**ЛАБОРАТОРНАЯ РАБОТА №3**

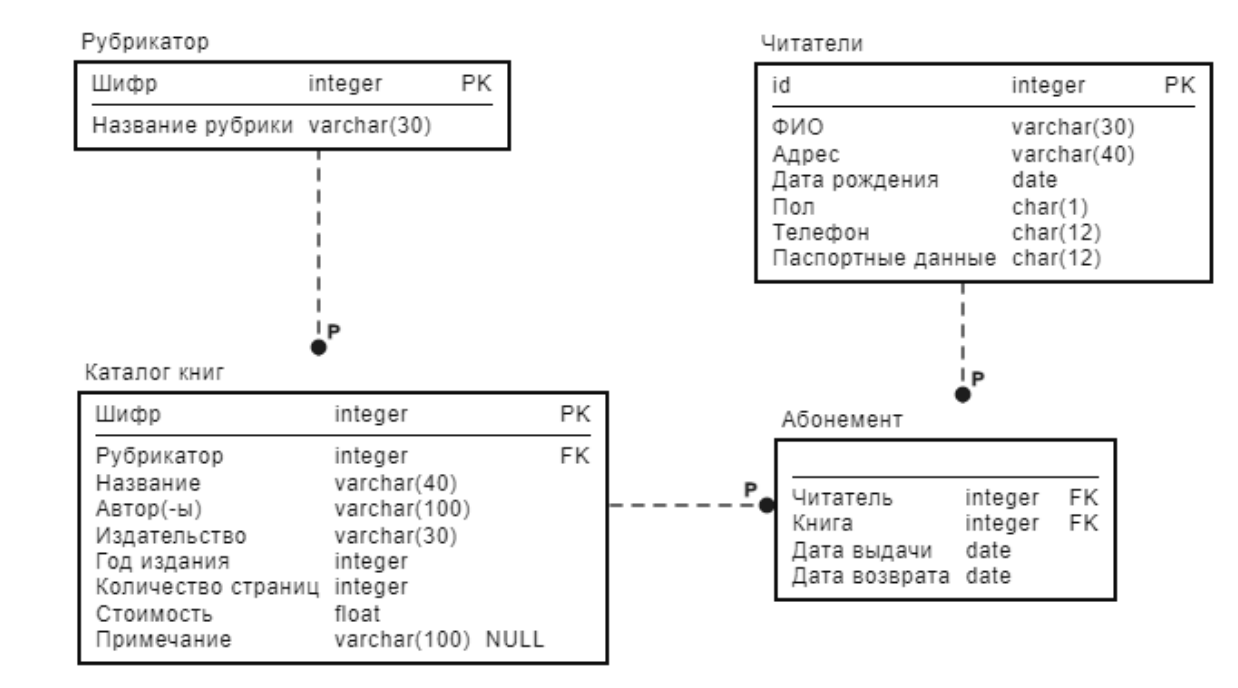
**«Язык SQL. Генераторы. Функции. Триггеры»**

**3.1 Цель работы**

Выработать у обучающихся практические навыки по работе с реляционными базами данных, ознакомить с принципом работы генераторов, функций и триггеров.

**3.2 Индивидуальный вариант**

Вариант лабораторной работы представлен на картинке 3.1.

Рисунок 3.1 – Вариант лабораторной работы

Написать функции:

1. Функция, возвращающая пустую строку или строку «старое издание» для учебников, выпущенных 20 и более лет назад, для справочников, выпущенных 10 и более лет назад и для остальных книг, если они выпущены более 30 лет назад. Параметры: год издания и тип издания (значение поля «примечание»).

2. Функция добавления в специальную таблицу «Отчет» списка книг по указанной рубрике. Таблица должна содержать следующие поля: название рубрики, авторы, название книги, год выпуска, место издания, примечание (старое издание). Параметры – название рубрики и режим: 0 – функция должна перед началом работы очищать таблицу, 1 – новые данные просто добавляются в таблицу. Использовать ранее созданную функцию.

Написать триггеры:

1. Проверка значений всех полей отношения «Каталог книг», для которых могут быть определены домены (в т.ч., год издания не может быть больше текущего года и меньше 1950; количество страниц – от 10 до 2000). Если год издания не указан, устанавливать текущий год.

2. При изменении данных о рубриках – копирование старых значений в специальную таблицу. Сохранять в этой таблице дату изменения и имя пользователя, который произвел изменения

**3.3 Ход выполнения работы**

3.3.1 Были созданы генераторы для всех полей первичных ключей таблиц, как показано на листинге 3.1. В листингах 3.2-3.5 показаны запросы на создание таблиц.

Листинг 3.1 – Создание генераторов для таблиц

create sequence RubricatorCode\_gen;

create sequence BooksCode\_gen;

create sequence ReaderId\_gen;

Листинг 3.2 – Создание таблицы рубрикаторов

create table Rubricators (

code integer default nextval('rubricatorcode\_gen') PRIMARY KEY,

name varchar(30) not null

);

Листинг 3.3 – Создание таблицы книг

create table Books (

code integer default nextval('bookscode\_gen') PRIMARY KEY,

rubricator integer not null,

name varchar(40) not null,

author varchar(100) not null,

publisher varchar(30) not null,

publish\_year integer not null,

pages integer not null,

cost float not null,

note varchar(100),

FOREIGN KEY(rubricator) REFERENCES rubricators(code)

);

Листинг 3.4 – Создание таблицы читателей

create table readers (

id integer default nextval('readerid\_gen') PRIMARY KEY,

fio varchar(30) not null,

address varchar(40) not null,

birthdate date not null,

gender char(1) not null,

phone char(12) not null,

passport char(12) not null

);

Листинг 3.5 – Создание таблицы абонементов

create table season\_tickets (

reader integer not null,

book integer not null,

start\_date date not null,

end\_date date not null,

FOREIGN KEY (reader) REFERENCES readers(id),

FOREIGN KEY (book) REFERENCES books(code)

);

3.3.2 Далее были написаны функции по заданию. В начале была написана функция, возвращающая пустую строку или строку «старое издание» для учебников, выпущенных 20 и более лет назад, для справочников, выпущенных 10 и более лет назад и для остальных книг, если они выпущены более 30 лет назад. Код функции представлен на листинге 3.6

Листинг 3.6 – Код функции для проверки книги на старость

create function is\_old(publish\_date integer, note varchar(100)) returns varchar language plpgsql as

$$

begin

if note = 'учебник' then

if date\_part('year', current\_date) - publish\_date >= 20 then

return 'старое издание';

else return '';

end if;

elseif note = 'справочник' then

if date\_part('year', current\_date) - publish\_date >= 10 then

return 'старое издание';

else return '';

end if;

else

if date\_part('year', current\_date) - publish\_date >= 30 then

return 'старое издание';

else return '';

end if;

end if;

end;

$$

Следующей была написана функция добавления в специальную таблицу «Отчет» списка книг по указанной рубрике. Код этой функции содержится в листинге 3.7.

Листинг 3.7 – Код функции

create function make\_report(rubric\_name varchar(30), mode integer) returns void language plpgsql as

$$

declare rec record;

begin

if mode = 0 then

delete from report;

end if;

for rec in ( select rubricators.name as rubric\_name, author, books.name as book\_name, publish\_year, publisher, note

from rubricators join books on rubricators.code = books.rubricator where rubricators.name = rubric\_name)

loop

insert into report

values (rec.rubric\_name, rec.author, rec.book\_name, rec.publish\_year, rec.publisher, is\_old(rec.publish\_year, rec.note));

end loop;

end;

$$

3.3.3 Также были написаны две триггерные функции, для проверки значений полей отношения «Каталог книг», для которых могут быть определены домены (год издания не может быть больше текущего года и меньше 1950; количество страниц – от 10 до 2000). Если год издания не указан, триггер устанавливает текущий год. И для копирования старых значений таблицы рубрикаторов при изменении в специальную таблицу, в которой также хранятся дата изменения и имя пользователя, который произвел изменения. Код этих триггеров можно увидеть на листингах 3.8 и 3.9.

Листинг 3.8 – Триггерная функция проверки полей таблицы книг

create function check\_book\_insert() returns trigger language plpgsql as

$$

begin

if new.publish\_year is null then

new.publish\_year := date\_part('year', current\_date);

elseif new.publish\_year < 1950 or new.publish\_year > date\_part('year', current\_date) then

raise exception 'Год публикации должен быть между 1950 и настоящим годом';

end if;

if new.pages < 10 or new.pages > 2000 then

raise exception 'Количество страниц должно быть меньше 2000 и больше 10';

end if;

return new;

end;

$$

Листинг 3.9 – Триггерная функции для копирования старых значений таблицы рубрикаторов

create function store\_old\_rubricators() returns trigger language plpgsql as

$$

begin

insert into rubricators\_old values (old.code, old.name, current\_user, current\_date);

return new;

end;

$$

create trigger store\_rubricators\_data

before update on rubricators

for each row execute procedure store\_old\_rubricators();

Далее были написаны триггеры, вызывающие триггерные функции. Их код представлен на листингах 3.10-3.11.

Листинг 3.10 – Триггер для триггерной функции проверки полей таблицы книг

create trigger check\_new\_book

before insert or update on books

for each row execute procedure check\_book\_insert()

Листинг 3.11 – Триггер для триггерной функции копирования старых значений таблицы рубрикаторов

create trigger store\_rubricators\_data

before update on rubricators

for each row execute procedure store\_old\_rubricators();

Также были созданы специальный таблицы для вышеописанных функций. Код создания этих таблиц показан на листингах 3.12-3.13.

Листинг 3.12 – Создание таблицы для отчётов

create table report (

rubric\_name varchar(30),

author varchar(100),

book\_name varchar(40),

publish\_year integer,

publisher varchar(30),

note varchar

);

Листинг 3.13 – Создание таблицы для хранения старых значений таблицы рубрикаторов

create table rubricators\_old(

code integer,

name varchar(30),

username varchar,

date date

);

Далее были написаны запросы на заполнение базы данных значениями.

Эти запросы представлены на листингах 3.14-3.17, а заполненные таблицы представлены на рисунках 3.2-3.5.

Листинг 3.14 – Запрос на заполнение таблицы рубрикаторов

insert into Rubricators (name) values

('Tech Gadgets'),

('Books'),

('Home Appliances'),

('Software'),

('Music Instruments'),

('Sports Equipment'),

('Clothing'),

('Toys'),

('Movies'),

('Games'),

('Furniture'),

('Jewelry'),

('Beauty Products'),

('Pet Supplies'),

('Office Supplies');

Листинг 3.15 – Запрос на заполнение таблицы читателей

insert into readers (fio, address, birthdate, gender, phone, passport) values

('Иван Иванов', 'ул. Пушкина, д. 10', '1990-05-21', 'M', '79991234567', '4509 123456'),

('Мария Петрова', 'ул. Ленина, д. 25', '1985-11-15', 'F', '79997654321', '4509 654321'),

('Алексей Смирнов', 'пр. Гагарина, д. 3', '1992-02-10', 'M', '79993456789', '4510 987654'),

('Ольга Сидорова', 'ул. Чехова, д. 12', '1994-07-22', 'F', '79992345678', '4510 876543'),

('Павел Орлов', 'ул. Маяковского, д. 7', '1988-03-30', 'M', '79994561234', '4511 765432'),

('Наталья Кузнецова', 'ул. Свердлова, д. 14', '1991-08-19', 'F', '79995612345', '4511 654321'),

('Дмитрий Соколов', 'ул. Герцена, д. 9', '1987-04-12', 'M', '79993456780', '4512 543210'),

('Елена Ковалева', 'ул. Гоголя, д. 2', '1995-12-05', 'F', '79997865432', '4512 432109'),

('Максим Лебедев', 'ул. Тургенева, д. 8', '1993-06-11', 'M', '79991233456', '4513 321098'),

('Анна Миронова', 'ул. Шевченко, д. 15', '1996-09-23', 'F', '79994567890', '4513 210987'),

('Владимир Крылов', 'ул. Кутузова, д. 19', '1990-01-17', 'M', '79992345679', '4514 109876'),

('Ирина Котова', 'ул. Жукова, д. 4', '1989-05-25', 'F', '79996789012', '4514 098765'),

('Сергей Новиков', 'ул. Толстого, д. 6', '1986-11-02', 'M', '79997865431', '4515 987654'),

('Татьяна Боброва', 'ул. Ясная, д. 13', '1992-03-21', 'F', '79993456788', '4515 876543'),

('Андрей Быков', 'ул. Лесная, д. 5', '1994-07-14', 'M', '79991234568', '4516 765432');

Листинг 3.16 – Запрос на заполнение таблицы книг

insert into Books (rubricator, name, author, publisher, publish\_year, pages, cost, note) values

(1, 'Учебник математики', 'Иванов И.И.', 'Просвещение', 2020, 300, 500.0, 'учебник'),

(2, 'Справочник по физике', 'Петров П.П.', 'Наука', 2019, 450, 700.0, 'справочник'),

(3, 'Литературный роман', 'Сидоров С.С.', 'Пальмира', 2021, 350, 400.0, null),

(4, 'История искусств', 'Кузнецова А.А.', 'АСТ', 2018, 600, 900.0, null),

(5, 'Биология', 'Лебедев В.В.', 'Эксмо', 2017, 320, 450.0, 'учебник'),

(6, 'Справочник по химии', 'Миронова Т.Т.', 'Мир', 2016, 500, 800.0, 'справочник'),

(7, 'Программирование на Python', 'Крылов К.К.', 'БХВ', 2022, 480, 650.0, null),

(8, 'Кулинарная книга', 'Боброва Н.Н.', 'Рипол Классик', 2020, 250, 300.0, null),

(9, 'Путеводитель по городам', 'Быков А.А.', 'Вече', 2019, 275, 350.0, null),

(10, 'Философия', 'Котова Е.Е.', 'КноРус', 2021, 400, 600.0, null),

(11, 'Основы геометрии', 'Новиков В.В.', 'Феникс', 2018, 300, 500.0, 'учебник'),

(12, 'Руководство по медицине', 'Соколов П.П.', 'ГЭОТАР-Медиа', 2017, 540, 750.0, null),

(13, 'Учебник по программированию', 'Лукина С.С.', 'ДМК Пресс', 2022, 410, 600.0, 'учебник'),

(14, 'Справочник по математике', 'Орлова И.И.', 'Высшая школа', 1952, 475, 700.0, 'справочник'),

(15, 'Руководство по фотографии', 'Тихонов А.А.', 'Эксмо', 1952, 360, 550.0, null);

Листинг 3.17 – Запрос на заполнение таблицы абонементов

insert into season\_tickets (reader, book, start\_date, end\_date) values

(1, 2, '2024-01-10', '2024-02-10'),

(2, 3, '2024-02-15', '2024-03-15'),

(3, 4, '2024-03-20', '2024-04-20'),

(4, 5, '2024-04-25', '2024-05-25'),

(5, 6, '2024-05-30', '2024-06-30'),

(6, 7, '2024-07-05', '2024-08-05'),

(7, 8, '2024-08-10', '2024-09-10'),

(8, 9, '2024-09-15', '2024-10-15'),

(9, 10, '2024-10-20', '2024-11-20'),

(10, 11, '2024-11-25', '2024-12-25'),

(11, 12, '2024-12-30', '2025-01-30'),

(12, 13, '2025-02-05', '2025-03-05'),

(13, 14, '2025-03-10', '2025-04-10'),

(14, 15, '2025-04-15', '2025-05-15'),

(15, 1, '2025-05-20', '2025-06-20');



Рисунок 3.2 – Таблица рубрикаторов

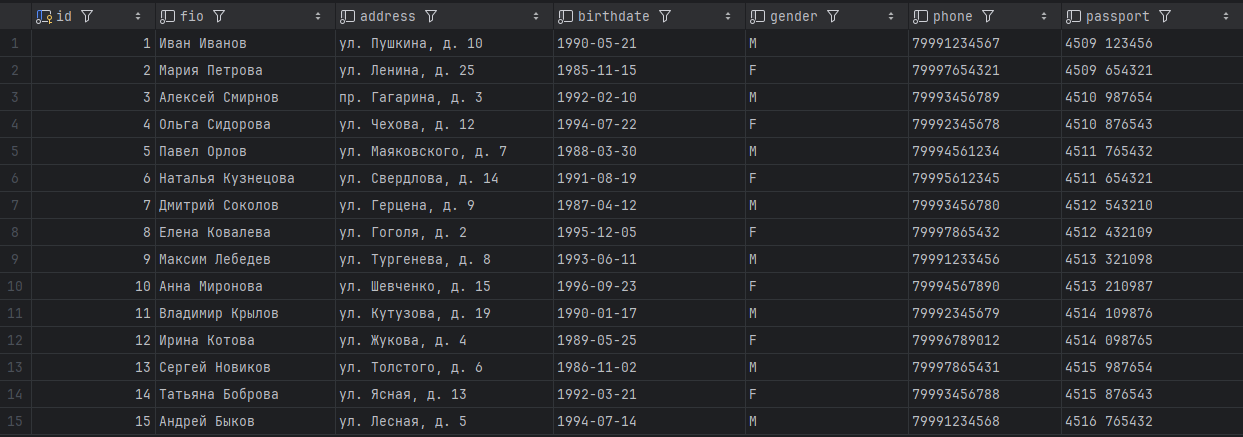


Рисунок 3.3 – Таблица читателей

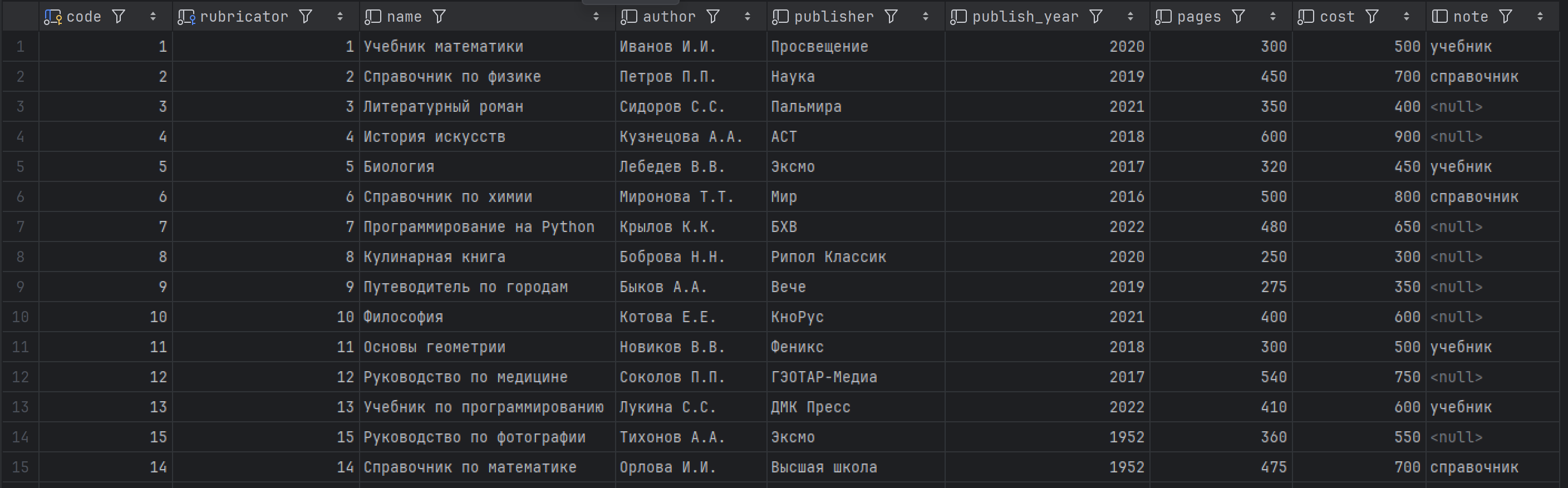


Рисунок 3.4 – Таблица книг

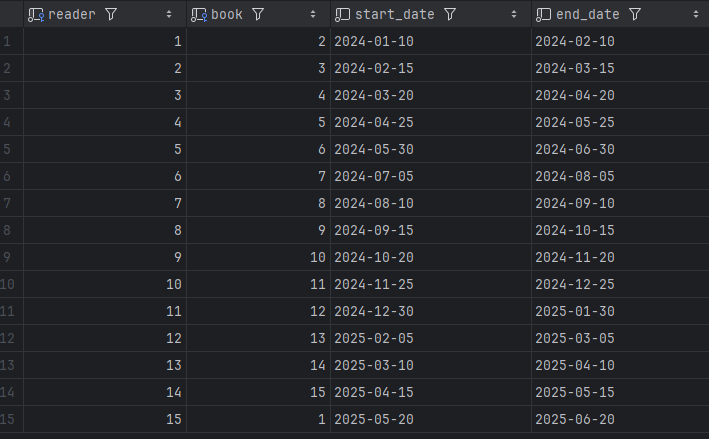


Рисунок 3.5 – Таблица абонементов

Далее была проверена работы функций. Код вызова функций можно увидеть на листинге 3.18, а результаты работы на рисунках 3.6-3.7.

Листинг 3.18 – Вызов функций

select make\_report('Books'::varchar(30), 1);

select make\_report('Pet Supplies'::varchar(30), 0);



Рисунок 3.6 – Первый тест функций



Рисунок 3.7 – Второй тест функций

Далее была проверена работа триггерных функций. Код, который провоцирует вызов этих функций, представлен на листингах 3.19-3.20. Результаты проверки представлены на рисунках 3.8-3.11.

Листинг 3.19 – Тест триггерной функции для проверки добавляемых значений в таблицу книг

insert into Books (rubricator, name, author, publisher, publish\_year, pages, cost, note) values

(15, 'Руководство по танцам', 'Тихонов А.А.', 'Эксмо', null, 360, 550.0, null);

insert into Books (rubricator, name, author, publisher, publish\_year, pages, cost, note) values

(15, 'Руководство по танцам', 'Тихонов А.А.', 'Эксмо', null, 2500, 550.0, null);

insert into Books (rubricator, name, author, publisher, publish\_year, pages, cost, note) values

(15, 'Руководство по танцам', 'Тихонов А.А.', 'Эксмо', 1900, 200, 550.0, null);

Листинг 3.20 – Тест триггерной функции для сохранения старых значений таблицы рубрикаторов

update rubricators set name = 'Electronic Devices' where code = 1;

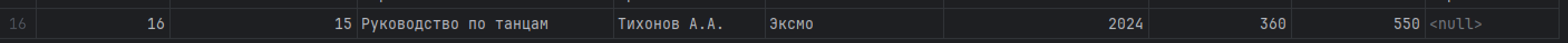


Рисунок 3.8 – Установка настоящей даты, если дата не указана



Рисунок 3.9 – Ошибка при добавлении неправильного количества страниц



Рисунок 3.10 – Ошибка при добавлении неправильного года издания



Рисунок 3.11 – Сохранённые значения при обновлении таблицы рубрикаторов

**Выводы**

В ходе выполнения работы были получены знания о генераторах, способах работы с ними и примеры практического использования. Также были рассмотрены функции в СУБД PostgreSQL, варианты их задания вызова и использования. Были рассмотрены триггеры и триггерные функции, изучены вопросы касающиеся их создания и применения. Далее была создана база данных библиотеки. Для создания полей-идентификаторов использовались генераторы. Было создано несколько функций. Также было создано два триггера для базы данных. В конце был написан отчет по лабораторной работе.