

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

Факультет инфокоммуникационных технологий

**Лабораторная работа No 1 «Создание
таблицы базы данных PostgreSQL.
Заполнение таблицы рабочими данными»
по дисциплине «Проектирование и
реализация баз данных»**

Выполнил:
студент 2 курса ФИКТ
группы K3241
Тимофеев Н.А.

Проверил:
Говорова М.В.

Санкт-Петербург
2022

Цель 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.

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.

Индивидуальное задание. Вариант 5. БД «Издательство компьютерной литературы»

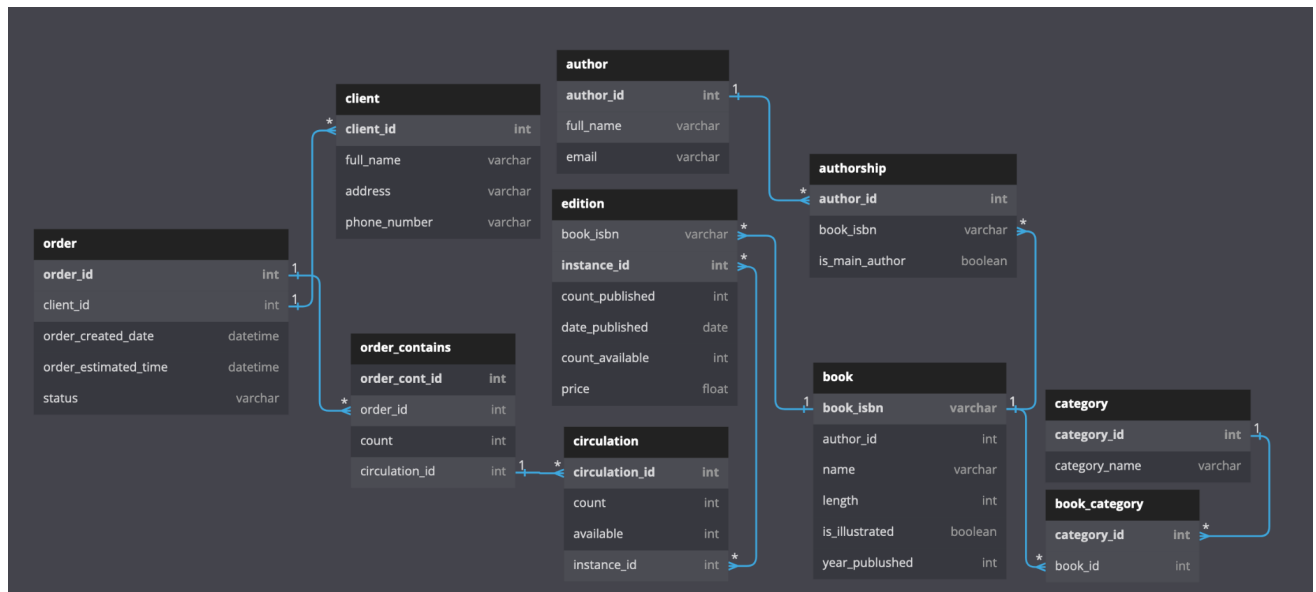
Описание предметной области: Издательство занимается выпуском литературы по различным областям информатики. Покупатели книг приобретают книги на базе издательства. Когда на базе заканчиваются книги, издается дополнительный тираж. В каждом заказе заказчик может заказать разную литературу.

БД должна содержать следующий минимальный набор сведений: Фамилия автора. Имя автора. Отчество автора. Код автора. E-mail автора. Код ISBN. Название книги. Количество страниц. Наличие иллюстраций. Код категории книги. Категория книги. Количество страниц. Год начала издания. Розничная цена книги. Тираж. Дата тиража. Количество экземпляров на базе издательства. Код заказчика. Фамилия заказчика. Имя заказчика. Отчество заказчика. Адрес заказчика. Телефон заказчика. Код заказа. Дата заказа. Срок заказа. Количество экземпляров книги в заказе. Статус заказа.

1. БД «Издательство компьютерной литературы»
2. Сущности:
 - a. Писатель (Код автора (PK), ФИО автора, Электронная почта автора)
 - b. Книга (Код ISBN (PK), название книги, количество страниц, наличие иллюстраций, год начала издания)
 - c. Категория (Код категории книги (PK), название категории)
 - d. Издание (Номер издания (PK), дата тиража, Код ISBN, тираж, кол-во экземпляров на базе издательства, код заказа, розничная цена книги)

- e. Заказчик (Код заказчика (PK), ФИО заказчика, адрес заказчика, телефон заказчика)
- f. Заказ (Код заказа (PK), код заказчика, дата заказа, срок заказа, статус заказа)
- g. Состав заказа (Номер состава заказа (PK), номер заказа, номер издания, кол-во экземпляров книги в заказе)

3. Схема логической модели базы данных



4. Backup (plain):

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 14.4 (Debian 14.4-1.pgdg110+1)
-- Dumped by pg_dump version 14.4

-- Started on 2022-10-04 20:36:12 +03

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_table_access_method = heap;

CREATE TABLE public.author (
    author_id integer NOT NULL,
```

```
        full_name character varying,  
        email character varying  
    );  
  
ALTER TABLE public.author OWNER TO postgres;
```

```
CREATE TABLE public.authorship (  
    author_id integer NOT NULL,  
    book_isbn character varying,  
    is_main_author boolean,  
    id integer  
);
```

```
ALTER TABLE public.authorship OWNER TO postgres;
```

```
CREATE TABLE public.book (  
    book_isbn character varying NOT NULL,  
    name character varying,  
    length integer,  
    is_illustrated boolean,  
    year_published integer  
);
```

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

```
CREATE TABLE public.circulation (  
    circulation_id integer NOT NULL,  
    count integer,  
    available integer,  
    instance_id integer  
);
```

```
ALTER TABLE public.circulation OWNER TO postgres;
```

```
CREATE TABLE public.edition (  
    book_isbn character varying,  
    instance_id integer NOT NULL,  
    date_published date,  
    price double precision  
);
```

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

```
CREATE TABLE public.order_contains (  
    order_cont_id integer NOT NULL,  
    order_id integer,  
    count integer,  
    circulation_id integer  
);
```

```
ALTER TABLE public.order_contains OWNER TO postgres;
```

```
CREATE VIEW public.book_best sellers AS
SELECT book.name,
       sum(oc.count) AS sum
FROM (((public.book
       JOIN public.edition ON (((book.book_isbn)::text = (edition.book_isbn)::text)))
       JOIN public.circulation c ON ((edition.instance_id = c.instance_id)))
       JOIN public.order_contains oc ON ((c.circulation_id = oc.circulation_id)))
GROUP BY book.name;
```

```
ALTER TABLE public.book_best sellers OWNER TO postgres;
```

```
CREATE TABLE public.book_category (
    category_id integer NOT NULL,
    book_id character varying
);
```

```
ALTER TABLE public.book_category OWNER TO postgres;
```

```
CREATE TABLE public.category (
    category_id integer NOT NULL,
    category_name character varying
);
```

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

```
CREATE TABLE public.client (
    client_id integer NOT NULL,
    full_name character varying,
    address character varying,
    phone_number character varying
);
```

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

```
CREATE TABLE public."order" (
    order_id integer NOT NULL,
    client_id integer,
    order_created_date date,
    order_estimated_time date,
    status character varying
);
```

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

```
CREATE VIEW public.customers_yearly_rating AS
SELECT client.full_name,
       count(*) AS orc_cnt
FROM ((public.client
       JOIN public."order" o ON ((client.client_id = o.client_id)))
       JOIN public.order_contains oc ON ((o.order_id = oc.order_id)))
GROUP BY client.full_name
ORDER BY (count(*)) DESC;
```

```
ALTER TABLE public.customers_yearly_rating OWNER TO postgres;
```

```
COPY public.author (author_id, full_name, email) FROM stdin;
```

```
1 Mikolas Timofeev timofeev412@gmail.com
3 Nikolaç Tâmoŧçyu mymail@google.com
4 Anna Timofeeva anne@google.com
5 Zarina Zemljanichka zemljanichka@google.com
2 Some author asd@exmaple.com
\.
```

```
COPY public.authorship (author_id, book_isbn, is_main_author, id) FROM stdin;
```

```
5 1 f 3
1 1 t 1
2 1 t 2
4 2 t 4
5 4 t 6
2 3 t 5
\.
```

```
COPY public.book (book_isbn, name, length, is_illustrated, year_published) FROM stdin;
```

```
1 Python programming 1120 f 2022
2 Art and science 122 t 2021
3 Popular Mechanics 320 t 2020
4 Guide on ORM in Python 220 t 2022
5 PostgreSQL v13 Handbook 1200 t 2019
6 Clean Architecture 670 t 2022
\.
```

```
COPY public.book_category (category_id, book_id) FROM stdin;
```

```
1 1
2 4
3 1
\.
```

```
COPY public.category (category_id, category_name) FROM stdin;
```

```
1 Programming
2 Physics
3 Tech
4 Other
\.
```

```
COPY public.circulation (circulation_id, count, available, instance_id) FROM stdin;
```

```
1 100 91 1
2 500 11 2
3 80 0 3
4 10 2 4
5 1000 821 5
6 1200 672 6
7 12 2 7
\.
```

```
COPY public.client (client_id, full_name, address, phone_number) FROM stdin;
```

```
1 Ivan Semenov 16th line vo 121 kv 3 +79533435285
2 Peter Zyablow 2th line vo 11B kv 32 +79533435281
3 Inna Maina lanskoye shosse 21k1 kv 122 +79535555281
```

\\.

```
COPY public.edition (book_isbn, instance_id, date_published, price) FROM stdin;
```

1	1	2002-10-02	1000
1	2	2023-10-02	1400
2	3	2019-11-02	990
2	4	2022-03-02	9990
4	7	2019-10-09	2900
3	5	2015-10-02	180
3	6	2017-10-02	270

\\.

```
COPY public."order" (order_id, client_id, order_created_date, order_estimated_time, status) FROM stdin;
```

1	3	2022-09-11	2022-11-19	finished
2	3	2022-11-11	2022-11-21	finished
3	3	2022-11-21	2022-11-29	process
4	4	2022-11-22	\\N	canceled
5	2	2022-09-15	\\N	canceled
6	2	2022-09-17	2022-09-27	finished
7	1	2022-09-21	2022-09-29	shipping
8	2	2022-02-21	2022-02-28	finished
9	4	2022-10-02	2022-10-12	process

\\.

```
COPY public.order_contains (order_cont_id, order_id, count, circulation_id) FROM stdin;
```

1	1	9	2
2	1	1	4
3	1	1	6
4	2	1	2
5	3	1	6
6	3	1	5
7	4	1	1
8	4	1	7
9	5	1	7
10	6	1	7
11	6	1	5
12	6	1	1
13	7	3	3
14	7	2	6
15	8	1	2

\\.

```
ALTER TABLE ONLY public.author  
    ADD CONSTRAINT author_pkey PRIMARY KEY (author_id);
```

```
ALTER TABLE ONLY public.book_category  
    ADD CONSTRAINT book_category_pkey PRIMARY KEY (category_id);
```

```
ALTER TABLE ONLY public.book  
    ADD CONSTRAINT book_pkey PRIMARY KEY (book_isbn);
```

```
ALTER TABLE ONLY public.category  
    ADD CONSTRAINT category_pkey PRIMARY KEY (category_id);
```

```
ALTER TABLE ONLY public.circulation  
    ADD CONSTRAINT circulation_pkey PRIMARY KEY (circulation_id);
```

```

ALTER TABLE ONLY public.client
    ADD CONSTRAINT client_pkey PRIMARY KEY (client_id);

ALTER TABLE ONLY public.edition
    ADD CONSTRAINT edition_pkey PRIMARY KEY (instance_id);

ALTER TABLE ONLY public.order_contains
    ADD CONSTRAINT order_contains_pkey PRIMARY KEY (order_cont_id);

ALTER TABLE ONLY public."order"
    ADD CONSTRAINT order_pkey PRIMARY KEY (order_id);

CREATE INDEX available_books ON public.circulation USING btree (circulation_id, available);

CREATE INDEX category_name ON public.category USING btree (category_name);

ALTER TABLE ONLY public.authorship
    ADD CONSTRAINT authorship_author_id_fkey FOREIGN KEY (author_id)
    REFERENCES public.author(author_id);

ALTER TABLE ONLY public.authorship
    ADD CONSTRAINT authorship_book_isbn_fkey FOREIGN KEY (book_isbn)
    REFERENCES public.book(book_isbn);

ALTER TABLE ONLY public.book_category
    ADD CONSTRAINT book_category_book_id_fkey FOREIGN KEY (book_id)
    REFERENCES public.book(book_isbn);

ALTER TABLE ONLY public.book_category
    ADD CONSTRAINT book_category_category_id_fkey FOREIGN KEY (category_id)
    REFERENCES public.category(category_id);

ALTER TABLE ONLY public.edition
    ADD CONSTRAINT edition_book_isbn_fkey FOREIGN KEY (book_isbn) REFERENCES
public.book(book_isbn);

-- Completed on 2022-10-04 20:36:12 +03

--
-- PostgreSQL database dump complete
--

```

Вывод: Я овладел навыками создания таблиц базы данных Postgres, заполнения их рабочими данными (с помощью INSERT INTO), резервного копирования (Backup) и восстановления БД (Restore)