Министерство науки и высшего образования Российской Федерации ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № _3_

«Создание таблиц базы данных PostgreSQL. Заполнение таблиц рабочими данными»

по дисциплине «Проектирование и реализация баз данных»

Обучающийся Смирновой Карины Факультет прикладной информатики Группа К3241 Направление подготовки 09.03.03 Прикладная информатика Образовательная программа Мобильные и сетевые технологии 2023 Преподаватель Говорова Марина Михайловна

Цель работы: овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4.

Практическое задание:

- 1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).
- 2. Создать схему в составе базы данных.
- 3. Создать таблицы базы данных.
- 4. Установить ограничения на данные: Primary Key, Unique, Check, Foreign Key.
- 5. Заполнить таблицы БД рабочими данными.
- 6. Создать резервную копию БД.

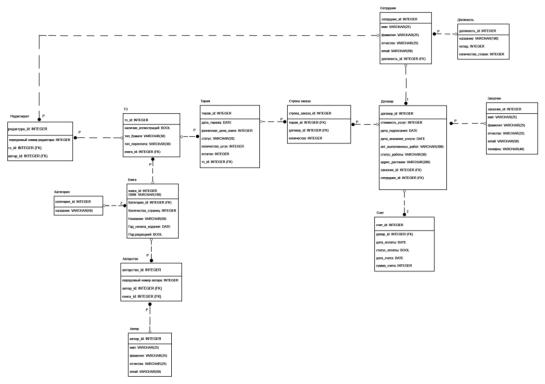
Указание:

Создать две резервные копии:

- с расширением CUSTOM для восстановления БД;
- с расширением PLAIN для листинга (в отчете);
- при создании резервных копий БД настроить параметры Dump options для Type of objects и Queries .
- 7. Восстановить БД.

Выполнение:

Моя схема базы данных IDEF1X из 2 Лабораторной работы:



1) Создание базы данных с названием «LB3»

```
Create - Database
                                                                                X
General Definition Security Parameters Advanced SQL
1 - CREATE DATABASE "LB3"
2
        WITH
3
        OWNER = postgres
4
        ENCODING = 'UTF8'
5
        CONNECTION LIMIT = -1
6
        IS_TEMPLATE = False;
7
8 V COMMENT ON DATABASE "LB3"
        IS 'База данных для лабораторной работы 3';
```

2) Создание схемы «mainS»

```
© Create - Schema

General Security Default privileges SQL

CREATE SCHEMA "mainS"
AUTHORIZATION postgres;

COMMENT ON SCHEMA "mainS"
IS 'OCHOBHAR CXEMA';
```

3) Создание таблиц

Создание таблицы «Автор»:

```
CREATE TABLE "mainS".author (
   author_id SERIAL PRIMARY KEY,
   first_name VARCHAR(25) NOT NULL CHECK (first_name ~ '^[A-Za-zA-Яa-яëË\\-\\s]+$'),
   surname VARCHAR(25) NOT NULL CHECK (surname ~ '^[A-Za-zA-Яa-яëË\s-]+$'),
   last_name VARCHAR(25) CHECK (last_name ~ '^[A-Za-zA-Яa-яëË\\-\\s]+$' OR last_name IS NULL),
   email VARCHAR(50) NOT NULL UNIQUE CHECK (email ~ '^[A-Za-z]+@.+\..+$')
);
```

Создание таблицы «Категория»:

```
E Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

1 

CREATE TABLE "mainS".category

(
    categoty_id serial NOT NULL,
    category_title character varying(50) NOT NULL,

PRIMARY KEY (categoty_id),

CONSTRAINT category_title CHECK (category_title ~ '^[A-Za-ZA-Яа-Я-;№:(),\s]+$') NOT VALID

);

8

ALTER TABLE IF EXISTS "mainS".category

OWNER to postgres;
```

Создание таблицы «Книга»:

```
CREATE TABLE "mainS".book

(

book_id serial NOT NULL PRIMARY KEY,

ISBN character varying(100) NOT NULL UNIQUE,

categoty_id integer REFERENCES "mainS".category(category_id) ON DELETE SET NULL,

page_count integer CHECK (page_count > 0),

book_title character varying(50) NOT NULL CHECK (book_title ~ E'^[A-Za-zA-Яa-ЯëЁ0-9\\s!@#$%^&*(

publish_start_date date CHECK (publish_start_date > '1445-01-01'),

under_publish boolean NOT NULL

);
```

Создание таблицы «Авторство»:

```
CREATE TABLE authorship (
    authorship_id SERIAL PRIMARY KEY,
    author_id INTEGER NOT NULL REFERENCES author(author_id) ON DELETE CASCADE,
    book_id INTEGER NOT NULL REFERENCES book(book_id) ON DELETE CASCADE,
    author_order INTEGER NOT NULL CHECK (author_order > 0)
);
Создание таблицы «Должность»:
 CREATE TABLE "mainS".post (
     post_id SERIAL PRIMARY KEY,
     post_name VARCHAR(100) NOT NULL CHECK (post_name ~ '^[A-Za-zA-Яa-я-:№(),\s]+$'),
     salary INTEGER NOT NULL CHECK (salary >= 0),
     num_positions INTEGER NOT NULL CHECK (num_positions > 0)
 );
Создание таблицы «Сотрудник»:
 CREATE TABLE "mainS".employee (
    employee_id SERIAL PRIMARY KEY,
    post_id INTEGER NOT NULL REFERENCES "mainS".post(post_id) ON DELETE CASCADE,
    first_name VARCHAR(25) NOT NULL CHECK (first_name ~ '^[A-Za-zA-Яa-Я\-]+$'),
    surname VARCHAR(25) NOT NULL CHECK (surname ~ '^[A-Za-zA-9a-9\-]+$'),
    last_name VARCHAR(25) CHECK (last_name ~ '^[A-Za-zA-Яa-я\-]+$' OR last_name IS NULL),
    email VARCHAR(50) NOT NULL UNIQUE CHECK (email ~ '^[^@]+@[^@]+\.[^@]+$')
 );
Создание таблицы «ТЗ»:
 CREATE TABLE "mainS".tz (
     tz_id SERIAL PRIMARY KEY,
     book_id INTEGER NOT NULL REFERENCES "mainS".book(book_id) ON DELETE CASCADE,
     has_illustrations BOOLEAN NOT NULL,
     paper_type VARCHAR(30) NOT NULL CHECK (paper_type IN (
         'мелованная бумага',
         'офсетная бумага',
         'дизайнерская бумага',
         'газетная бумага'
     )),
     binding_type VARCHAR(30) NOT NULL CHECK (binding_type IN (
         'переплет термоклеевой',
         'мягкий переплет скоба',
         'мягкий переплет пружина',
         'твердый переплет',
         'твердый переплет супер-обложка'
     ))
 );
Создание таблицы «Редактирует»:
 CREATE TABLE "mainS".edit (
    edit_id SERIAL PRIMARY KEY,
    tz_id INTEGER NOT NULL REFERENCES "mainS".tz(tz_id) ON DELETE CASCADE,
    author_id INTEGER NOT NULL REFERENCES "mainS".author(author_id) ON DELETE CASCADE,
    author_order INTEGER NOT NULL CHECK (author_order > 0)
 );
```

```
CREATE TABLE "mainS".edition (
   edition_id SERIAL PRIMARY KEY,
   tz_id INTEGER NOT NULL REFERENCES "mainS".tz(tz_id) ON DELETE CASCADE,
   edition_date DATE NOT NULL CHECK (edition_date >= CURRENT_DATE) NOT VALID,
   retail_price INTEGER NOT NULL CHECK (retail_price > 0),
   status VARCHAR(20) NOT NULL CHECK (status IN ('подготовка', 'в печати', 'в продаже', 'распродан
   quantity INTEGER NOT NULL CHECK (quantity > 0),
   remaining INTEGER NOT NULL CHECK (remaining >= 0 AND remaining <= quantity) DEFAULT 0
);
```

Создание таблицы «Заказчик»:

```
CREATE TABLE "mainS".customer (
    customer_id SERIAL PRIMARY KEY,
    first_name VARCHAR(25) NOT NULL CHECK (first_name ~ '^[A-Za-zA-Яa-Я\-]+$'),
    surname VARCHAR(25) NOT NULL CHECK (surname ~ '^[A-Za-zA-Яa-Я\-]+$'),
    last_name VARCHAR(25) CHECK (last_name ~ '^[A-Za-zA-Яa-Я\-]+$' OR last_name IS NULL),
    email VARCHAR(50) NOT NULL UNIQUE CHECK (email ~ '^[^@]+@[^@]+\.[^@]+$'),
    phone VARCHAR(11) CHECK (phone ~ '^\d{11}$')
);
```

Создание таблицы «Договор»:

```
CREATE TABLE "mainS".contract (
    contract_id SERIAL PRIMARY KEY,
    customer_id INTEGER NOT NULL REFERENCES "mainS".customer(customer_id),
    employee_id INTEGER NOT NULL REFERENCES "mainS".employee(employee_id),
    service_cost INTEGER NOT NULL CHECK (service_cost >= 0),
    signing_date DATE NOT NULL CHECK (signing_date <= service_date),
    service_date DATE CHECK (service_date >= signing_date OR service_date IS NULL),
    work_act VARCHAR(300),
    work_status VARCHAR(30) NOT NULL CHECK (work_status IN ('в очереди', 'выполняется', 'завершен', 'delivery_address VARCHAR(200) NOT NULL CHECK (delivery_address ~ '^[A-Za-zA-Яa-я-:№(),\s]+$')
);
```

Создание таблицы «Счет»:

```
CREATE TABLE "mainS".invoice (
    invoice_id SERIAL PRIMARY KEY,
    contract_id INTEGER NOT NULL REFERENCES "mainS".contract(contract_id) ON DELETE CASCADE,
    payment_date DATE CHECK (payment_date >= invoice_date),
    payment_status BOOLEAN NOT NULL,
    invoice_date DATE NOT NULL CHECK (invoice_date <= payment_date),
    invoice_amount INTEGER NOT NULL CHECK (invoice_amount >= 0)
);
```

Создание таблицы «Строка заказа»:

```
CREATE TABLE "mainS".order_line (
    order_line_id SERIAL PRIMARY KEY,
    edition_id INTEGER NOT NULL REFERENCES "mainS".edition(edition_id) ON DELETE CASCADE,
    contract_id INTEGER NOT NULL REFERENCES "mainS".contract(contract_id) ON DELETE CASCADE,
    quantity INTEGER NOT NULL CHECK (quantity > 0)
);
```

4) Заполнить таблицы БД рабочими данными.

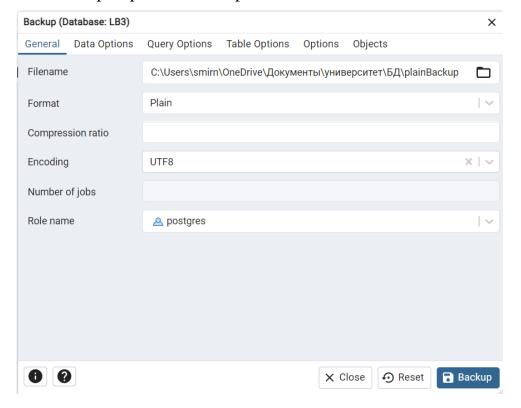
```
-- Данные в таблицу post
INSERT INTO "mainS".post (post_name, salary, num_positions) VALUES
('Редактор', 70000, 5),
('Менеджер по продажам', 80000, 3),
('Технический писатель', 75000, 4),
('Программист', 90000, 2),
('Графический дизайнер', 65000, 3);
-- Данные в таблицу employee
INSERT INTO "mainS".employee (post_id, first_name, surname, last_name, email) VALUES
(1, 'Сергей', 'Морозов', 'Владимирович', 'sergey.morozov@mail.com'),
(2, 'Ирина', 'Новикова', 'Анатольевна', 'irina.novikova@mail.com'),
(3, 'Олег', 'Семенов', 'Григорьевич', 'oleg.semenov@mail.com'),
(4, 'Екатерина', 'Орлова', 'Дмитриевна', 'ekaterina.orlova@mail.com'),
(5, 'Роман', 'Макаров', 'Александрович', 'roman.makarov@mail.com'),
(1, 'Анастасия', 'Зуева', 'Олеговна', 'anastasia.zueva@mail.com'),
(2, 'Геннадий', 'Титов', 'Иванович', 'gennady.titov@mail.com'),
(3, 'Людмила', 'Киселева', 'Сергеевна', 'lyudmila.kiseleva@mail.com'),
(4, 'Владимир', 'Фролов', 'Васильевич', 'vladimir.frolov@mail.com'),
(5, 'Светлана', 'Соловьева', 'Петровна', 'svetlana.solovieva@mail.com');
-- Данные в таблицу author
INSERT INTO "mainS".author (first_name, surname, last_name, email) VALUES
('Дональд', 'Кнут', NULL, 'donaldknuth@mail.com'),
('Эндрю', 'Таненбаум', NULL, 'andrewtanenbaum@mail.com'),
('Бьерн', 'Страуструп', NULL, 'bjarnestroustrup@mail.com'),
('Мартин', 'Фаулер', NULL, 'martinfowler@mail.com'),
('Роберт', 'Мартин', NULL, 'robertmartin@mail.com'),
('Гвидо', 'ван Россум', NULL, 'guidorossum@mail.com'),
('Кен', 'Томпсон', NULL, 'kenthompson@mail.com'),
('Деннис', 'Ритчи', NULL, 'dennisritchie@mail.com'),
('Линус', 'Торвальдс', NULL, 'linustorvalds@mail.com'),
('Ричард', 'Столлман', NULL, 'richardstallman@mail.com');
-- Данные в таблицу category
INSERT INTO "mainS".category (category_name) VALUES
('Учебник'),
('Учебное пособие'),
('Монография'),
('Справочник'),
('Практическое руководство');
-- Данные в таблицу book
INSERT INTO "mainS".book (ISBN, categoty_id, page_count, book_title, publish_start_date, under_publ
('978-5-907144-23-1', 1, 800, 'Искусство программирования', '1968-01-01', TRUE),
('978-5-4461-1155-8', 1, 1050, 'Операционные системы', '2008-03-15', TRUE),
('978-5-6045724-6-7', 2, 725, 'Язык программирования С++', '1997-05-14', TRUE),
('978-5-6041394-0-0', 2, 350, 'Шаблоны корпоративных приложений', '2002-06-15', TRUE),
('978-5-4461-0960-9', 3, 450, 'Чистый код', '2008-08-01', TRUE),
('978-0-596-15806-4', 4, 250, 'Python для всех', '2016-07-11', TRUE),
('978-0-13-110362-7', 1, 300, 'Язык программирования С', '1988-02-22', TRUE),
('978-0-13-142901-7', 3, 400, 'Программирование на языке С', '2000-04-10', TRUE),
('978-0-13-110362-8', 4, 600, 'Архитектура UNIX', '1983-09-01', TRUE),
('978-0-13-409341-3', 5, 550, 'Свобода программирования', '2015-01-12', TRUE);
-- Данные в таблицу authorship
INSERT INTO "mainS".authorship (author_id, book_id, author_order) VALUES
(1, 1, 1),
(2, 2, 1),
(3, 3, 1),
(4, 4, 1),
(5, 5, 1),
(6, 6, 1),
(7, 7, 1),
(8, 8, 1),
(9, 9, 1),
(10, 10, 1);
```

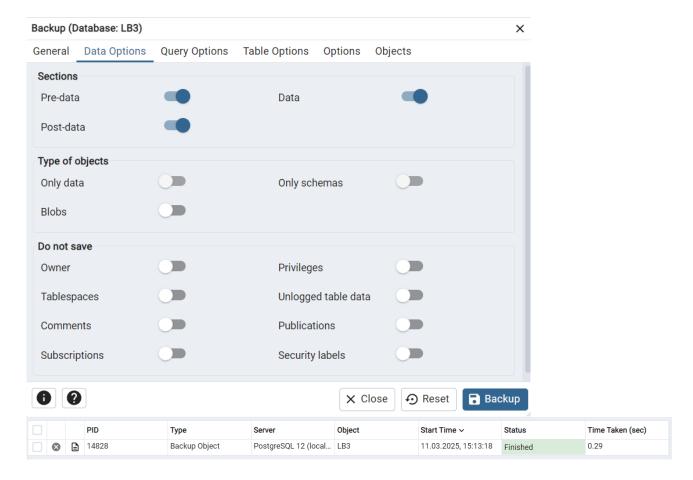
```
-- Данные в таблицу tz
 INSERT INTO "mainS".tz (book_id, has_illustrations, paper_type, binding_type) VALUES
 (1, TRUE, 'мелованная бумага', 'твердый переплет'),
 (2, FALSE, 'офсетная бумага', 'мягкий переплет скоба'),
 (3, TRUE, 'дизайнерская бумага', 'твердый переплет супер-обложка'),
 (4, FALSE, 'газетная бумага', 'переплет термоклеевой'),
 (5, TRUE, 'мелованная бумага', 'мягкий переплет пружина'),
 (6, FALSE, 'дизайнерская бумага', 'мягкий переплет пружина'),
 (7, FALSE, 'газетная бумага', 'переплет термоклеевой'),
 (8, TRUE, 'мелованная бумага', 'твердый переплет'),
 (9, TRUE, 'мелованная бумага', 'твердый переплет супер-обложка'),
 (10, FALSE, 'газетная бумага', 'переплет термоклеевой');
 -- Данные в таблицу edit
 INSERT INTO "mainS".edit (tz_id, author_id, author_order) VALUES
 (1, 1, 1),
 (2, 2, 1),
 (3, 3, 1),
 (4, 4, 1),
 (5, 5, 1),
 (6, 6, 1),
 (7, 6, 1),
 (8, 8, 1),
 (9, 9, 1),
 (10, 10, 1);
 -- Данные в таблицу edition
 INSERT INTO "mainS".edition (tz_id, edition_date, retail_price, status, quantity, remaining)
 VALUES
     (1, CURRENT_DATE, 500, 'подготовка', 1000, 100),
     (2, CURRENT_DATE + 1, 700, 'в печати', 500, 300),
     (3, CURRENT_DATE + 2, 600, 'в продаже', 300, 50),
     (4, CURRENT_DATE + 3, 800, 'распродан', 400, 0),
     (5, CURRENT_DATE + 4, 900, 'подготовка', 600, 200),
     (6, CURRENT_DATE, 500, 'подготовка', 1000, 100),
     (7, CURRENT_DATE + 1, 700, 'в печати', 500, 300),
     (8, CURRENT_DATE + 2, 600, 'в продаже', 300, 50),
     (9, CURRENT_DATE + 3, 800, 'распродан', 400, 0),
     (10, CURRENT_DATE + 4, 900, 'подготовка', 600, 200);
 -- Данные в таблицу customer
· INSERT INTO "mainS".customer (first_name, surname, last_name, email, phone) VALUES
 ('000', 'ИнфоТех', NULL, 'contact@infotech.com', '79161234567'),
 ('AO', 'АйТи Солюшнс', NULL, 'info@itsolutions.com', '79261112233'),
 ('ЗАО', 'Разработчик', NULL, 'contact@developer.ru', '79374445566'),
 ('000', 'Издательство Код', \texttt{NULL}, 'contact@codepub.com', '79152223344'),
 ('AO', 'Программист', NULL, 'info@programmer.com', '79213334455');
```

```
-- Данные в таблицу contract
INSERT INTO "mainS".contract (customer_id, employee_id, service_cost, signing_date, service_date, w
(1, 21, 100000, '2023-06-01', '2023-09-15', 'act_001.pdf', 'завершен', 'Москва, ул. Ленина, 10'), (2, 22, 85000, '2023-07-10', '2023-10-20', 'act_002.pdf', 'выполняется', 'Санкт-Петербург, Невский
(3, 23, 120000, '2023-05-05', '2023-08-25', 'act_003.pdf', 'в очереди', 'Казань, ул. Баумана, 17'), (4, 24, 35000, '2023-05-11', '2023-08-04', 'act_004.pdf', 'выполняется', 'Москва, ул.Космонавтов, 1
(5, 25, 10000, '2023-05-12', '2023-08-20', 'act_005.pdf', 'выполняется', 'Киров, ул.Лермонтова, 12'
-- Данные в таблицу invoice
INSERT INTO "mainS".invoice (contract_id, payment_date, payment_status, invoice_date, invoice_amoun
(1, '2023-06-10', TRUE, '2023-06-01', 50000),
(1, '2023-09-20', TRUE, '2023-09-15', 50000),
(2, '2023-07-15', TRUE, '2023-07-10', 40000),
(2, '2023-10-30', FALSE, '2023-10-20', 45000),
(3, '2023-07-10', TRUE, '2023-07-01', 70000), (4, '2023-06-15', FALSE, '2023-06-03', 90000),
(5, '2023-08-13', TRUE, '2023-08-02', 20000);
-- Данные в таблицу order_line
INSERT INTO "mainS".order_line (edition_id, contract_id, quantity) VALUES
(6, 10, 50),
(2, 11, 30),
(3, 12, 40),
(4, 13, 20),
(5, 14, 35);
```

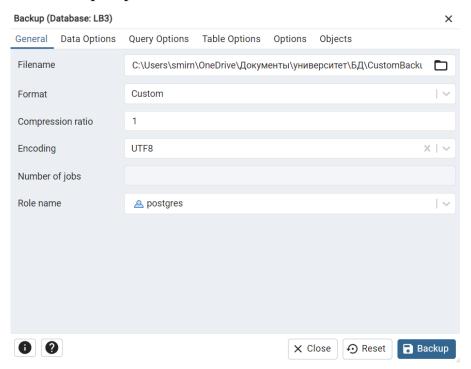
5) Создать резервную копию БД.

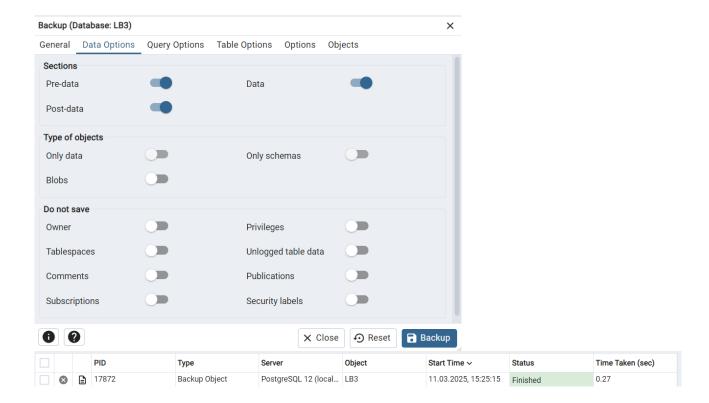
Создание резервной копии plain:



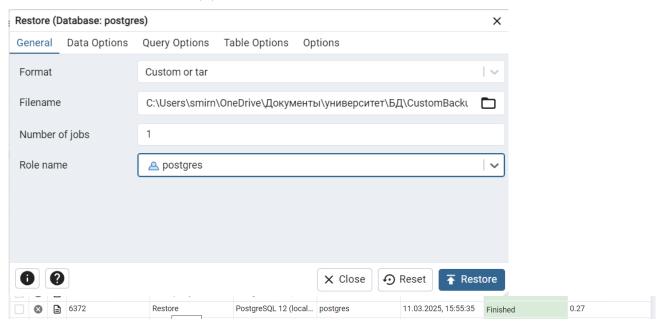


Создание резервной копии custom:





6) Восстановление БД



Выводы:

В данной лабораторной работе я научилась создавать таблицы для базы данных и заполнять ее значениями с помощью pgAdmin. Для создания таблиц я изучила, как создаются ограничения на значения. Также я научилась создавать различные виды резервных копий базы дынных, таких как Custom, Plain, и восстанавливать базу данных по этим копиям.