

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Лабораторная работа № 3

на тему
“Создание таблиц базы данных POSTGRESQL. Заполнение таблиц рабочими данными”
по дисциплине «**Базы данных**»

Автор: Исхакова Эмина

Факультет: ИКТ

Группа: К3242



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2021

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

2. Практическое задание.

Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).

Создать схему в составе базы данных.

Создать таблицы базы данных.

Установить ограничения на данные: *Primary Key, Unique, Check, Foreign Key*.

Заполнить таблицы БД рабочими данными.

Создать резервную копию БД.

Указание:

Создать две резервные копии:

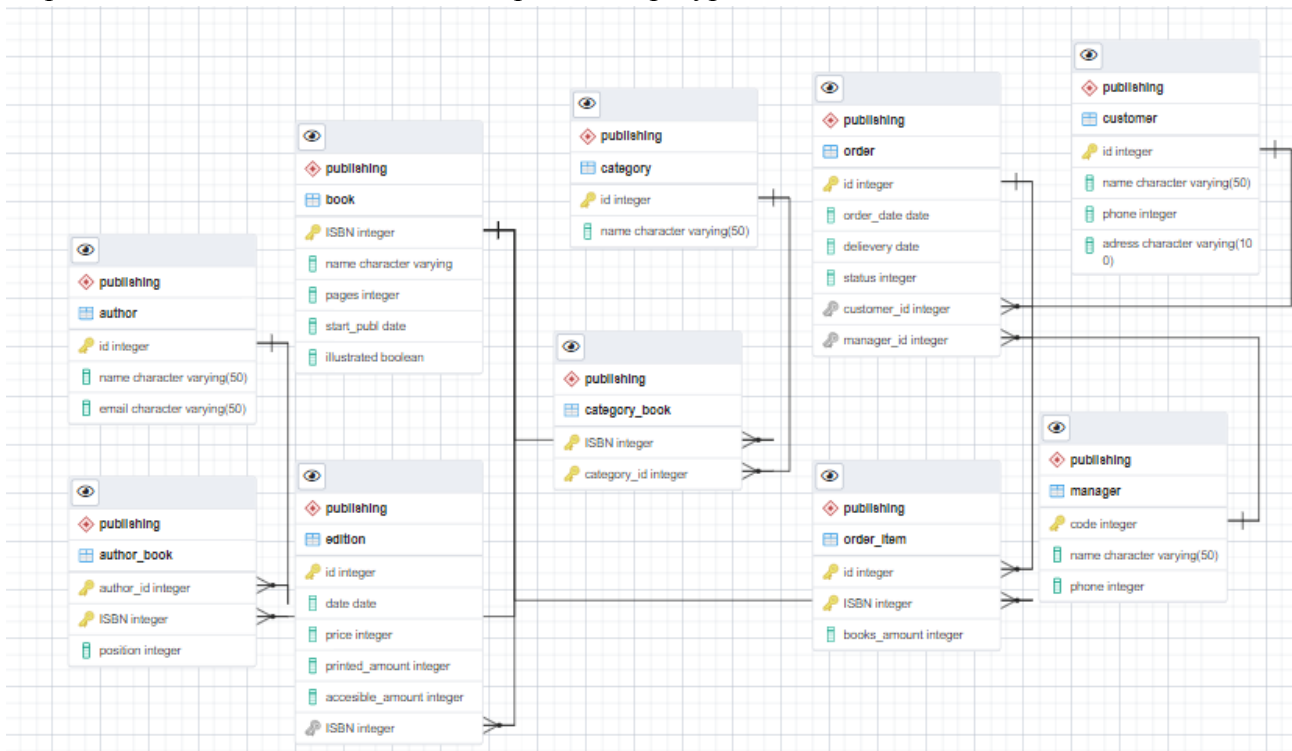
- с расширением *CUSTOM* для восстановления БД;
- с расширением *PLAIN* для листинга (в отчете);
- при создании резервных копий БД настроить параметры *Dump options* для

Type of objects и Queries .

Восстановить БД.

3. Выполнение:

Вариант 5. «Издательство компьютерной литературы»



-- SCHEMA: publishing - СОЗДАНИЕ СХЕМЫ

-- DROP SCHEMA publishing ;

CREATE SCHEMA publishing
AUTHORIZATION postgres;

-- DROP TABLE publishing.author; - СОЗДАНИЕ ТАБЛИЦ (здесь дефолтные ограничения только для ПК)

CREATE TABLE publishing.author

(
id integer NOT NULL,
name character varying(50) COLLATE pg_catalog."default" NOT NULL,
email character varying(50) COLLATE pg_catalog."default" NOT NULL,
CONSTRAINT author_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

-- Table: publishing.category

-- DROP TABLE publishing.category;

CREATE TABLE publishing.category

(
id integer NOT NULL,
name character varying(50) COLLATE pg_catalog."default" NOT NULL,
CONSTRAINT category_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE publishing.category
OWNER to postgres;

-- Table: publishing.customer

-- DROP TABLE publishing.customer;

CREATE TABLE publishing.customer

(
id integer NOT NULL,
name character varying(50) COLLATE pg_catalog."default" NOT NULL,
phone integer NOT NULL,
adress character varying(100) COLLATE pg_catalog."default" NOT NULL,
CONSTRAINT customer_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE publishing.customer
OWNER to postgres;

-- Table: publishing.manager

```
-- DROP TABLE publishing.manager;
```

```
CREATE TABLE publishing.manager  
(  
    code integer NOT NULL,  
    name character varying(50) COLLATE pg_catalog."default" NOT NULL,  
    phone integer NOT NULL,  
    CONSTRAINT manager_pkey PRIMARY KEY (code)  
)
```

```
TABLESPACE pg_default;
```

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

СОЗДАНИЕ ТАБЛИЦ С УСТАНОВЛЕНИЕМ ОГРАНИЧЕНИЙ (цена больше 0)

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

```
-- Table: publishing.edition
```

```
-- DROP TABLE publishing.edition;
```

```
CREATE TABLE publishing.edition  
(  
    id integer NOT NULL,  
    date date NOT NULL,  
    price integer NOT NULL,  
    printed_amount integer NOT NULL,  
    accesible_amount integer NOT NULL,  
    "ISBN" integer NOT NULL,  
    CONSTRAINT edition_pkey PRIMARY KEY (id),  
    CONSTRAINT isbn FOREIGN KEY ("ISBN")  
        REFERENCES publishing.book ("ISBN") MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID,  
    CONSTRAINT "price check" CHECK (price > 0),  
    CONSTRAINT "print - access" CHECK (printed_amount >= accesible_amount) NOT  
VALID  
)
```

```
TABLESPACE pg_default;
```

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

-- Table: publishing.author_book

-- DROP TABLE publishing.author_book;

```
CREATE TABLE publishing.author_book
(
    author_id integer NOT NULL,
    "ISBN" integer NOT NULL,
    "position" integer NOT NULL,
    CONSTRAINT author_book_pkey PRIMARY KEY (author_id, "ISBN"),
    CONSTRAINT id FOREIGN KEY (author_id)
        REFERENCES publishing.author (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT isbn FOREIGN KEY ("ISBN")
        REFERENCES publishing.book ("ISBN") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT pos CHECK ("position" > 0) NOT VALID
)
```

TABLESPACE pg_default;

ALTER TABLE publishing.author_book
OWNER to postgres;

-- Table: publishing.book

-- DROP TABLE publishing.book;

```
CREATE TABLE publishing.book
(
    "ISBN" integer NOT NULL,
    name character varying COLLATE pg_catalog."default" NOT NULL,
    pages integer NOT NULL,
    start_publ date NOT NULL,
    illustrated boolean NOT NULL,
    CONSTRAINT book_pkey PRIMARY KEY ("ISBN"),
    CONSTRAINT num_pages CHECK (pages > 0) положительное кол-во страниц
)
```

TABLESPACE pg_default;

ALTER TABLE publishing.book
OWNER to postgres;

-- Table: publishing.category_book

```
-- DROP TABLE publishing.category_book;
```

```
CREATE TABLE publishing.category_book
(
  "ISBN" integer NOT NULL,
  category_id integer NOT NULL,
  CONSTRAINT category_book_pkey PRIMARY KEY ("ISBN", category_id),
  CONSTRAINT id FOREIGN KEY (category_id)
    REFERENCES publishing.category (id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID,
  CONSTRAINT isbn FOREIGN KEY ("ISBN")
    REFERENCES publishing.book ("ISBN") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID
)
```

```
TABLESPACE pg_default;
```

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

```
-- Table: publishing.order
```

```
-- DROP TABLE publishing."order";
```

```
CREATE TABLE publishing."order"
(
  id integer NOT NULL,
  order_date date NOT NULL,
  delievery date NOT NULL,
  status integer NOT NULL,
  customer_id integer NOT NULL,
  manager_id integer NOT NULL,
  CONSTRAINT order_pkey PRIMARY KEY (id),      ограничения по ключам
  CONSTRAINT customer FOREIGN KEY (customer_id)
    REFERENCES publishing.customer (id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID,
  CONSTRAINT manager FOREIGN KEY (manager_id)
    REFERENCES publishing.manager (code) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE publishing."order"
```

```

OWNER to postgres;

-- Table: publishing.order_item

-- DROP TABLE publishing.order_item;

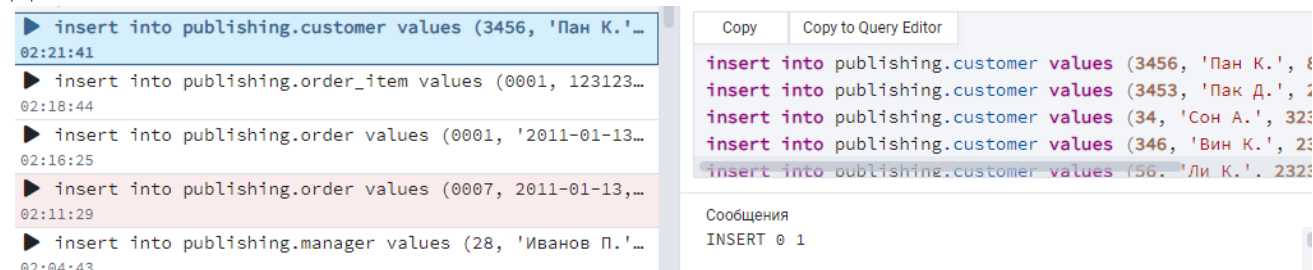
CREATE TABLE publishing.order_item
(
    id integer NOT NULL,
    "ISBN" integer NOT NULL,
    books_amount integer NOT NULL,
    CONSTRAINT order_item_pkey PRIMARY KEY (id, "ISBN"),
    CONSTRAINT id FOREIGN KEY (id)
        REFERENCES publishing."order" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT isbn FOREIGN KEY ("ISBN")
        REFERENCES publishing.book ("ISBN") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT amount CHECK (books_amount > 0) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE publishing.order_item
    OWNER to postgres;

```

Добавление данных в таблицы:



4. Выводы.

В ходе выполнения лабораторной работы были созданы таблицы базы данных PostgreSQL 1X. Были установлены ограничения на данные (первичный и внешний ключи, проверки на наличие и корректность значения). Затем таблицы были заполнены рабочими данными. Были созданы две резервные копии (в текстовом и кастомном вариантах), первая использовалась для листинга в отчете, а с помощью второй было произведено восстановление базы данных.