Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №3 по дисциплине: «Создание таблиц базы данных POSTGRESQL. Заполнение таблиц рабочими данными»

Выполнил:
Студент 2 курса ИКТ
группы К3241
Павел Золотов

Проверил:
Говорова Марина Михайловна

Санкт-Петербург
2021

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

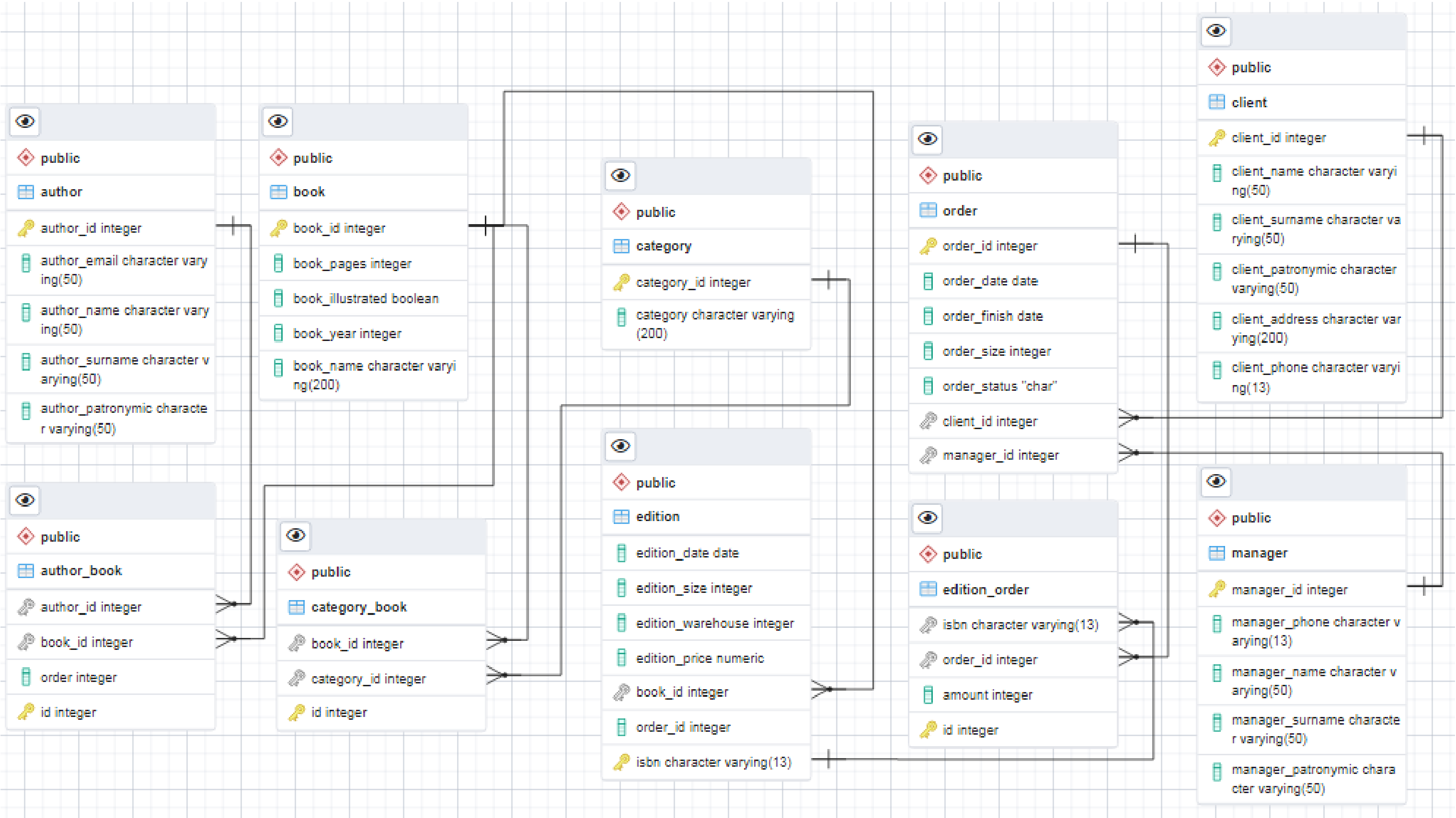
Практическое задание:

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.

Выполнение:

Название создаваемой БД – homework3

ER диаграмма



Запрос для создания таблиц author

```
CREATE TABLE public.author
(
    author_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    author_email character varying(50) COLLATE pg_catalog."default",
    author_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
    author_surname character varying(50) COLLATE pg_catalog."default" NOT NULL,
    author_patronymic character varying(50) COLLATE pg_catalog."default",
    CONSTRAINT "Author_pkey" PRIMARY KEY (author_id),
    CONSTRAINT author_email_check CHECK (author_email::text ~~ '%@%'::text) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.author
    OWNER to postgres;
```

Запрос для создания таблиц author_book

```
CREATE TABLE public.author_book
(
    author_id integer NOT NULL,
    book_id integer NOT NULL,
    "order" integer,
    CONSTRAINT author_id FOREIGN KEY (author_id)
        REFERENCES public.author (author_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT book_id FOREIGN KEY (book_id)
        REFERENCES public.book (book_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE public.author_book
    OWNER to postgres;
```

Запрос для создания таблиц book

```
CREATE TABLE public.book
(
    book_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    book_pages integer NOT NULL,
    book_illustrated boolean,
    book_year integer,
    book_name character varying(200) COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT "Book_pkey" PRIMARY KEY (book_id),
    CONSTRAINT book_pages_check CHECK (book_pages < 10000 AND
book_pages > 0) NOT VALID,
    CONSTRAINT book_year_check CHECK (book_year > 1900 AND
book_year < 2100) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.book
    OWNER to postgres;
```

Запрос для создания таблиц `category`

```
CREATE TABLE public.category
(
    category_id integer NOT NULL GENERATED ALWAYS AS IDENTITY (
INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    category character varying(200) COLLATE pg_catalog."default"
NOT NULL,
    CONSTRAINT "Category_pkey" PRIMARY KEY (category_id)
)

TABLESPACE pg_default;

ALTER TABLE public.category
    OWNER to postgres;
```

Запрос для создания таблиц `category_book`

```
CREATE TABLE public.category_book
(
    book_id integer NOT NULL,
    category_id integer NOT NULL,
    CONSTRAINT book_id FOREIGN KEY (book_id)
        REFERENCES public.book (book_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT category_id FOREIGN KEY (category_id)
        REFERENCES public.category (category_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE public.category_book
    OWNER to postgres;
```

Запрос для создания таблиц `client`

```
CREATE TABLE public.client
(
    client_id integer NOT NULL GENERATED ALWAYS AS IDENTITY (
INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    client_name character varying(50) COLLATE
pg_catalog."default" NOT NULL,
    client_surname character varying(50) COLLATE
pg_catalog."default" NOT NULL,
    client_patronymic character varying(50) COLLATE
pg_catalog."default",
    client_address character varying(200) COLLATE
pg_catalog."default" NOT NULL,
    client_phone character varying(13) COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT "Client_pkey" PRIMARY KEY (client_id),
    CONSTRAINT client_phone_check CHECK (client_phone::text !~~
'%[^0-9]%'::text) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.client
    OWNER to postgres;
```

Запрос для создания таблиц `edition`

```
CREATE TABLE public.edition
(
    edition_date date NOT NULL,
    edition_size integer NOT NULL,
    edition_warehouse integer NOT NULL,
    edition_price numeric NOT NULL,
    book_id integer NOT NULL,
    order_id integer NOT NULL,
    isbn character varying(13) COLLATE pg_catalog."default" NOT
NULL,
    CONSTRAINT "Edition_pkey" PRIMARY KEY (isbn),
    CONSTRAINT book_id FOREIGN KEY (book_id)
        REFERENCES public.book (book_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT order_id FOREIGN KEY (order_id)
        REFERENCES public."order" (order_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT edition_size_check CHECK (edition_size > 0 AND
edition_size < 100000) NOT VALID,
    CONSTRAINT edition_warehouse_check CHECK (edition_warehouse
> 0 AND edition_warehouse < 100000) NOT VALID,
    CONSTRAINT edition_price_check CHECK (edition_price >
0::numeric AND edition_price < 100000::numeric) NOT VALID,
    CONSTRAINT edition_date_check CHECK (edition_date >= '1990-01-01'::date AND edition_date <= '2100-01-01'::date) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.edition
    OWNER to postgres;
```

Запрос для создания таблиц `manager`

```
CREATE TABLE public.manager
(
    manager_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    manager_phone character varying(13) COLLATE pg_catalog."default" NOT NULL,
    manager_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
    manager_surname character varying(50) COLLATE pg_catalog."default" NOT NULL,
    manager_patronymic character varying(50) COLLATE pg_catalog."default",
    CONSTRAINT "Manager_pkey" PRIMARY KEY (manager_id),
    CONSTRAINT manager_phone_check CHECK (manager_phone::text !~~ '%[^0-9]%'::text) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.manager
    OWNER to postgres;
```

Запрос для создания таблиц `order`

```
CREATE TABLE public."order"
(
    order_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    order_date date NOT NULL,
    order_finish date NOT NULL,
    order_size integer NOT NULL,
    order_status "char" NOT NULL,
    client_id integer NOT NULL,
    manager_id integer NOT NULL,
    CONSTRAINT "Order_pkey" PRIMARY KEY (order_id),
    CONSTRAINT client_id FOREIGN KEY (client_id)
        REFERENCES public.client (client_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT manager_id FOREIGN KEY (manager_id)
        REFERENCES public.manager (manager_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT order_size_check CHECK (order_size > 0 AND order_size < 100000) NOT VALID,
    CONSTRAINT order_date_check CHECK (order_date >= '1990-01-01'::date AND order_date <= '2100-01-01'::date) NOT VALID,
    CONSTRAINT order_finish_check CHECK (order_finish >= '1990-01-01'::date AND order_finish <= '2100-01-01'::date) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public."order"
    OWNER to postgres;
```

Запросы вставки данных

```
INSERT INTO author (author_name, author_surname, author_patronymic, author_email)
VALUES
     ('Иван', 'Иванов', 'Иванович', 'ivan@gmail.com'),
     ('Петр', 'Петров', 'Петрович', 'petr@gmail.com');
SELECT * FROM author;

INSERT INTO book (book_pages, book_illustrated, book_year, book_name)
VALUES
     (100, true, 1999, 'Колобок'),
     (200, false, 2010, 'Русские народные сказки');
SELECT * FROM book;

INSERT INTO category (category)
VALUES
     ('Сказки');
SELECT * FROM category;

INSERT INTO client (client_name, client_surname, client_patronymic, client_address, client_phone)
VALUES
     ('Олег', 'Олегов', NULL, 'Россия, Санкт-Петербруг, Невский пр., 1', '79211111111'),
     ('Мария', 'Сидоровна', NULL, 'Россия, Санкт-Петербруг, Кронверкский пр., 1', '79212222222');
SELECT * FROM client;

INSERT INTO manager (manager_name, manager_surname, manager_patronymic, manager_phone)
VALUES
     ('Василий', 'Васильев', NULL, '79213333333');
SELECT * FROM manager;



INSERT INTO author_book (author_id, book_id, "order")
```

```
VALUES
     (75, 75, 1),
     (76, 76, 1);
SELECT * FROM author_book;

INSERT INTO category_book (book_id, category_id)
VALUES
     (75, 35),
     (76, 35);
SELECT * FROM category_book;

INSERT INTO "order" (order_date, order_finish, order_size, order_status, client_id, manager_id)
VALUES
     ('2021-04-18', '2021-05-14', 6000, 'p', 39, 1);
SELECT * FROM "order";



INSERT INTO edition (edition_date, edition_size, edition_warehouse, edition_price, book_id, order_id, isbn)
VALUES
     ('2021-04-20', 1000, 500, 220.0, 75, 1, '123456789012'),
     ('2021-04-23', 5000, 1000, 499.9, 76, 1, '0123456789012');
SELECT * FROM edition;
```

Выводы

В результате выполнения работы была создана БД и заполнена данными, учтены все ограничение по типу и формату данных.