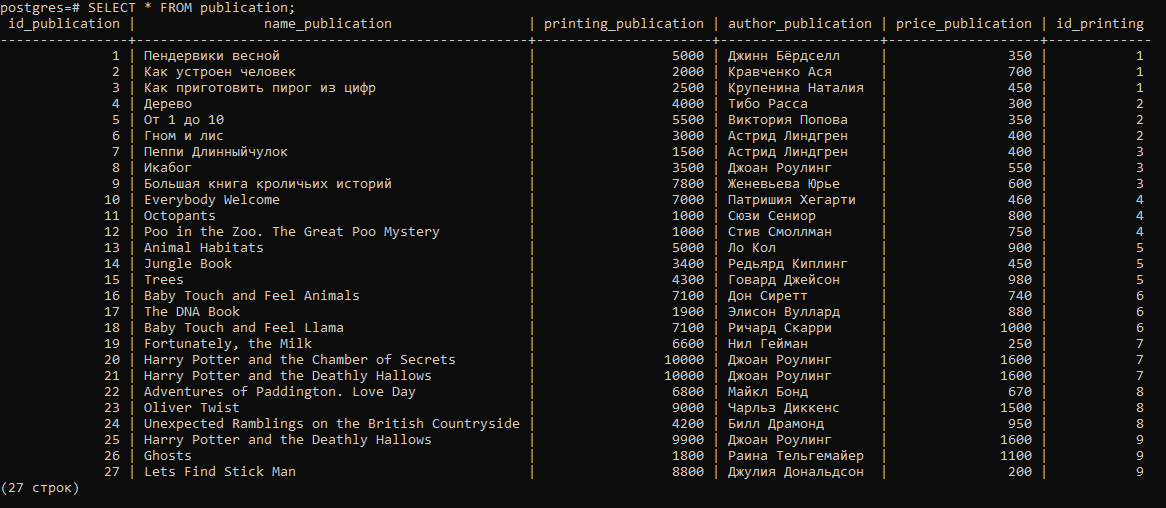


Рисунок 16 – Содержимое таблицы *publication*.

Заполним таблицу *sales*, в которой хранится информация о покупках детских печатных изданиях (рисунок 17). После этого выведем содержимое на экран (рисунок 18).

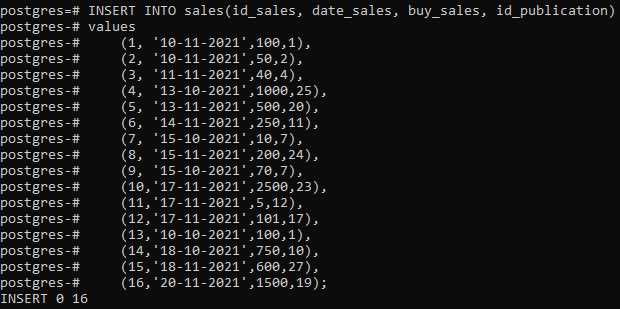


Рисунок 17 – Заполнение таблицы *sales*.

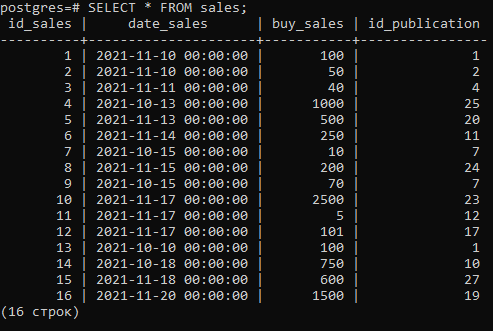


Рисунок 18 – Содержимое таблицы *sales*.

# Руководство пользователя

Таблица 9. Описание функционала.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название функции** | **Передаваемые аргументы** | **Описание функции** |
| 1 | find\_name\_publication\_with\_price\_over(integer) | * integer($1) – цена | Найти издания с ценой свыше введённой |
| 2 | find\_publication\_by\_printing\_interval(integer, integer) | * integer($1, $2) – интервал тиража | Найти все детские печатные издания, чей тираж попадает в заданные пределы (ввод интервала). |
| 3 | find\_printing(integer) | * integer($1) – выбор издательства | Для заданного издательства (выбор) найти все детские печатные издания. |
| 4 | find\_printing\_by\_price(integer, integer, integer) | * integer($1) – выбор издательства * integer($2,$3) – ввод интервала стоимости | Найти все детские печатные издания, чья стоимость находится в заданных пределах (ввод интервала) для заданного издательства. |
| 5 | find\_child\_publication\_by\_all\_publication(integer) | * integer($1) – ввод цены | Найти долю дешевых детских печатных изданий (меньше заданного, ввод ограничения) от общего числа детских печатных изданий. |
| 6 | find\_publication\_by\_date(date, date) | * date($1, $2) – ввод ограничения даты | Найти долю детских печатных изданий, проданной за определенный период (ввод). |
| 7 | find\_printing\_and\_publication\_by\_price(integer, integer) | * integer($1) – ввод издательства * integer($2) – ввод стоимости | Найти все виды детских печатных изданий, поступивших из заданного издательства (ввод издательства) чья стоимость больше заданной (ввод стоимости). |
| 8 | find\_child\_publication\_from\_country(integer,integer) | * integer($1) – выбор издателя * integer($2) – выбор страны | 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). |
| 9 | find\_child\_publication\_by\_publication(integer, integer) | * integer($1) – выбор издателя * integer($2) – выбор цены | Найти долю дешевых детских печатных изданий (чья стоимость меньше заданной, ввод стоимости), поступивших от заданного издательства и в целом. |
| 10 | find\_avg\_for\_date(date, date) | * date($1,$2) – ввод промежутка времени | Найти среднюю стоимость детских печатных изданий, проданных за определенный промежуток времени (ввод интервала). |
| 11 | find\_all\_publication\_by\_avg\_publication(integer) | * integer($1) – выбор цены | Найти все детские печатные издания, чья стоимость выше, чем средняя стоимость детских печатных изданий заданного издательства |

# Тексты запросов и примеры работы

## **Создание запросов**

Создадим запрос, в котором выведем фототовары отсортированные по виду и дате выпуска при помощи функции SELECT <поля\_таблиц> FROM <название\_таблицы> INNER JOIN <присоединяемая таблица> ON <> <связь\_между\_таблицами> ORDER BY <поля\_для\_сортировки> .

Создадим запрос, в котором получим список, отсортированный в алфавитном порядке по наименованию (рисунок 19).

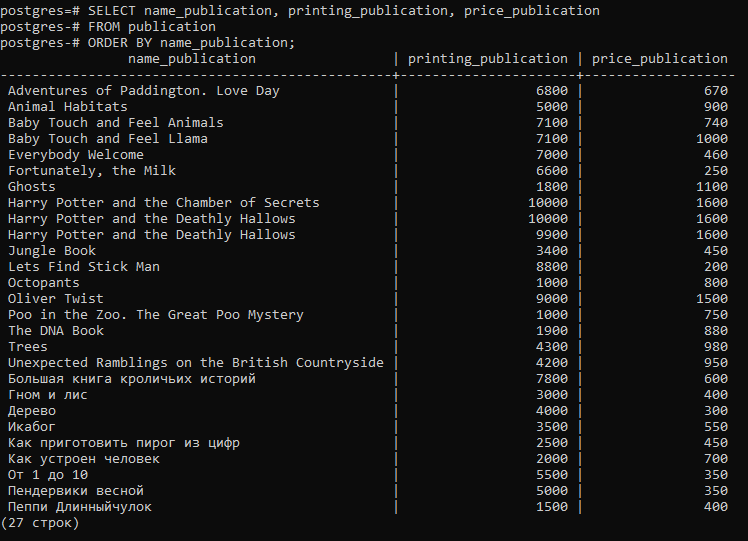


Рисунок 19 – Книжные издания, отсортированные в алфавитном порядке

Создадим запрос, в котором получим список, отсортированный по тиражу (рисунок 20).

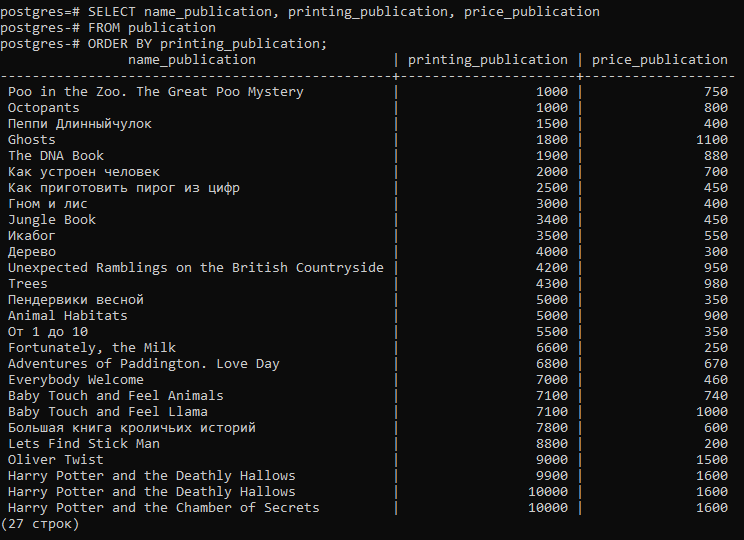


Рисунок 20 – Книжные издания, отсортированные по тиражу

Создадим запрос, в котором получим список, отсортированный по стоимости (рисунок 21).

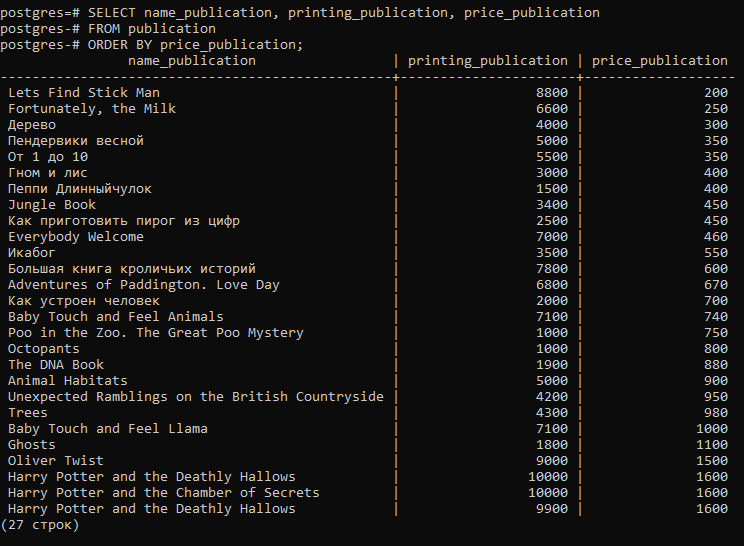


Рисунок 21 – Книжные издания, отсортированные по стоимости

Создадим запрос, в котором выведем самое дорогое издание (рисунок 22).

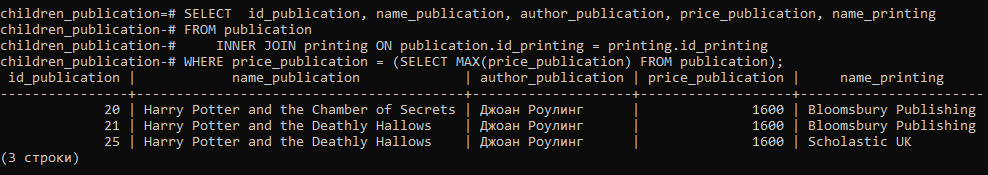


Рисунок 22 – Самые дорогие издания.

Создадим запрос, в котором выведем самое дешевое издание (рисунок 23).

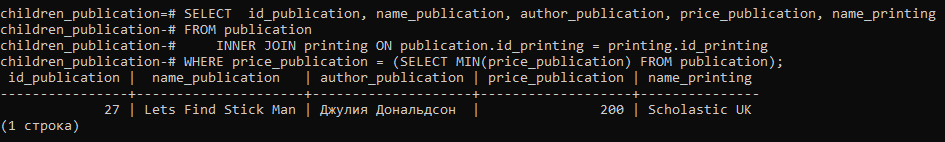


Рисунок 23 – Самые дешевые издания

Создадим запрос, в котором выведем среднюю стоимость всех изданий (рисунок 24).

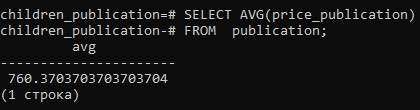


Рисунок 24 – средняя стоимость всех изданий

Создадим запрос, в котором найдем детские печатные издания, которые лучше всего раскупают (рисунок 25).

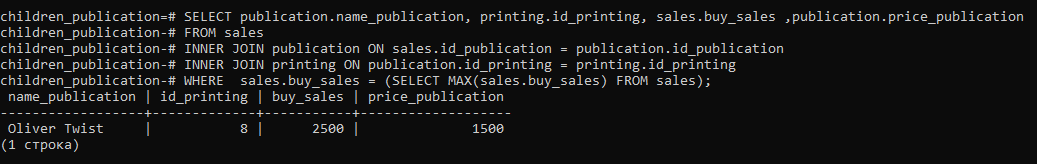


Рисунок 25 – поиск печатных изданий, которые лучше всего раскупают

## **Создание функций**

Создадим функцию *find\_name\_publication\_with\_price\_over(integer)* которая будет выводить искать печатные издания с ценой, выше указанной (рисунок 26).

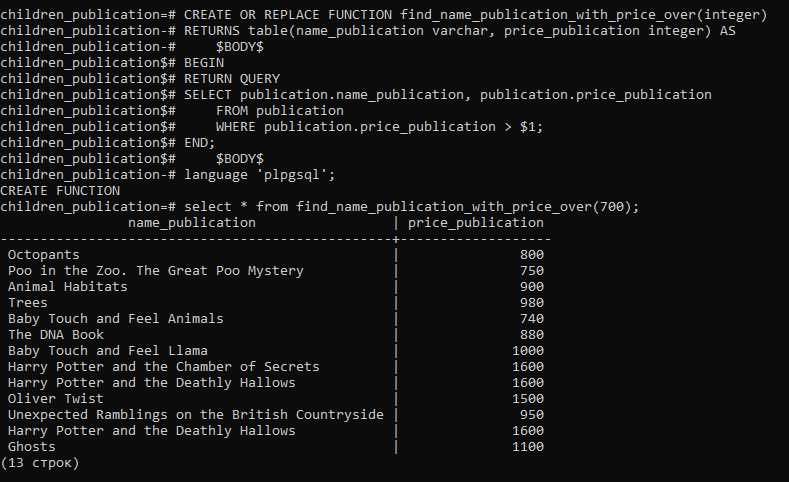


Рисунок 26 – Создание функции *find\_name\_publication\_with\_price\_over (integer)* и демонстрация её работы на примере.

Напишем функцию *find\_publication\_by\_printing\_interval(integer, integer)* которая будет выводить все детские печатные издания, чей тираж попадает в заданные пределы (рисунок 27).

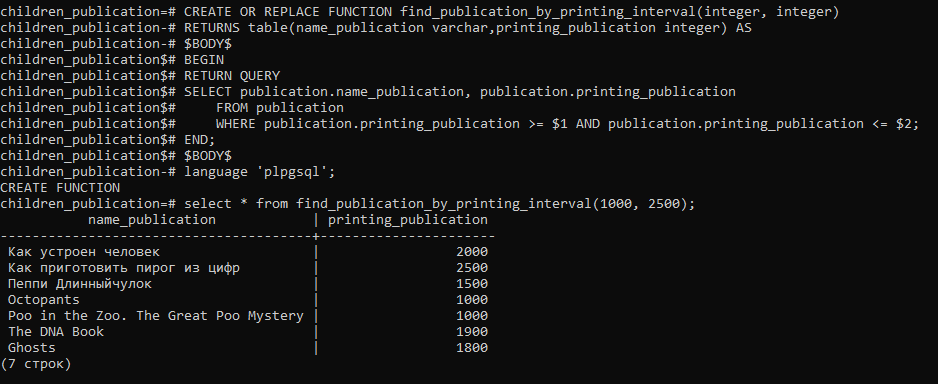


Рисунок 27 – Создание функции *find\_publication\_by\_printing\_interval(integer, integer)* и демонстрация её работы на примере.

Создадим функцию *find\_printing(varchar)*, которая позволяет найти все детские печатные издания для заданного издательства (рисунок 28).

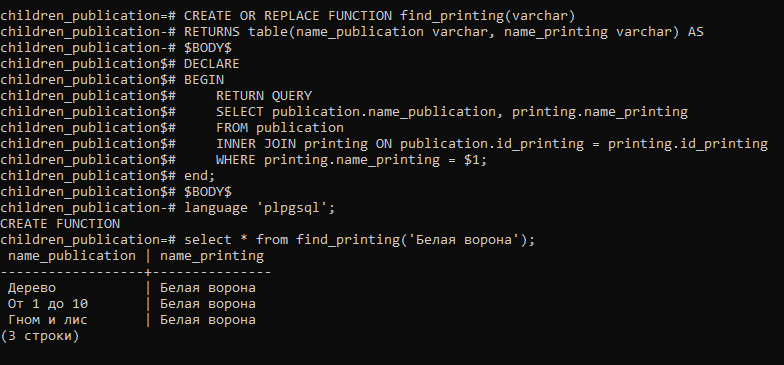


Рисунок 28 – Создание функции *find\_printing(varchar) и демонстрация её работы.*

Создание функции *find\_printing\_by\_price(varchar, integer, integer)*, которая позволяет найти все детские печатные издания, чья стоимость находится на заданных пределах, для заданного издательства (рисунок 29).

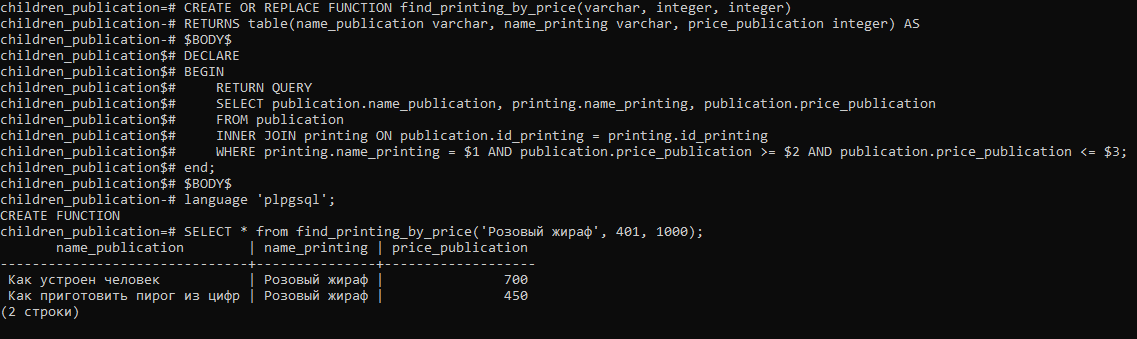


Рисунок 29 – Создание функции *find\_printing\_by\_price(varchar, integer, integer)*и демонстрация её работы.

Создание функции *find\_child\_publication\_by\_all\_publication(integer)*, которая позволяет найти долю дешевых детских печатных изданий (меньше заданного) от общего числа детских печатных изданий (рисунок 30).

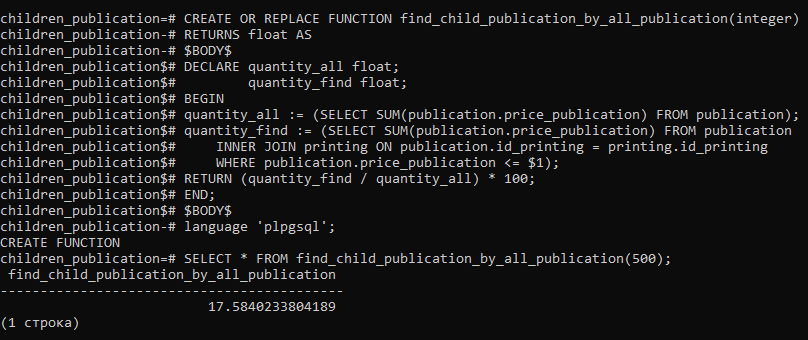


Рисунок 30 – Создание функции *find\_child\_publication\_by\_all\_publication(integer)* и демонстрация её работы.

Также для проверки нашей функции выведем отдельно сумму всех публикаций и сумму публикаций, меньше заданного числа (в нашем примере меньше 500) (рисунок 31)

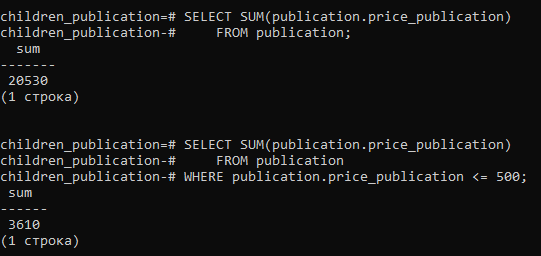


Рисунок 31 – показ сумму публикация и сумму меньше заданного числа

Создадим функцию *find\_publication\_by\_date(date, date)*, которая позволяет найти долю детских печатных изданий, проданной за определенный период (рисунок 32).

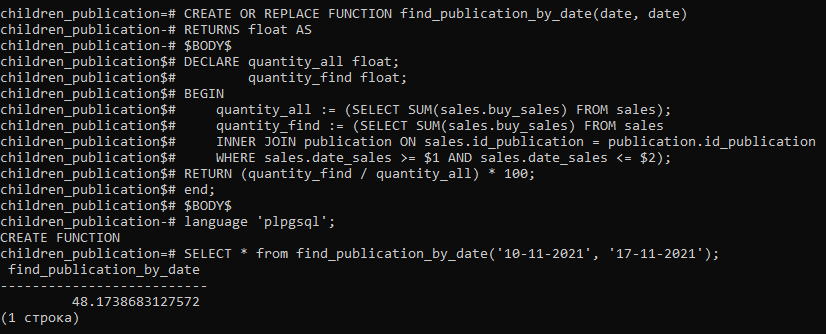


Рисунок 32 – Создание функции *find\_publication\_by\_date(date, date)* и демонстрация работы.

Также для проверки нашей функции выведем отдельно сумму всех проданных печатных изданий и сумму всех проданных печатных изданий за определенный период (рисунок 33).

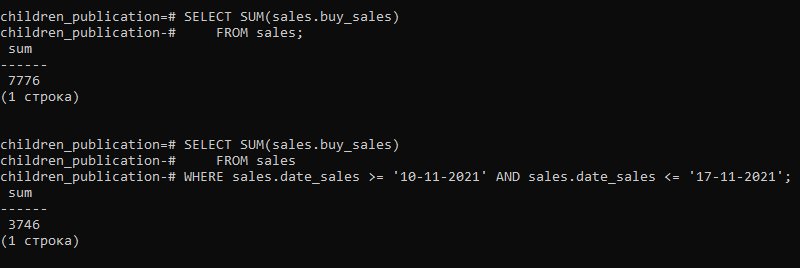


Рисунок 33 – ввод суммы проданных изданий и сумму проданных изданий за определенный период

Создадим функцию *find\_printing\_and\_publication\_by\_price(varchar, integer),* которая позволяет найти все виды детских печатных изданий, поступивших из заданного издательства, чья стоимость больше заданной (рисунок 34).

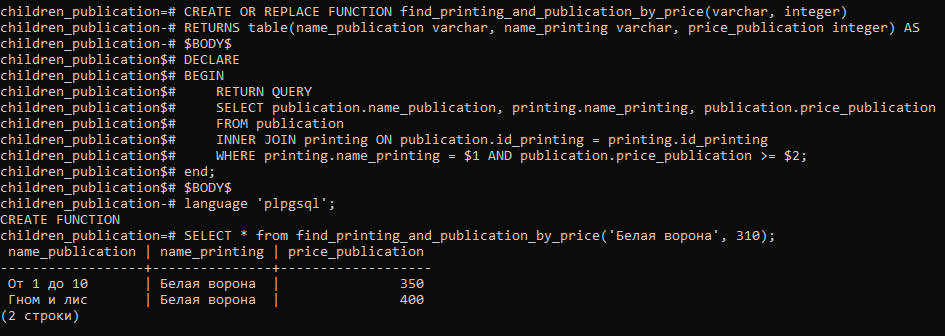


Рисунок 34 – Создание функции *find\_printing\_and\_publication\_by\_price(varchar, integer*) и демонстрация её работы.

Создадим функцию *find\_child\_publication\_from\_country(varchar,varchar)*, которая позволит найти все детские печатные издания, поступившие от заданного издательства, чья стоимость больше, чем средняя стоимость заданного детского печатного издания, поступившего из заданной страны (рисунок 35).



Рисунок 35 – Создание функции *find\_child\_publication\_from\_country(varchar, varchar)* и демонстрация её работы на примере.

Также для проверки нашей функции выведем отдельно среднюю стоимость печатных изданий из выбранной страны (в нашем примере из россии) (рисунок 36).

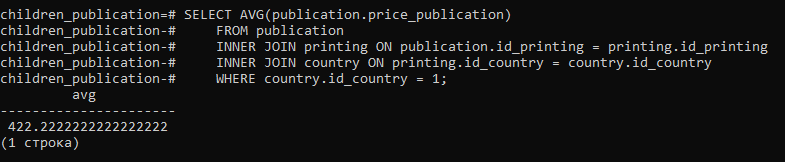


Рисунок 36 – ввод средней стоимости печатных изданий из выбранной страны

Создадим функцию *find\_child\_publication\_by\_publication(integer, integer)*, которая позволяет найти долю дешевых детских печатных изданий, поступивших от заданного издательства (рисунок 37).

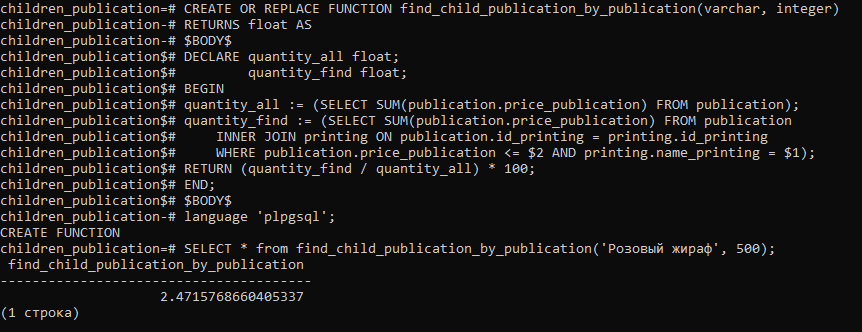


Рисунок 37 – Создание функции *find\_child\_publication\_by\_publication(integer, integer)* и демонстрация её работы.

Также для проверки нашей функции выведем отдельно стоимость всех печатных изданий и стоимость печатных изданий меньше заданного (в данном примере 500) заданного издательства (рисунок 38).

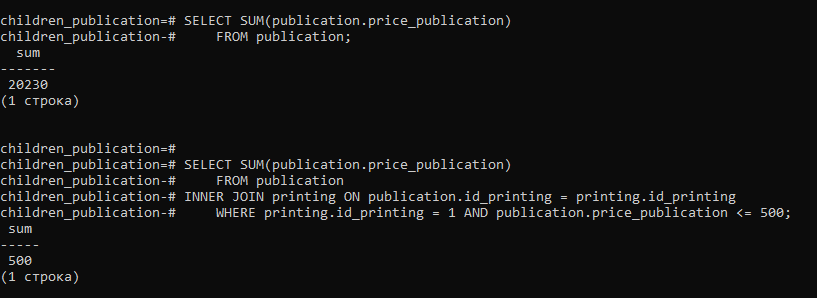


Рисунок 38 – ввод стоимости всех печатных изданий и стоимость печатных изданий меньше заданного

Создадим функцию *find\_avg\_for\_date(date, date)*, которая позволяет найти среднюю стоимость детских печатных изданий, проданных за определенный промежуток времени (рисунок 39).

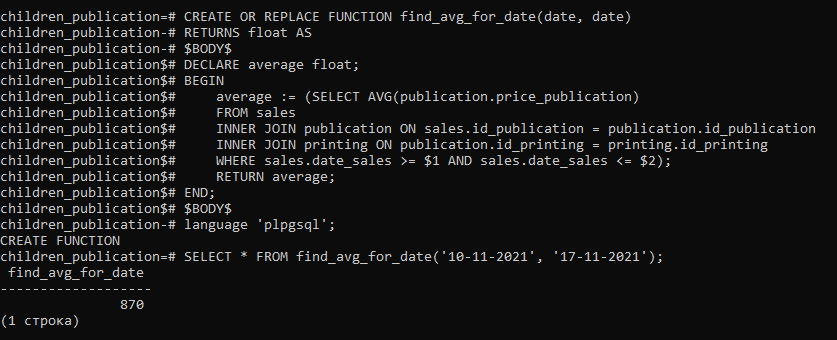


Рисунок 39 – Создание *функции find\_avg\_for\_date(date, date)* и демонстрация её работы.

Создаём функцию *find\_all\_publication\_by\_avg\_publication(varchar)*, которая позволяет найти все детские печатные издания, чья стоимость выше, чем средняя стоимость детских печатных изданий заданного издательства (рисунок 40).

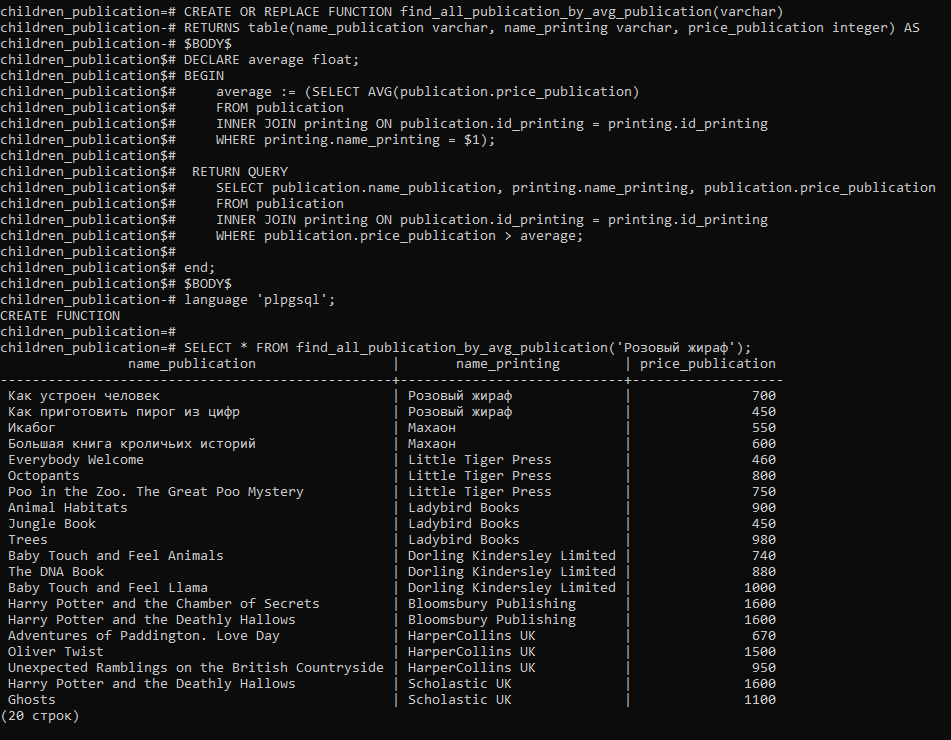


Рисунок 40 – Создание функции *find\_all\_publication\_by\_avg\_publication(varchar)* и демонстрация её работы.

Также для проверки выведем среднюю стоимость публикации заданного издательства (рисунок 41).

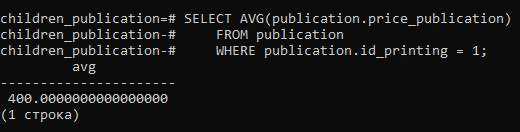


Рисунок 41 – ввод средней стоимости публикации заданного издательства.

## **Добавление ролей**

Создадим роли оператора, пользователя, и администратора (рисунок 42).

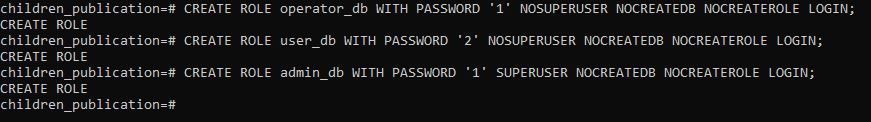


Рисунок 42 – Создание ролей.

Оператор базы данных:

* Не может создавать других пользователей, базы данных и роли;
* Может просматривать, удалять, обновлять и добавлять справочники.

Пользователь базы данных:

* Не может создавать других пользователей, базы данных и роли;
* Может просматривать, удалять, обновлять и добавлять все таблиц, кроме таблицы справочников.

Администратор базы данных:

* Может создавать других пользователей, другие базы данных и роли;

Назначим права оператору базы данных (рисунок 43):

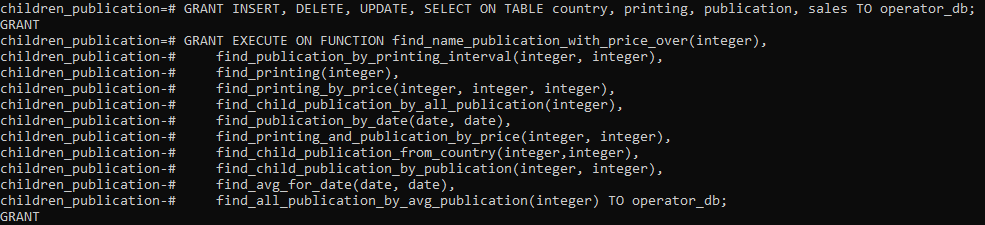


Рисунок 43 – Назначение прав оператору базы данных.

Назначим права пользователю базы данных (рисунок 44):

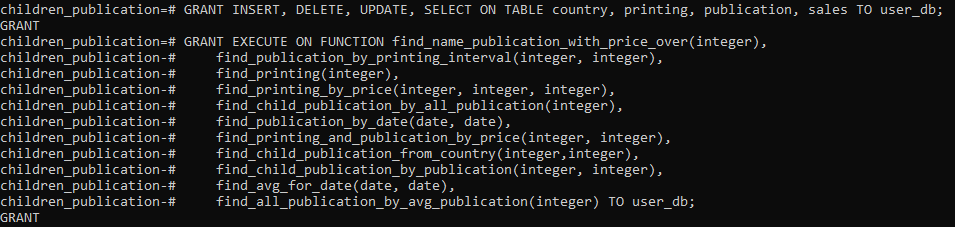


Рисунок 44 – Назначение прав пользователю базы данных.

Назначим права администратору базы данных (рисунок 45):

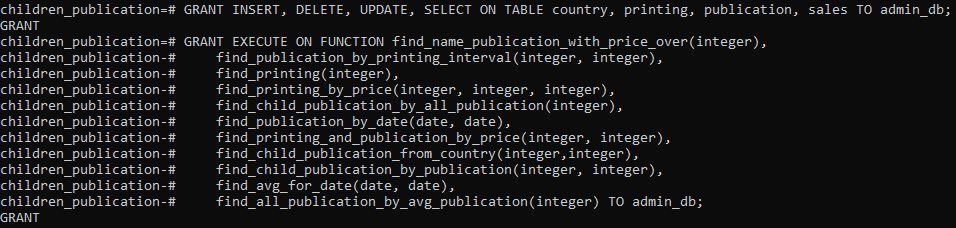


Рисунок 45 – Назначение прав администратору базы данных.

## **Проверка прав ролей**

Продемонстрируем работу ролей. Зайдём под ролью оператора базы данных и попытаемся добавить/удалить/обновить записи таблицы. Попробуем создать роль и использовать одну из функций, к которой у Администратора есть права (рисунок 46-47):

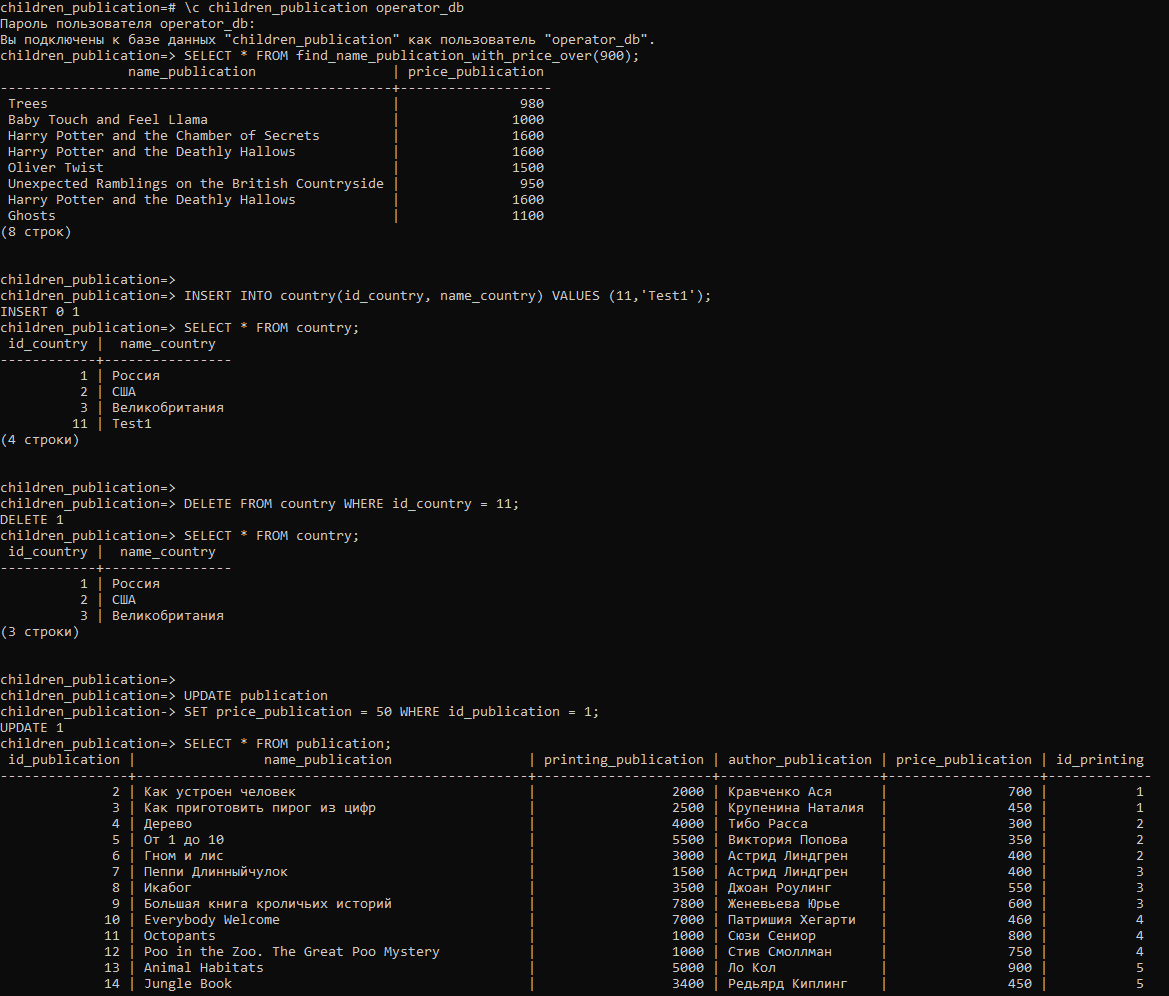


Рисунок 46 – Проверка роли оператора.

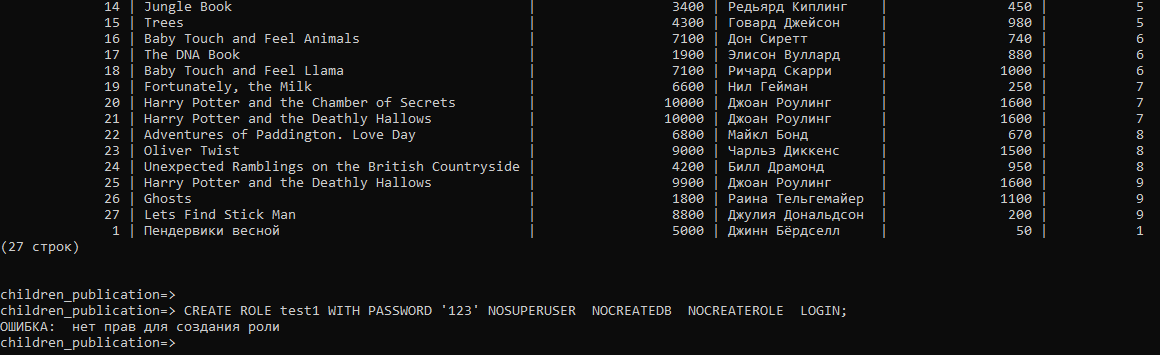


Рисунок 47 – Продолжение демонстрации проверки оператора БД.

Зайдём под ролью пользователя базы данных и попытаемся добавить/удалить/обновить записи в таблице. Попробуем создать роль для проверки (рисунки 48-50):

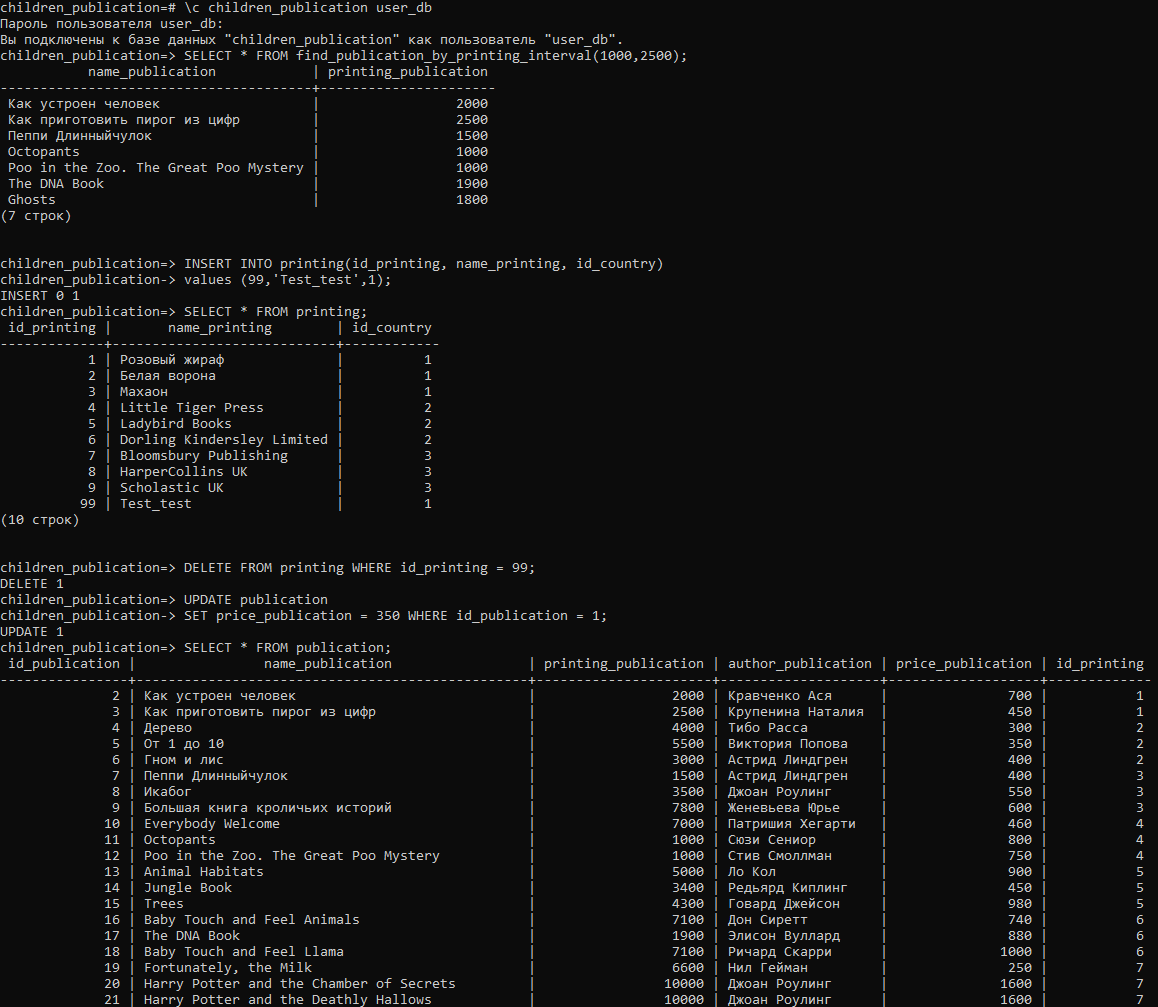


Рисунок 48 – Проверка роли пользователя базы данных.

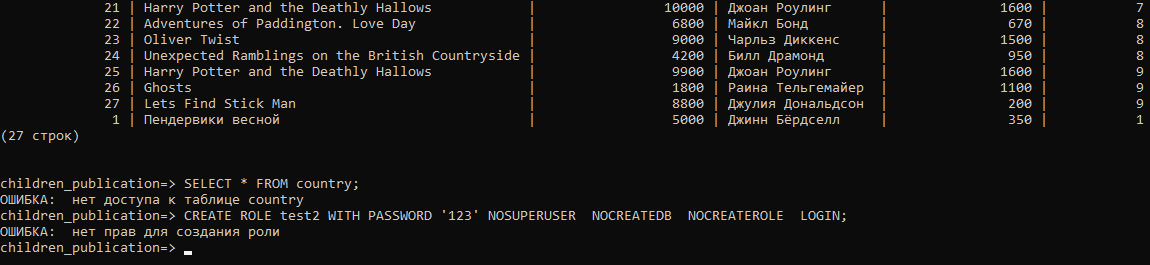


Рисунок 49 – Продолжение демонстрации проверки пользователя БД.

Зайдём под ролью администратора базы данных и просмотрим таблицу. Так же опробуем вставку и удаление, создание роли (рисунок 50-51):

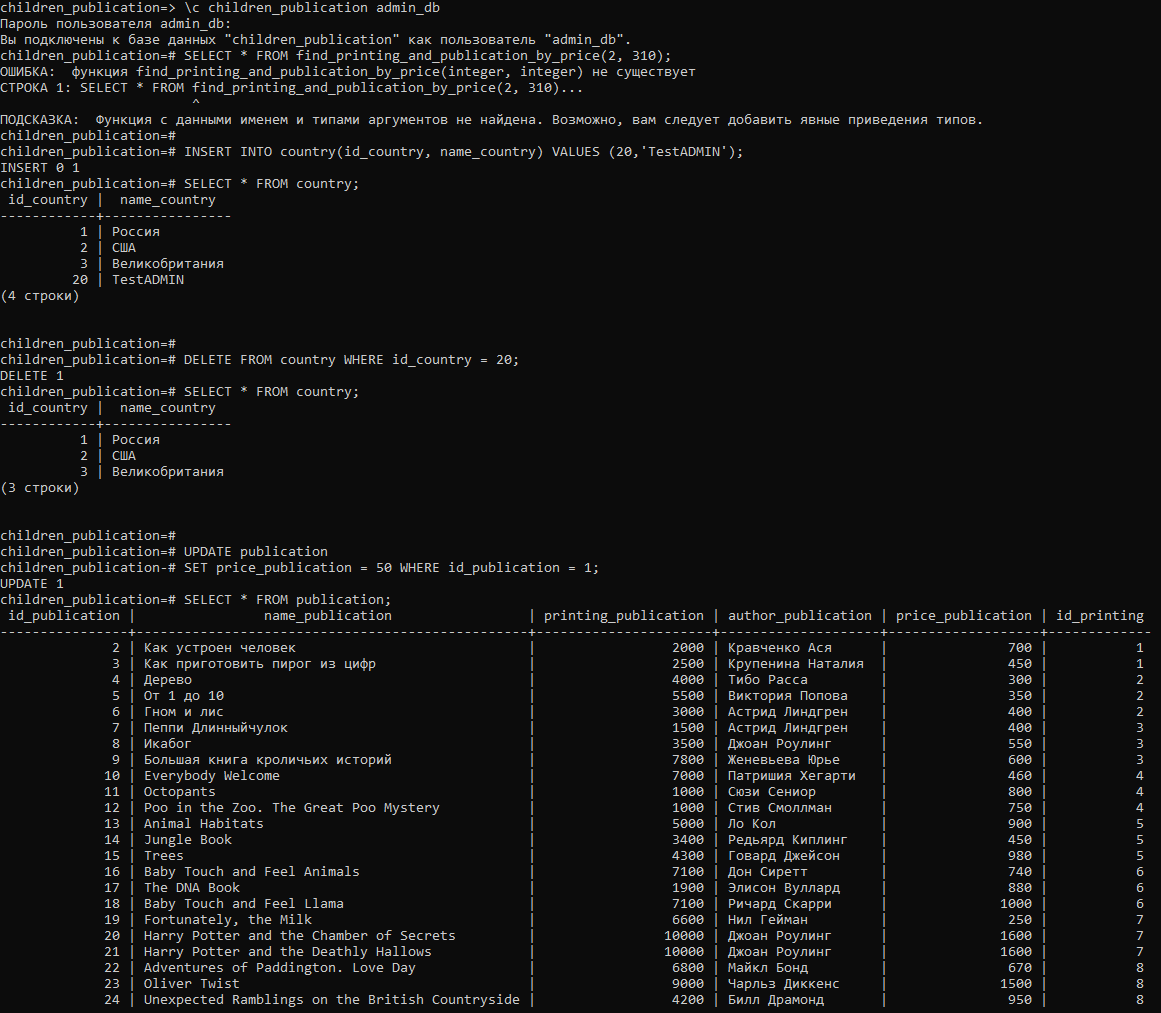


Рисунок 50 – Проверка роли администратора базы данных.

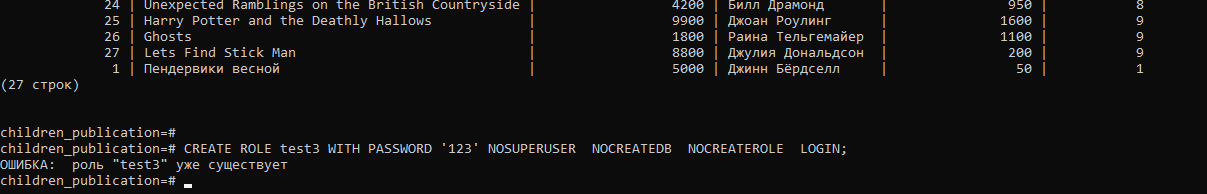


Рисунок 51 – Продолжение демонстрации проверки администратора БД.

# Анализ результатов и выводы

В ходе выполнения курсовой работы была создана и заполнена база данных *children\_publication*. В данной базе данных были созданы 4 таблицы, связанных между собой, а именно: *country, publication, printing, sales*.

Было создано 11 функций для поиска различных величин, которые были протестированы на работоспособность и корректность результата.

Так же для разработанной базы данных были созданы роли – роль оператора, пользователя, администратора. Их различие заключается в правах доступа к базе данных. При проверке прав пользователей ожидаемые результаты сошлись с реальными.

При выполнении работы применялись навыки построения баз данных, написание запросов и функций, создание ролей, полученные в ходе выполнения лабораторных работ по дисциплине «Базы данных».

# Список литературы

1. Трошина Г. В. Электронное учебное пособие \"Базы данных\": учеб. пособие / Г. В. Трошина. – : ИДО НГТУ, 2006.
2. Лузанов П. В. «Postgres: первое знакомство» \ Москва, Постгрес Профессиональный, 2021 - ISBN 978-5-6041193-8-9