|  |
| --- |
| МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ  ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА» (РУТ (МИИТ))  ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Кафедра «Управление и защита информации» |
|  |
| **КУРСОВОЙ ПРОЕКТ**  **по дисциплине «Основы построения защищенных баз данных»** |
|  |
|  |
|  |
| Выполнил: ст. гр. ТКИ – 441  Кузнецов К. К. |
| Проверил: к.т.н., доц.  Васильева М.А. |
|  |
|  |
|  |
| Москва 2024 |

**Оглавление**

[Описание предметной области 3](#_Toc168393295)

[Проектирование базы данных 4](#_Toc168393296)

[Создание базы данных 8](#_Toc168393297)

[Скрипт 8](#_Toc168393298)

[Результат выполнения скрипта 17](#_Toc168393299)

[Вывод 18](#_Toc168393300)

[Список литературы 19](#_Toc168393301)

Описание предметной области

“Аптека”. База данных (БД) создаётся для информационного обслуживания посетителей аптеки. В аптеку города поступает ассортимент лекарств со склада каждые семь дней. Аптека предлагает услуги по продаже лекарств и их бронированию. Срок бронирования лекарств - три дня. В справочной аптеки можно получить информацию о лекарствах, находящихся в аптеке: название, форма выпуска, срок годности, аннотация, цена, изготовитель.

Готовые запросы:

-- Выдавать данные о лекарствах.

-- Предоставлять покупателям возможность бронирования лекарств, сроком на три дня.

-- Выдавать информацию о поступлении лекарства в данную аптеку, исходя из ассортимента на складе.

-- Выдавать информацию о продажах за неделю (месяц, год) данного лекарства.

-- Выполнять поиск лекарства по названию, форме выпуска, изготовителю.

-- Выдавать список лекарств, применяемых для выбранной болезни (легких недугах).

Проектирование базы данных

Выделим сущности базы данных «Аптека»:

* Manufacturer – производитель лекарств
* Medicines - лекарство
* Pharmacy - аптека
* Stock – наличие лекарства в аптеке
* Sales – продажи лекарства в аптеке
* Reservation – бронирование лекарства в аптеке
* Desease – заболевание (недуг)
* DiseasesMedicines – лекарство от заболевания (недуга)

|  |
| --- |
|  |
| 1. – ER-диаграмма базы данных «Аптека» |

1. - Схема отношения таблицы Manufacturer

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Название поля** | **Тип** | **Примечания** |
| Идентификатор | id | int | Первичный ключ |
| Название | name | varchar(255) |  |
| Страна | country | varchar(255) |  |

1. - Схема отношения таблицы Medicines

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Название поля** | **Тип** | **Примечания** |
| Идентификатор лекарства | id | serial | Первичный ключ |
| Имя лекарства | name | VARCHAR(255) | - |
| Форма выпуска лекарства | form | VARCHAR(255) | - |
| Срок годности лекарства | expiration\_date | DATE | - |
| Аннотация к лекарству | annotation | TEXT | - |
| Цена лекарства | price | DECIMAL(10, 2) | - |
| Идентификатор производителя | manufacturer\_id | INT | Внешний ключ, ссылается на "Manufacturer"(id) |

Таблица 3 - Схема отношения таблицы Pharmacy

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор аптеки | id | serial | Первичный ключ |
| Имя аптеки | name | VARCHAR(255) | - |
| Местоположение аптеки | location | VARCHAR(255) | - |

Таблица 4 - Схема отношения таблицы Stock

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор записи о наличии лекарства | id | serial | Первичный ключ |
| Идентификатор аптеки | pharmacy\_id | INT | Внешний ключ, ссылается на "Pharmacy"(id) |
| Идентификатор лекарства | medicine\_id | INT | Внешний ключ, ссылается на "Medicines"(id) |
| Количество лекарства | quantity | INT | - |

Таблица 5 - Схема отношения таблицы Sales

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор записи о продаже лекарства | id | serial | Первичный ключ |
| Идентификатор аптеки | pharmacy\_id | INT | Внешний ключ, ссылается на "Pharmacy"(id) |
| Идентификатор лекарства | medicine\_id | INT | Внешний ключ, ссылается на "Medicines"(id) |
| Количество проданных единиц лекарства | quantity | INT | - |
| Дата продажи | sale\_date | DATE | - |

Таблица 6 - Схема отношения таблицы Reservation

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор записи о бронировании лекарства | id | serial | Первичный ключ |
| Идентификатор аптеки | pharmacy\_id | INT | Внешний ключ, ссылается на "Pharmacy"(id) |
| Идентификатор лекарства | medicine\_id | INT | Внешний ключ, ссылается на "Medicines"(id) |

Таблица 7 - Схема отношения таблицы Disease

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор болезни | id | serial | Первичный ключ |
| Название болезни | name | VARCHAR(255) | - |

Таблица 8 - Схема отношения таблицы DiseasesMedicines

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечания** |
| Идентификатор записи о связи болезни и лекарства | id | serial | Первичный ключ |
| Идентификатор болезни | disease\_id | INT | Внешний ключ, ссылается на "Disease"(id) |
| Идентификатор лекарства | medicine\_id | INT | Внешний ключ, ссылается на "Medicines"(id) |

Создание базы данных

Скрипт

-- Создание базы данных

CREATE DATABASE drug\_store;

-- Удаление всех таблиц, если они существуют

DROP TABLE IF EXISTS "Reservation";

DROP TABLE IF EXISTS "Sales";

DROP TABLE IF EXISTS "Stock";

DROP TABLE IF EXISTS "DiseaseMedicines";

DROP TABLE IF EXISTS "Disease";

DROP TABLE IF EXISTS "Pharmacy";

DROP TABLE IF EXISTS "Medicines";

DROP TABLE IF EXISTS "Manufacturer";

-- Удаление всех функций, если они существуют

DROP FUNCTION IF EXISTS "getMedicines"();

DROP FUNCTION IF EXISTS "reserveMedicines"(INT, INT, INT);

DROP FUNCTION IF EXISTS "getStock"(INT, INT);

DROP FUNCTION IF EXISTS "getSales"(INT, DATE, DATE);

DROP FUNCTION IF EXISTS "searchMedicine"(VARCHAR(255));

DROP FUNCTION IF EXISTS "getMedicinesForDisease"(INT);

-- Создание таблицы "Manufacturer"

CREATE TABLE "Manufacturer" (

id serial PRIMARY KEY,

name VARCHAR(255),

country VARCHAR(255)

);

-- Создание таблицы "Medicines"

CREATE TABLE "Medicines" (

id serial PRIMARY KEY,

name VARCHAR(255),

form VARCHAR(255),

expiration\_date DATE,

annotation TEXT,

price DECIMAL(10, 2),

manufacturer\_id INT,

FOREIGN KEY (manufacturer\_id) REFERENCES "Manufacturer"(id)

);

-- Создание таблицы "Pharmacy"

CREATE TABLE "Pharmacy" (

id serial PRIMARY KEY,

name VARCHAR(255),

location VARCHAR(255)

);

-- Создание таблицы "Stock"

CREATE TABLE "Stock" (

id serial PRIMARY KEY,

pharmacy\_id INT,

medicine\_id INT,

quantity INT,

FOREIGN KEY (pharmacy\_id) REFERENCES "Pharmacy"(id),

FOREIGN KEY (medicine\_id) REFERENCES "Medicines"(id)

);

-- Создание таблицы "Sales"

CREATE TABLE "Sales" (

id serial PRIMARY KEY,

pharmacy\_id INT,

medicine\_id INT,

quantity INT,

sale\_date DATE,

FOREIGN KEY (pharmacy\_id) REFERENCES "Pharmacy"(id),

FOREIGN KEY (medicine\_id) REFERENCES "Medicines"(id)

);

-- Создание таблицы "Reservation"

CREATE TABLE "Reservation" (

id serial PRIMARY KEY,

pharmacy\_id INT,

medicine\_id INT,

quantity INT,

reservation\_date DATE,

FOREIGN KEY (pharmacy\_id) REFERENCES "Pharmacy"(id),

FOREIGN KEY (medicine\_id) REFERENCES "Medicines"(id)

);

-- Создание таблицы "Disease"

CREATE TABLE "Disease" (

id serial PRIMARY KEY,

name VARCHAR(255)

);

-- Создание таблицы "DiseaseMedicines"

CREATE TABLE "DiseaseMedicines" (

id serial PRIMARY KEY,

disease\_id INT,

medicine\_id INT,

FOREIGN KEY (disease\_id) REFERENCES "Disease"(id),

FOREIGN KEY (medicine\_id) REFERENCES "Medicines"(id)

);

-- Функция для получения данных о лекарствах

CREATE FUNCTION "getMedicines"()

RETURNS TABLE (

id INT,

name VARCHAR(255),

form VARCHAR(255),

expiration\_date DATE,

annotation TEXT,

price DECIMAL(10, 2),

manufacturer\_id INT

)

AS $$

BEGIN

RETURN QUERY SELECT \* FROM "Medicines";

END;

$$ LANGUAGE plpgsql;

-- Функция для бронирования лекарств

CREATE FUNCTION "reserveMedicines"(pharmacy\_id INT, medicine\_id INT, quantity INT)

RETURNS VOID

AS $$

BEGIN

INSERT INTO "Reservation" (pharmacy\_id, medicine\_id, quantity, reservation\_date)

VALUES (pharmacy\_id, medicine\_id, quantity, CURRENT\_DATE + INTERVAL '3 days');

END;

$$ LANGUAGE plpgsql;

-- Функция для получения информации о поступлении лекарства в аптеку

CREATE FUNCTION "getStock"(p\_id INT, m\_id INT)

RETURNS TABLE (

id INT,

pharmacy\_id INT,

medicine\_id INT,

quantity INT

)

AS $$

BEGIN

RETURN QUERY SELECT \* FROM "Stock"

WHERE "Stock".pharmacy\_id = p\_id AND "Stock".medicine\_id = m\_id;

END;

$$ LANGUAGE plpgsql;

-- Функция для получения информации о продажах лекарства за определенный период

CREATE FUNCTION "getSales"(m\_id INT, start\_date DATE, end\_date DATE)

RETURNS TABLE (

id INT,

pharmacy\_id INT,

medicine\_id INT,

quantity INT,

sale\_date DATE

)

AS $$

BEGIN

RETURN QUERY SELECT \* FROM "Sales"

WHERE "Sales".medicine\_id = m\_id AND "Sales".sale\_date BETWEEN start\_date AND end\_date;

END;

$$ LANGUAGE plpgsql;

-- Функция для поиска лекарства по названию, форме выпуска или изготовителю

CREATE FUNCTION "searchMedicine"(keyword VARCHAR(255))

RETURNS TABLE (

id INT,

name VARCHAR(255),

form VARCHAR(255),

expiration\_date DATE,

annotation TEXT,

price DECIMAL(10, 2),

manufacturer\_id INT

)

AS $$

BEGIN

RETURN QUERY SELECT \* FROM "Medicines"

WHERE "Medicines".name ILIKE '%' || keyword || '%' OR "Medicines".form ILIKE '%' || keyword || '%' OR CAST("Medicines".manufacturer\_id AS TEXT) ILIKE '%' || keyword || '%';

END;

$$ LANGUAGE plpgsql;

-- Функция для получения списка лекарств для выбранной болезни

CREATE FUNCTION "getMedicinesForDisease"(d\_id INT)

RETURNS TABLE (

id INT,

name VARCHAR(255),

form VARCHAR(255),

expiration\_date DATE,

annotation TEXT,

price DECIMAL(10, 2),

manufacturer\_id INT

)

AS $$

BEGIN

RETURN QUERY SELECT m.id, m.name, m.form, m.expiration\_date, m.annotation, m.price, m.manufacturer\_id

FROM "Medicines" m

INNER JOIN "DiseaseMedicines" dm ON m.id = dm.medicine\_id

WHERE dm.disease\_id = d\_id;

END;

$$ LANGUAGE plpgsql;

-- Вставка данных в таблицу "Manufacturer"

INSERT INTO "Manufacturer" (id, name, country)

VALUES (1, 'Bayer', 'Germany'),

(2, 'Johnson & Johnson', 'USA'),

(3, 'Pfizer', 'USA');

-- Вставка данных в таблицу "Medicines"

INSERT INTO "Medicines" (id, name, form, expiration\_date, annotation, price, manufacturer\_id)

VALUES (1, 'Aspirin', 'Tablet', '2023-01-01', 'Aspirin is a medication used to treat pain, fever, or inflammation.', 5.99, 1),

(2, 'Paracetamol', 'Tablet', '2023-01-01', 'Paracetamol is a medication used to treat pain and fever.', 3.99, 2),

(3, 'Ibuprofen', 'Tablet', '2023-01-01', 'Ibuprofen is a medication used to treat pain, fever, or inflammation.', 4.99, 3);

-- Вставка данных в таблицу "Pharmacy"

INSERT INTO "Pharmacy" (id, name, location)

VALUES (1, 'Pharmacy 1', '123 Main Street'),

(2, 'Pharmacy 2', '456 Elm Street');

-- Вставка данных в таблицу "Stock"

INSERT INTO "Stock" (id, pharmacy\_id, medicine\_id, quantity)

VALUES (1, 1, 1, 100),

(2, 1, 2, 200),

(3, 2, 3, 150);

-- Вставка данных в таблицу "Sales"

INSERT INTO "Sales" (id, pharmacy\_id, medicine\_id, quantity, sale\_date)

VALUES (1, 1, 1, 10, '2023-01-01'),

(2, 1, 2, 20, '2023-01-02'),

(3, 2, 3, 15, '2023-01-03');

-- Вставка данных в таблицу "Reservation"

INSERT INTO "Reservation" (pharmacy\_id, medicine\_id, quantity, reservation\_date)

VALUES (1, 1, 5, '2023-01-01'),

(1, 2, 10, '2023-01-02');

-- Вставка данных в таблицу "Disease"

INSERT INTO "Disease" (id, name)

VALUES (1, 'Headache'),

(2, 'Fever');

-- Вставка данных в таблицу "DiseaseMedicines"

INSERT INTO "DiseaseMedicines" (id, disease\_id, medicine\_id)

VALUES (1, 1, 1),

(2, 2, 2);

-- Вызов функции для получения данных о лекарствах

SELECT \* FROM "getMedicines"();

-- Вызов функции для бронирования лекарств

SELECT "reserveMedicines"(1, 1, 5);

-- Вызов функции для получения информации о поступлении лекарства в аптеку

SELECT \* FROM "getStock"(1, 1);

-- Вызов функции для получения информации о продажах лекарства за определенный период

SELECT \* FROM "getSales"(1, '2023-01-01', '2023-01-02');

-- Вызов функции для поиска лекарства по названию, форме выпуска или изготовителю

SELECT \* FROM "searchMedicine"('Aspirin');

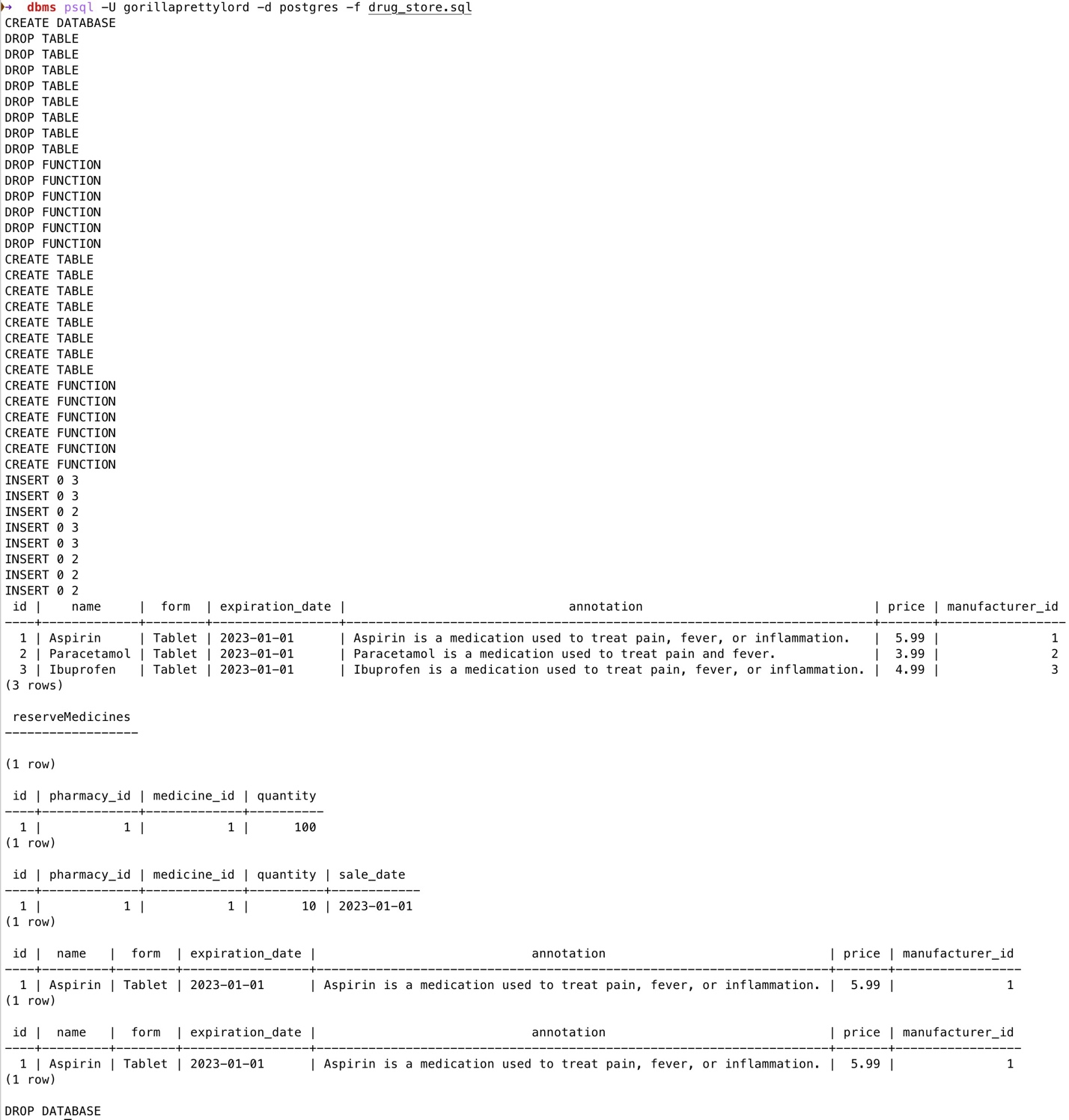
-- Вызов функции для получения списка лекарств для выбранной болезни

SELECT \* FROM "getMedicinesForDisease"(1);

-- Удаление базы данных

DROP DATABASE drug\_store;

Результат выполнения скрипта



1. – результат выполнения drug\_store.sql

Вывод

В ходе данного курсового проекта была создана база данных для информационного обслуживания посетителей аптеки. Были изучены операторы языка определения данных DDL, операторы языка модификации данных DML, возможности диалекта PostrgreSQL для создания функций с готовыми запросами – что позволяет пользоваться базой данных без необходимости создания над ней ORM системы. Также был приобретен навык в построении ER диаграмм и в составлении схем отношений сущностей в виде таблиц.

Список литературы

1. Балакина Е. П., Васильева М. А., Филипченко К. М. Информационное обеспечение систем управления. Методические указания к курсовому проектированию. Учебно-методическое пособие. Издание второе, исправленное и дополненное, 2023. 102 –с.
2. Васильева М. А., Степанов Д. Е. Операторы языка определения данных. Учебно– методическое пособие. – М.: РУТ (МИИТ). 2024. – 33с
3. Васильева М. А., Балакина Е. П., Васильева М. В. Соединение данных из множества таблиц. Рекомендации по выполнению работы и перечень типовых заданий. Учебно–методическое пособие. Издание второе, исправленное. – М.: РУТ (МИИТ).2024. – 72с.
4. Васильева М. А, Филипченко К. М., Пугачёв В. А. Фильтрация набора данных. Рекомендации по выполнению работы и перечень типовых заданий: Учебно-методическое пособие. Издание третье, исправленное и дополненное. Москва–2024, 133 с.
5. Васильева М. А., Кузнецов К. К. Операторы языка управления данными. Учебно– методическое пособие. – М.: РУТ (МИИТ).2024. – 36с.