

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 1

по теме:

*«Создание таблиц базы данных POSTGRESQL. Заполнение
таблиц рабочими данными»*

по дисциплине: Проектирование и реализация баз данных

Специальность:

45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверила:

Говорова М.М.

Дата: «29» мая 2023 г.

Оценка _____

Выполнила:

студентка группы К32422

Королева Е. М.

Санкт-Петербург 2022/2023

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

Оборудование: компьютерный класс.

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.

Предметная область: Вариант 3. БД «Библиотека»

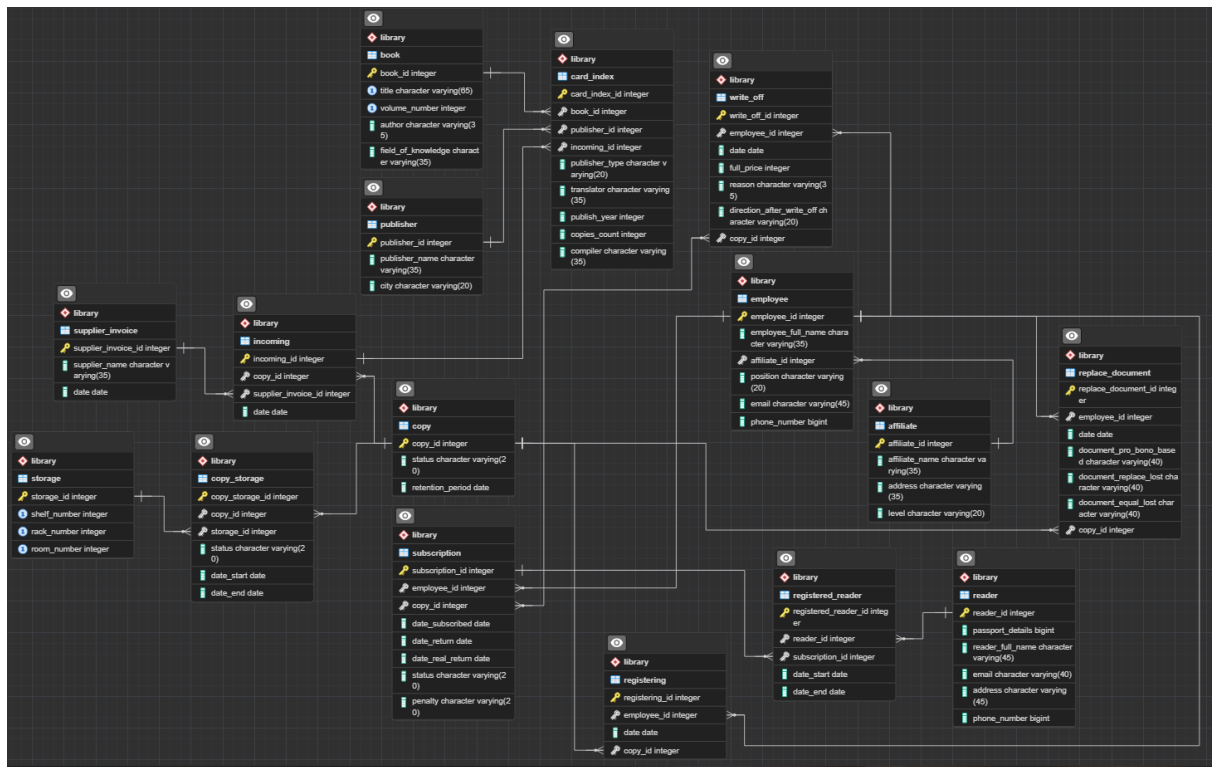


Рисунок 1 – ERD базы данных с pgAdmin

Выполнение работы:

Создадим резервную копию нашей бд используя инструмент pg_dump:

1. Создание схемы

```
CREATE SCHEMA library;
ALTER SCHEMA library OWNER TO postgres;
SET default_tablespace = '';
SET default_table_access_method = heap;
```

2. Создание таблиц

```
CREATE TABLE library.affiliate (
    affiliate_id integer NOT NULL,
    affiliate_name character varying(35) NOT NULL,
    address character varying(35) NOT NULL,
    level character varying(20) NOT NULL
);

ALTER TABLE library.affiliate OWNER TO postgres;
```

```
CREATE SEQUENCE library.affiliate_affiliate_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE library.affiliate_affiliate_id_seq OWNER TO postgres;
ALTER SEQUENCE library.affiliate_affiliate_id_seq OWNED BY
library.affiliate.affiliate_id;
```

```
CREATE TABLE library.book (
  book_id integer NOT NULL,
  title character varying(65) NOT NULL,
  volume_number integer NOT NULL,
  author character varying(35) NOT NULL,
  field_of_knowledge character varying(35) NOT NULL
);

ALTER TABLE library.book OWNER TO postgres;

CREATE SEQUENCE library.book_book_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE library.book_book_id_seq OWNER TO postgres;

ALTER SEQUENCE library.book_book_id_seq OWNED BY library.book.book_id;
```

```
CREATE TABLE library.card_index (
  card_index_id integer NOT NULL,
  book_id integer NOT NULL,
  publisher_id integer NOT NULL,
  incoming_id integer NOT NULL,
  publisher_type character varying(20) NOT NULL,
  translator character varying(35) NOT NULL,
  publish_year integer NOT NULL,
  copies_count integer NOT NULL,
  compiler character varying(35) NOT NULL
);

ALTER TABLE library.card_index OWNER TO postgres;
```

```

CREATE TABLE library.copy (
    copy_id integer NOT NULL,
    status character varying(20) NOT NULL,
    retention_period date NOT NULL
);

ALTER TABLE library.copy OWNER TO postgres;

CREATE SEQUENCE library.copy_copy_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library.copy_copy_id_seq OWNER TO postgres;
ALTER SEQUENCE library.copy_copy_id_seq OWNED BY library.copy.copy_id;

```

```

CREATE TABLE library.copy_storage (
    copy_storage_id integer NOT NULL,
    copy_id integer NOT NULL,
    storage_id integer NOT NULL,
    status character varying(20) NOT NULL,
    date_start date NOT NULL,
    date_end date NOT NULL
);

ALTER TABLE library.copy_storage OWNER TO postgres;

CREATE SEQUENCE library.copy_storage_copy_storage_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library.copy_storage_copy_storage_id_seq OWNER TO postgres;

ALTER SEQUENCE library.copy_storage_copy_storage_id_seq OWNED BY
library.copy_storage.copy_storage_id;

```

```

CREATE TABLE library.employee (
    employee_id integer NOT NULL,
    employee_full_name character varying(35) NOT NULL,
    affiliate_id integer NOT NULL,
    "position" character varying(20) NOT NULL,
    email character varying(45) NOT NULL,
    phone_number bigint NOT NULL
);

ALTER TABLE library.employee OWNER TO postgres;

```

```
CREATE SEQUENCE library.employee_employee_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE library.employee_employee_id_seq OWNER TO postgres;
ALTER SEQUENCE library.employee_employee_id_seq OWNED BY library.employee.employee_id;
```

```
CREATE TABLE library.incoming (
  incoming_id integer NOT NULL,
  copy_id integer NOT NULL,
  supplier_invoice_id integer NOT NULL,
  date date NOT NULL
);

ALTER TABLE library.incoming OWNER TO postgres;

CREATE SEQUENCE library.incoming_incoming_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE library.incoming_incoming_id_seq OWNER TO postgres;
ALTER SEQUENCE library.incoming_incoming_id_seq OWNED BY library.incoming.incoming_id;
```

```
CREATE TABLE library.publisher (
  publisher_id integer NOT NULL,
  publisher_name character varying(35) NOT NULL,
  city character varying(20) NOT NULL
);

ALTER TABLE library.publisher OWNER TO postgres;

CREATE SEQUENCE library.publisher_publisher_id_seq
  AS integer
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

ALTER TABLE library.publisher_publisher_id_seq OWNER TO postgres;
ALTER SEQUENCE library.publisher_publisher_id_seq OWNED BY
library.publisher.publisher_id;
```

```
CREATE TABLE library.reader (  
    reader_id integer NOT NULL,  
    passport_details bigint NOT NULL,  
    reader_full_name character varying(45) NOT NULL,  
    email character varying(40) NOT NULL,  
    address character varying(45) NOT NULL,  
    phone_number bigint NOT NULL  
);  
  
ALTER TABLE library.reader OWNER TO postgres;  
  
CREATE SEQUENCE library.reader_reader_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
  
ALTER TABLE library.reader_reader_id_seq OWNER TO postgres;  
  
ALTER SEQUENCE library.reader_reader_id_seq OWNED BY library.reader.reader_id;
```

```
CREATE TABLE library.registered_reader (  
    registered_reader_id integer NOT NULL,  
    reader_id integer NOT NULL,  
    subscription_id integer NOT NULL,  
    date_start date NOT NULL,  
    date_end date NOT NULL  
);  
  
ALTER TABLE library.registered_reader OWNER TO postgres;  
  
CREATE SEQUENCE library.registered_reader_registered_reader_id_seq  
    AS integer  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;  
  
ALTER TABLE library.registered_reader_registered_reader_id_seq OWNER TO postgres;  
ALTER SEQUENCE library.registered_reader_registered_reader_id_seq OWNED BY  
library.registered_reader.registered_reader_id;
```

```

CREATE TABLE library.registering (
    registering_id integer NOT NULL,
    employee_id integer NOT NULL,
    date date NOT NULL,
    copy_id integer NOT NULL
);

ALTER TABLE library.registering OWNER TO postgres;

CREATE SEQUENCE library.registering_registering_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library.registering_registering_id_seq OWNER TO postgres;
ALTER SEQUENCE library.registering_registering_id_seq OWNED BY
library.registering.registering_id;

```

```

CREATE TABLE library.replace_document (
    replace_document_id integer NOT NULL,
    employee_id integer NOT NULL,
    date date NOT NULL,
    document_pro_bono_based character varying(40) NOT NULL,
    document_replace_lost character varying(40) NOT NULL,
    document_equal_lost character varying(40) NOT NULL,
    copy_id integer NOT NULL
);

ALTER TABLE library.replace_document OWNER TO postgres;

CREATE SEQUENCE library.replace_document_replace_document_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library.replace_document_replace_document_id_seq OWNER TO postgres;
ALTER SEQUENCE library.replace_document_replace_document_id_seq OWNED BY
library.replace_document.replace_document_id;

```



```
CREATE TABLE library.storage (
    storage_id integer NOT NULL,
    shelf_number integer NOT NULL,
    rack_number integer NOT NULL,
    room_number integer NOT NULL
);

ALTER TABLE library.storage OWNER TO postgres;

CREATE SEQUENCE library.storage_storage_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library.storage_storage_id_seq OWNER TO postgres;
ALTER SEQUENCE library.storage_storage_id_seq OWNED BY library.storage.storage_id;
```

```
CREATE TABLE library.subscription (
    subscription_id integer NOT NULL,
    employee_id integer NOT NULL,
    copy_id integer NOT NULL,
    date_subscribed date NOT NULL,
    date_return date NOT NULL,
    date_real_return date,
    status character varying(20) NOT NULL,
    penalty character varying(20)
);

ALTER TABLE library.subscription OWNER TO postgres;

CREATE SEQUENCE library.subscription_subscription_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library.subscription_subscription_id_seq OWNER TO postgres;
ALTER SEQUENCE library.subscription_subscription_id_seq OWNED BY
library.subscription.subscription_id;
```

```

CREATE TABLE library.supplier_invoice (
    supplier_invoice_id integer NOT NULL,
    supplier_name character varying(35) NOT NULL,
    date date NOT NULL
);

ALTER TABLE library.supplier_invoice OWNER TO postgres;

CREATE SEQUENCE library.supplier_invoice_supplier_invoice_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library.supplier_invoice_supplier_invoice_id_seq OWNER TO postgres;
ALTER SEQUENCE library.supplier_invoice_supplier_invoice_id_seq OWNED BY
library.supplier_invoice.supplier_invoice_id;

```

```

CREATE TABLE library.write_off (
    write_off_id integer NOT NULL,
    employee_id integer NOT NULL,
    date date NOT NULL,
    full_price integer NOT NULL,
    reason character varying(35) NOT NULL,
    direction_after_write_off character varying(20) NOT NULL,
    copy_id integer NOT NULL
);

ALTER TABLE library.write_off OWNER TO postgres;

CREATE SEQUENCE library.write_off_write_off_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library.write_off_write_off_id_seq OWNER TO postgres;
ALTER SEQUENCE library.write_off_write_off_id_seq OWNED BY
library.write_off.write_off_id;

CREATE SEQUENCE library."card_Index_card_Index_id_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE library."card_Index_card_Index_id_seq" OWNER TO postgres;
ALTER SEQUENCE library."card_Index_card_Index_id_seq" OWNED BY
library.card_index.card_index_id;

```

3. Установка значений по умолчанию для первичных ключей

```
ALTER TABLE ONLY library.affiliate ALTER COLUMN affiliate_id SET DEFAULT
nextval('library.affiliate_affiliate_id_seq'::regclass);

ALTER TABLE ONLY library.book ALTER COLUMN book_id SET DEFAULT
nextval('library.book_book_id_seq'::regclass);

ALTER TABLE ONLY library.card_index ALTER COLUMN card_index_id SET DEFAULT
nextval('library."card_Index_card_Index_id_seq"'::regclass);

ALTER TABLE ONLY library.copy ALTER COLUMN copy_id SET DEFAULT
nextval('library.copy_copy_id_seq'::regclass);

ALTER TABLE ONLY library.copy_storage ALTER COLUMN copy_storage_id SET DEFAULT
nextval('library.copy_storage_copy_storage_id_seq'::regclass);

ALTER TABLE ONLY library.employee ALTER COLUMN employee_id SET DEFAULT
nextval('library.employee_employee_id_seq'::regclass);

ALTER TABLE ONLY library.incoming ALTER COLUMN incoming_id SET DEFAULT
nextval('library.incoming_incoming_id_seq'::regclass);

ALTER TABLE ONLY library.publisher ALTER COLUMN publisher_id SET DEFAULT
nextval('library.publisher_publisher_id_seq'::regclass);

ALTER TABLE ONLY library.reader ALTER COLUMN reader_id SET DEFAULT
nextval('library.reader_reader_id_seq'::regclass);

ALTER TABLE ONLY library.registered_reader ALTER COLUMN registered_reader_id SET DEFAULT
nextval('library.registered_reader_registered_reader_id_seq'::regclass);

ALTER TABLE ONLY library.registering ALTER COLUMN registering_id SET DEFAULT
nextval('library.registering_registering_id_seq'::regclass);

ALTER TABLE ONLY library.replace_document ALTER COLUMN replace_document_id SET DEFAULT
nextval('library.replace_document_replace_document_id_seq'::regclass);

ALTER TABLE ONLY library.storage ALTER COLUMN storage_id SET DEFAULT
nextval('library.storage_storage_id_seq'::regclass);

ALTER TABLE ONLY library.subscription ALTER COLUMN subscription_id SET DEFAULT
nextval('library.subscription_subscription_id_seq'::regclass);

ALTER TABLE ONLY library.supplier_invoice ALTER COLUMN supplier_invoice_id SET DEFAULT
nextval('library.supplier_invoice_supplier_invoice_id_seq'::regclass);

ALTER TABLE ONLY library.write_off ALTER COLUMN write_off_id SET DEFAULT
nextval('library.write_off_write_off_id_seq'::regclass);
```

4. Заполнение таблиц через поток ввода

```
COPY library.affiliate (affiliate_id, affiliate_name, address, level) FROM stdin;
```

```
1      Филиал имени Пушкина ул. Литературная, 1      Высокий
2      Филиал имени Лермонтова пр. Поэтический, 5      Средний
3      Филиал имени Толстого ул. Прозаическая, 10      Средний
4      Филиал имени Достоевского пл. Романовская, 3      Высокий
5      Филиал имени Чехова ул. Драматическая, 7      Низкий
6      Филиал имени Гоголя пр. Сатирический, 2      Низкий
7      Филиал имени Тургенева ул. Реалистическая, 9      Средний
8      Филиал имени Чуковского пр. Детский, 4      Низкий
9      Филиал имени Пастернака ул. Лирическая, 12      Средний
10     Филиал имени Блока пр. Символический, 8      Высокий
11     Филиал имени Маяковского ул. Революционная, 6      Средний
12     Филиал имени Есенина пр. Лирический, 3      Высокий
13     Филиал имени Ахматовой ул. Эмоциональная, 15      Низкий
14     Филиал имени Бродского пр. Метафорический, 7      Средний
15     Филиал имени Паустовского ул. Автобиографическая, 10      Высокий
16     Филиал имени Фета пр. Лирический, 2      Низкий
17     Филиал имени Гумилева ул. Метафорическая, 8      Низкий
18     Филиал имени Ерофеева пр. Сатирический, 3      Средний
19     Филиал имени Андреева ул. Драматическая, 9      Высокий
20     Филиал имени Фадеева пр. Реалистический, 6      Высокий
21     Филиал имени Бунина ул. Литературная, 4      Низкий
22     Филиал имени Шолохова пр. Эпический, 5      Средний
23     Филиал имени Солженицына ул. Историческая, 11      Высокий
24     Филиал имени Булгакова пр. Фантастический, 9      Низкий
25     Филиал имени Айтматова ул. Реалистическая, 7      Низкий
26     Филиал имени Крылова пр. Сатирический, 6      Средний
27     Филиал имени Гайдара ул. Детская, 10      Низкий
28     Филиал имени Успенского пр. Фантастический, 8      Высокий
29     Филиал имени Аксакова ул. Литературная, 3      Средний
30     Филиал имени Островского пр. Драматический, 10      Низкий
31     Филиал имени Чернышевского ул. Реалистическая, 5      Высокий
32     Филиал имени Грибоедова пр. Сатирический, 7      Высокий
33     Филиал имени Ломоносова ул. Поэтическая, 9      Низкий
34     Филиал имени Фета пр. Лирический, 2      Низкий
35     Филиал имени Горького ул. Реалистическая, 8      Средний
36     Филиал имени Бальмонта пр. Символический, 4      Высокий
37     Филиал имени Есенина ул. Лирическая, 7      Низкий
38     Филиал имени Ахматовой пр. Эмоциональный, 3      Высокий
39     Филиал имени Бродского ул. Метафорическая, 6      Средний
40     Филиал имени Паустовского пр. Автобиографический, 8      Низкий
41     Филиал имени Фета ул. Лирическая, 2      Средний
42     Филиал имени Гумилева пр. Метафорический, 7      Высокий
43     Филиал имени Ерофеева ул. Сатирическая, 5      Низкий
44     Филиал имени Андреева пр. Драматический, 4      Средний
45     Филиал имени Фадеева ул. Реалистическая, 10      Высокий
46     Филиал имени Бунина пр. Литературный, 9      Высокий
47     Филиал имени Шолохова ул. Эпическая, 6      Низкий
48     Филиал имени Солженицына пр. Исторический, 5      Средний
49     Филиал имени Булгакова ул. Фантастическая, 3      Низкий
50     Филиал имени Айтматова пр. Реалистический, 8      Высокий
51     Филиал имени Крылова ул. Сатирическая, 9      Средний
52     Филиал имени Гайдара пр. Детский, 7      Низкий
53     Филиал имени Успенского ул. Фантастическая, 6      Высокий
```

54	Филиал имени Аксакова пр.	Литературный, 5	Средний
55	Филиал имени Островского ул.	Драматическая, 8	Низкий
56	Филиал имени Чернышевского пр.	Реалистический, 4	Высокий
57	Филиал имени Грибоедова ул.	Сатирическая, 7	Высокий
58	Филиал имени Ломоносова пр.	Поэтический, 6	Низкий
59	Филиал имени Фета ул.	Лирическая, 4	Низкий
60	Филиал имени Горького пр.	Реалистический, 7	Средний
61	Филиал имени Бальмонта ул.	Символическая, 5	Высокий
62	Филиал имени Есенина пр.	Лирический, 6	Низкий
63	Филиал имени Ахматовой ул.	Эмоциональная, 4	Высокий
64	Филиал имени Бродского пр.	Метафорический, 5	Средний
65	Филиал имени Паустовского ул.	Автобиографическая, 7	Низкий
66	Филиал имени Фета пр.	Лирический, 3	Средний
67	Филиал имени Гумилева ул.	Метафорическая, 4	Высокий
68	Филиал имени Ерофеева пр.	Сатирический, 6	Низкий
69	Филиал имени Андреева ул.	Драматическая, 5	Средний
70	Филиал имени Фадеева пр.	Реалистический, 9	Высокий
71	Филиал имени Бунина ул.	Литературная, 8	Высокий
72	Филиал имени Шолохова пр.	Эпический, 4	Низкий
73	Филиал имени Солженицына ул.	Историческая, 7	Средний
74	Филиал имени Булгакова пр.	Фантастический, 5	Низкий
75	Филиал имени Айтматова ул.	Реалистическая, 6	Высокий
76	Филиал имени Крылова пр.	Сатирический, 5	Средний
77	Филиал имени Гайдара ул.	Детская, 9	Низкий
78	Филиал имени Успенского пр.	Фантастический, 4	Высокий
79	Филиал имени Аксакова ул.	Литературная, 6	Средний
80	Филиал имени Островского пр.	Драматический, 7	Низкий
81	Филиал имени Чернышевского ул.	Реалистическая, 3	Высокий
82	Филиал имени Грибоедова пр.	Сатирический, 4	Высокий
83	Филиал имени Ломоносова ул.	Поэтическая, 5	Низкий
84	Филиал имени Фета пр.	Лирический, 2	Низкий
85	Филиал имени Горького ул.	Реалистическая, 6	Средний
86	Филиал имени Бальмонта пр.	Символический, 3	Высокий
87	Филиал имени Есенина ул.	Лирическая, 4	Низкий
88	Филиал имени Ахматовой пр.	Эмоциональная, 2	Высокий
89	Филиал имени Бродского ул.	Метафорическая, 3	Средний
90	Филиал имени Паустовского пр.	Автобиографический, 6	Низкий
91	Филиал имени Фета ул.	Лирическая, 1	Средний
92	Филиал имени Гумилева пр.	Метафорический, 2	Высокий
93	Филиал имени Ерофеева ул.	Сатирическая, 4	Низкий
94	Филиал имени Андреева пр.	Драматический, 3	Средний
95	Филиал имени Фадеева ул.	Реалистическая, 8	Высокий
96	Филиал имени Бунина пр.	Литературный, 7	Высокий
97	Филиал имени Шолохова ул.	Эпическая, 3	Низкий
98	Филиал имени Солженицына пр.	Исторический, 4	Средний
99	Филиал имени Булгакова ул.	Фантастическая, 2	Низкий
100	Филиал имени Айтматова пр.	Реалистический, 7	Высокий
101	Филиал имени Крылова ул.	Сатирическая, 8	Средний
102	Филиал имени Гайдара пр.	Детский, 6	Низкий
103	Филиал имени Успенского ул.	Фантастическая, 5	Высокий
104	Филиал имени Аксакова пр.	Литературный, 4	Средний
105	Филиал имени Островского ул.	Драматическая, 7	Низкий
106	Филиал имени Чернышевского пр.	Реалистический, 2	Высокий
107	Филиал имени Грибоедова ул.	Сатирическая, 3	Высокий
108	Филиал имени Ломоносова пр.	Поэтический, 4	Низкий
109	Филиал имени Фета ул.	Лирическая, 1	Низкий

110	Филиал имени Горького пр. Реалистический, 5 Средний
111	Филиал имени Бальмонта ул. Символическая, 3 Высокий
112	Филиал имени Есенина пр. Лирический, 2 Низкий
113	Филиал имени Ахматовой ул. Эмоциональная, 3 Высокий
114	Филиал имени Бродского пр. Метафорический, 2 Средний

```
COPY library.book (book_id, title, volume_number, author, field_of_knowledge) FROM
stdin;
```

1	Война и мир	1	Лев Толстой	Литература
2	Война и мир	2	Лев Толстой	Литература
3	Преступление и наказание	1	Федор Достоевский	Литература
4	Властелин колец	1	Дж. Р. Р. Толкиен	Фантазия
5	Властелин колец	2	Дж. Р. Р. Толкиен	Фантазия
6	Властелин колец	3	Дж. Р. Р. Толкиен	Фантазия
7	Гарри Поттер и философский камень	1	Дж. К. Роулинг	Фантазия
8	Гарри Поттер и Тайная комната	2	Дж. К. Роулинг	Фантазия
9	Гарри Поттер и узник Азкабана	3	Дж. К. Роулинг	Фантазия
10	1984	1	Джордж Оруэлл	Научная фантастика
11	Скотный двор	1	Джордж Оруэлл	Политическая сатира
12	Теория относительности	1	Альберт Эйнштейн	Физика
13	Основные концепции математики	1	Р. Д. Кумар	Математика
14	Путешествие в центр Земли	1	Жюль Верн	Научная фантастика
15	Вокруг света за 80 дней	1	Жюль Верн	Приключения
16	C# для начинающих	1	Джон Смит	Программирование на C#
17	Основы C#	1	Питер Джонсон	Программирование на C#
18	C# и .NET: Подробное руководство	1	Марк Майкл	Программирование на C#
19	Эффективное программирование на C#	1	Стив Мартин	Программирование на C#
20	Изучаем C# через примеры	1	Лилия Петрова	Программирование на C#
21	C# в деталях	1	Александр Смирнов	Программирование на C#
22	C# и ASP.NET: разработка веб-приложений	1	Михаил Семенов	Программирование на C#
23	Проектирование приложений на C#	1	Анна Иванова	Программирование на C#
24	C# и Unity: Создание игр	1	Владимир Козлов	Программирование на C#
25	C# для профессионалов	1	Игорь Николаев	Программирование на C#
26	C# и Entity Framework: Работа с базами данных	1	Ольга Сергеева	Программирование на C#
27	C# и Visual Studio: Советы и трюки	1	Денис Петров	Программирование на C#
28	C# и WPF: Разработка графических интерфейсов	1	Елена Волкова	Программирование на C#
29	C# и Xamarin: Мобильная разработка	1	Сергей Ковалев	Программирование на C#
30	C# и Machine Learning: Введение в Искусственный Интеллект	1	Антон Чернов	Программирование на C#

```
COPY library.card_index (card_index_id, book_id, publisher_id, incoming_id,
publisher_type, translator, publish_year, copies_count, compiler) FROM stdin;
```

1	11	11	11	Электронный	Николай Некрасов	2022	55	Николаев
Николай								
2	12	12	12	Печатный	Анна Ахматова	2021	60	Андреев Андрей
3	13	13	13	Электронный	Марина Цветаева	2020	65	Маринов
Марин								
4	14	14	14	Печатный	Борис Пастернак	2019	70	Борисов
Борис								
5	15	15	15	Электронный	Владимир Набоков	2018	75	
Владимиров Владимир								
6	16	16	16	Печатный	Евгений Евтушенко	2017	80	Евгеньев
Евгений								
7	17	17	17	Электронный	Иосиф Бродский	2016	85	Иосифов
Иосиф								
8	18	18	18	Печатный	Виктор Цой	2015	90	Викторов Виктор
9	19	19	19	Электронный	Сергей Есенин	2014	95	Сергеев Сергей
11	6	6	6	Печатный	Федор Достоевский	2018	35	Ульянов
Ульян								
12	7	7	7	Электронный	Максим Горький	2017	40	Сергеев Сергей
13	8	8	8	Печатный	Алексей Толстой	2016	45	Алексеев
Алексей								
14	9	9	9	Электронный	Иван Тургенев	2015	50	Ильин Илья
15	1	1	1	Печатный	Александр Пушкин	2023	10	Иванов
Иван								
16	2	2	2	Печатный	Михаил Лермонтов	2022	15	Петров
Петр								
17	3	3	3	Электронный	Николай Гоголь	2021	20	Сидоров Сидор
18	4	4	4	Печатный	Антон Чехов	2020	25	Смирнов Смирно
19	5	5	5	Электронный	Лев Толстой	2019	30	Федоров Федор

```
COPY library.copy (copy_id, status, retention_period) FROM stdin;
```

1	выданный	2023-05-01
2	возвращенный	2023-04-18
3	просроченный	2023-03-20
4	выданный	2023-02-15
5	возвращенный	2023-01-10
6	просроченный	2022-12-05
7	выданный	2022-11-01
8	возвращенный	2022-10-20
9	просроченный	2022-09-15
10	выданный	2022-08-10
11	возвращенный	2022-07-05
12	просроченный	2022-06-01
13	выданный	2022-05-20
14	возвращенный	2022-04-15
15	просроченный	2022-03-10
16	выданный	2022-02-05
17	возвращенный	2022-01-01
18	просроченный	2021-12-20
19	выданный	2021-11-15
20	возвращенный	2021-10-10
21	просроченный	2021-09-05
22	выданный	2021-08-01

23	возвращенный	2021-07-20
24	просроченный	2021-06-15
25	выданный	2021-05-10
26	возвращенный	2021-04-05
27	просроченный	2021-03-01
28	выданный	2021-02-20
29	возвращенный	2021-01-15
30	просроченный	2020-12-10
31	выданный	2023-06-01
32	возвращенный	2023-05-01
33	просроченный	2023-04-01
34	выданный	2023-06-02
35	возвращенный	2023-05-02
36	просроченный	2023-04-02
37	выданный	2023-06-03
38	возвращенный	2023-05-03
39	просроченный	2023-04-03
40	выданный	2023-06-04
41	возвращенный	2023-05-04
42	просроченный	2023-04-04
43	выданный	2023-06-05
44	возвращенный	2023-05-05
45	просроченный	2023-04-05
46	выданный	2023-06-06
47	возвращенный	2023-05-06
48	просроченный	2023-04-06
49	выданный	2023-06-07
50	возвращенный	2023-05-07
51	просроченный	2023-04-07
52	выданный	2023-06-08
53	возвращенный	2023-05-08
54	просроченный	2023-04-08
55	выданный	2023-06-09
56	возвращенный	2023-05-09
57	просроченный	2023-04-09
58	выданный	2023-06-10
59	возвращенный	2023-05-10
60	просроченный	2023-04-10

```
COPY library.copy_storage (copy_storage_id, copy_id, storage_id, status, date_start,
date_end) FROM stdin;
```

1	1	1	выданный	2023-05-01	2023-06-01
2	2	2	возвращенный	2023-05-02	2023-06-02
3	3	3	просроченный	2023-05-03	2023-06-03
4	60	4	выданный	2023-05-04	2023-06-04
5	5	5	возвращенный	2023-05-05	2023-06-05
6	6	6	просроченный	2023-05-06	2023-06-06
7	7	7	выданный	2023-05-07	2023-06-07
8	58	8	возвращенный	2023-05-08	2023-06-08
9	9	9	просроченный	2023-05-09	2023-06-09
10	10	10	выданный	2023-05-10	2023-06-10
11	11	1	возвращенный	2023-05-11	2023-06-11
12	12	2	просроченный	2023-05-12	2023-06-12
13	13	3	выданный	2023-05-13	2023-06-13
14	14	4	возвращенный	2023-05-14	2023-06-14

15	15	5	просроченный	2023-05-15	2023-06-15
16	16	6	выданный	2023-05-16	2023-06-16
17	17	7	возвращенный	2023-05-17	2023-06-17
18	18	8	просроченный	2023-05-18	2023-06-18
19	19	9	выданный	2023-05-19	2023-06-19
20	20	10	возвращенный	2023-04-20	2023-05-20
21	21	1	просроченный	2023-04-21	2023-05-21
22	22	2	выданный	2023-04-22	2023-05-22
23	23	3	возвращенный	2023-04-23	2023-05-23
24	24	4	просроченный	2023-04-24	2023-05-24
25	25	5	выданный	2023-04-25	2023-05-25
26	26	6	возвращенный	2023-04-26	2023-05-26
27	27	7	просроченный	2023-04-27	2023-05-27
28	28	8	выданный	2023-04-28	2023-05-28
29	29	9	возвращенный	2023-04-29	2023-05-29
30	30	10	просроченный	2023-04-30	2023-05-30

```
COPY library.employee (employee_id, employee_full_name, affiliate_id, "position", email,
phone_number) FROM stdin;
```

1	Иванов Иван Иванович	1	Библиотекарь	ivanov@mail.com	79000000001
2	Петров Петр Петрович	2	Администратор	petrov@mail.com	79000000002
3	Сидоров Сидр Сидорович	3	Технический персонал	sidorov@mail.com	79000000003
4	Смирнов Александр Иванович	4	Библиотекарь	smirnov@mail.com	79000000004
5	Кузнецов Николай Петрович	5	Библиотекарь	kuznetsov@mail.com	79000000005
6	Попов Дмитрий Сергеевич	6	Администратор	popov@mail.com	79000000006
7	Васильев Василий Иванович	7	Технический персонал	vasiliev@mail.com	79000000007
8	Павлов Павел Павлович	8	Библиотекарь	pavlov@mail.com	79000000008
9	Семенов Семен Семенович	9	Библиотекарь	semenov@mail.com	79000000009
10	Голубев Геннадий Петрович	10	Администратор	golubev@mail.com	79000000010
11	Зайцев Захар Борисович	11	Технический персонал	zaytsev@mail.com	79000000011
12	Киселев Кирилл Владимирович	12	Библиотекарь	kiselev@mail.com	79000000012
13	Ильин Илья Ильич	13	Библиотекарь	ilin@mail.com	79000000013
14	Гусев Геннадий Сергеевич	14	Администратор	gusev@mail.com	79000000014
15	Титов Тимур Тимурович	15	Технический персонал	titov@mail.com	79000000015

```
COPY library.incoming (incoming_id, copy_id, supplier_invoice_id, date) FROM stdin;
```

1	1	1	2023-05-01
2	2	2	2023-05-02
3	3	3	2023-05-03
4	60	4	2023-05-04
5	5	5	2023-05-05
6	6	6	2023-05-06
7	7	7	2023-05-07
8	58	8	2023-05-08
9	9	9	2023-05-09
10	10	10	2023-05-10
11	11	1	2023-05-11
12	12	2	2023-05-12
13	13	3	2023-05-13
14	14	4	2023-05-14
15	15	5	2023-05-15
16	16	6	2023-05-16
17	17	7	2023-05-17
18	18	8	2023-05-18
19	19	9	2023-05-19

```
COPY library.publisher (publisher_id, publisher_name, city) FROM stdin;
```

1	Издательство Альфа	Москва
2	Издательский Дом Бета	Санкт-Петербург
3	Издательство Гамма	Екатеринбург
4	Издательство Дельта	Новосибирск
5	Издательский Клуб Эпсилон	Красноярск
6	Издательство Зета	Владивосток
7	Издательство Иота	Казань
8	Издательский Дом Каппа	Нижний Новгород
9	Издательство Лямбда	Калининград
10	Издательский Клуб Мю	Самара
11	Издательство Нью	Омск
12	Издательский Дом Кси	Челябинск
13	Издательство Омикрон	Уфа
14	Издательский Клуб Пи	Ростов-на-Дону
15	Издательство Ро	Пермь
16	Издательский Дом Сигма	Волгоград
17	Издательство Тау	Ульяновск
18	Издательский Клуб Фи	Тюмень
19	Издательство Хи	Ижевск
20	Издательский Дом Ци	Томск
21	Издательство Чи	Барнаул

```
COPY library.reader (reader_id, passport_details, reader_full_name, email, address,
phone_number) FROM stdin;
1      1234567890      Иванов Иван Иванович ivanov@example.com      Москва, ул. Пушкина,
д.10      987654321
2      9876543210      Петров Петр Петрович petrov@example.com      Санкт-Петербург, пр.
Невский, д.20      123456789
3      2468135790      Сидоров Сидор Сидорович      sidorov@example.com      Екатеринбург,
ул. Ленина, д.30      456789123
4      1357924680      Смирнова Анна Владимировна      smirnova@example.com      Новосибирск, ул.
Гоголя, д.40      789123456
5      3692581470      Иванова Екатерина Игоревна      ivanova@example.com      Казань, ул.
Кирова, д.50      321654987
6      8024675913      Попова Ольга Александровна      popova@example.com      Ростов-на-Дону,
ул. Лермонтова, д.60      654987321
7      9513572864      Сергеев Сергей Сергеевич      sergeev@example.com      Уфа, ул. Мира,
д.70      987321654
8      6745192836      Кузнецова Анастасия Ивановна      kuznetsova@example.com
Волгоград, ул. Победы, д.80      321987654
9      2753918465      Александров Александр Александрович      alexandrov@example.com
Пермь, ул. Комсомольская, д.90      654321987
10     1486239757      Николаева Елена Андреевна      nikolaeva@example.com      Воронеж, ул.
Ленинградская, д.100      987654321
```

```
COPY library.registered_reader (registered_reader_id, reader_id, subscription_id,
date_start, date_end) FROM stdin;
1      1      1      2023-01-01      2023-01-31
2      2      2      2023-02-01      2023-02-28
3      3      3      2023-03-01      2023-03-31
4      4      4      2023-04-01      2023-04-30
5      5      5      2023-01-15      2023-02-14
6      6      6      2023-02-15      2023-03-17
7      7      7      2023-03-15      2023-04-14
8      8      8      2023-04-15      2023-05-15
9      9      9      2023-01-10      2023-02-09
10     10     10     2023-02-10      2023-03-12
11     1      11     2023-03-10      2023-04-09
12     2      12     2023-04-10      2023-05-10
13     3      13     2023-01-05      2023-02-04
14     4      14     2023-02-05      2023-03-07
15     5      15     2023-03-05      2023-04-04
16     6      16     2023-04-05      2023-05-05
17     7      17     2023-01-20      2023-02-19
18     8      18     2023-02-20      2023-03-22
19     9      19     2023-03-20      2023-04-19
20     10     20     2023-04-20      2023-05-20
21     1      21     2023-01-25      2023-02-24
22     2      22     2023-02-25      2023-03-27
23     3      23     2023-03-25      2023-04-24
24     4      24     2023-04-25      2023-05-25
25     7      27     2023-03-30      2023-04-29
26     8      28     2023-04-30      2023-05-30
27     9      29     2023-01-12      2023-02-11
28     10     30     2023-02-12      2023-03-14
```

```
COPY library.registering (registering_id, employee_id, date, copy_id) FROM stdin;
```

1	1	2023-01-12	30
2	7	2023-03-05	45
3	5	2022-12-12	15
4	9	2022-11-20	5
5	2	2023-02-28	50
6	6	2022-09-15	10
7	8	2022-08-07	38
8	3	2023-05-10	9
9	13	2023-04-22	18
10	14	2022-10-01	25
11	11	2022-07-14	48
12	12	2023-04-01	33
13	10	2022-06-25	54
14	4	2022-05-30	6
15	15	2022-07-20	21
16	2	2023-01-05	35
17	3	2022-09-10	60
18	6	2023-03-15	11
19	4	2022-08-08	40
20	1	2023-02-27	51
21	8	2022-10-15	14
22	10	2022-12-07	39
23	12	2023-05-18	8
24	14	2023-04-12	20
25	13	2022-06-12	26
26	15	2022-11-14	49
27	9	2023-03-20	34
28	7	2022-05-05	55
29	5	2022-04-30	7
30	11	2022-07-10	22

```
COPY library.replace_document (replace_document_id, employee_id, date,
document_pro_bono_based, document_replace_lost, document_equal_lost, copy_id) FROM
stdin;
```

1	3	2022-12-20	Протокол №25	Приказ №42	Заявление №10	10
2	5	2023-03-15	Распоряжение №15	Уведомление №20	Протокол №30	25
3	1	2023-01-05	Заявление №3	Протокол №23	Приказ №45	40
4	9	2023-02-18	Уведомление №9	Распоряжение №18	Протокол №52	52
5	12	2022-11-30	Приказ №12	Заявление №31	Уведомление №6	6
6	7	2023-03-19	Документ поддержка №57	Документ замена №23	Документ эквивалент №45	23
7	5	2023-02-20	Документ защита №68	Документ обновление №34	Документ заместитель №52	45
8	13	2023-01-15	Документ обеспечение №12	Документ восстановление №78	Документ аналог №39	12
9	9	2022-11-30	Документ сопровождение №83	Документ реконструкция №47	Документ схожий №28	53
10	3	2022-12-17	Документ обслуживание №46	Документ преобразование №32	Документ идентичный №15	33
11	15	2023-04-07	Документ служба №24	Документ модернизация №51	Документ подобный №58	59
12	1	2023-03-23	Документ сервис №36	Документ обновление №19	Документ равнозначный №22	6
13	11	2022-10-11	Документ обслуживание №74	Документ реорганизация №11	Документ сходный №41	17
14	2	2023-01-30	Документ поддержка №38	Документ замена №56	Документ аналогичный №31	25
15	14	2022-11-15	Документ обеспечение №81	Документ восстановление №14	Документ идентичный №67	35

```
COPY library.storage (storage_id, shelf_number, rack_number, room_number) FROM stdin;
```

```
1      1      1      101
2      1      2      101
3      2      1      101
4      2      2      101
5      3      1      101
6      3      2      101
7      4      1      101
8      4      2      101
9      5      1      101
10     5      2      101
11     1      1      102
12     1      2      102
13     2      1      102
14     2      2      102
15     3      1      102
16     3      2      102
17     4      1      102
18     4      2      102
19     5      1      102
20     5      2      102
21     1      1      103
22     1      2      103
23     2      1      103
24     2      2      103
25     3      1      103
26     3      2      103
27     4      1      103
28     4      2      103
29     5      1      103
30     5      2      103
31     1      1      104
32     1      2      104
33     2      1      104
34     2      2      104
35     3      1      104
36     3      2      104
37     4      1      104
38     4      2      104
39     5      1      104
40     5      2      104
41     1      1      105
42     1      2      105
43     2      1      105
44     2      2      105
45     3      1      105
46     3      2      105
47     4      1      105
48     4      2      105
49     5      1      105
50     5      2      105
51     1      1      106
52     1      2      106
53     2      1      106
54     2      2      106
```

55	3	1	106
56	3	2	106
57	4	1	106
58	4	2	106
59	5	1	106
60	5	2	106
61	1	1	107
62	1	2	107
63	2	1	107
64	2	2	107
65	3	1	107
66	3	2	107
67	4	1	107
68	4	2	107
69	5	1	107
70	5	2	107
71	1	1	108
72	1	2	108
73	2	1	108
74	2	2	108
75	3	1	108
76	3	2	108
77	4	1	108
78	4	2	108
79	5	1	108
80	5	2	108
81	1	1	109
82	1	2	109
83	2	1	109
84	2	2	109
85	3	1	109
86	3	2	109
87	4	1	109
88	4	2	109
89	5	1	109
90	5	2	109

```
COPY library.subscription (subscription_id, employee_id, copy_id, date_subscribed,
date_return, date_real_return, status, penalty) FROM stdin;
```

1	1	1	2023-01-01	2023-01-31	\N	активный	\N
2	2	2	2023-02-01	2023-02-28	\N	неактивный	\N
3	3	3	2023-03-01	2023-03-31	\N	активный	\N
4	4	4	2023-04-01	2023-04-30	\N	активный	\N
5	5	5	2023-05-01	2023-05-31	\N	неактивный	\N
6	6	6	2022-06-01	2022-06-30	\N	активный	\N
7	7	7	2022-07-01	2022-07-31	\N	активный	\N
8	8	8	2022-08-01	2022-08-31	\N	неактивный	\N
9	9	9	2022-09-01	2022-09-30	\N	активный	\N
10	10	10	2022-10-01	2022-10-31	\N	активный	\N
11	11	11	2022-11-01	2022-11-30	\N	неактивный	\N
12	12	12	2022-12-01	2022-12-31	\N	активный	\N
13	13	13	2022-01-01	2024-01-31	\N	активный	\N
14	14	14	2022-02-01	2024-02-28	\N	неактивный	\N
15	15	15	2022-03-01	2024-03-31	\N	активный	\N
16	1	16	2023-01-01	2023-01-31	\N	активный	\N
17	2	17	2023-02-01	2023-02-28	\N	неактивный	\N
18	3	18	2023-03-01	2023-03-31	\N	активный	\N
19	4	19	2023-04-01	2023-04-30	\N	активный	\N
20	5	20	2023-05-01	2023-05-19	\N	неактивный	\N
21	6	21	2023-01-01	2023-01-31	\N	активный	\N
22	7	22	2023-02-01	2023-02-28	\N	активный	\N
23	8	23	2023-03-01	2023-03-31	\N	неактивный	\N
24	9	24	2023-04-01	2023-04-30	\N	активный	\N
25	10	25	2023-05-01	2023-05-19	\N	активный	\N
26	11	26	2023-01-01	2023-01-31	\N	неактивный	\N
27	12	27	2023-02-01	2023-02-28	\N	активный	\N
28	13	28	2023-03-01	2023-03-31	\N	активный	\N
29	14	29	2023-04-01	2023-04-30	\N	неактивный	\N
30	15	30	2023-05-01	2023-05-19	\N	активный	\N

```
COPY library.supplier_invoice (supplier_invoice_id, supplier_name, date) FROM stdin;
```

1	ООО "Книжный мир"	2023-05-10
2	ИП Иванов	2023-05-11
3	АО "Успех"	2023-05-12
4	ИП Петров	2023-05-13
5	ООО "Книжный салон"	2023-05-14
6	АО "Деловая литература"	2023-05-15
7	ИП Сидоров	2023-05-16
8	ООО "Партнеры по книгам"	2023-05-17
9	АО "Прогресс"	2023-05-18
10	ИП Новиков	2023-05-19


```
COPY library.write_off (write_off_id, employee_id, date, full_price, reason,
direction_after_write_off, copy_id) FROM stdin;
```

1	1	2023-05-01	500	Повреждение	Склад	1
2	2	2023-04-01	600	Потеря	Утилизация	2
3	3	2023-03-01	700	Износ	Утилизация	3
4	4	2023-02-01	800	Повреждение	Склад	4
5	5	2023-01-01	900	Потеря	Утилизация	5
6	3	2022-12-20	500	Повреждение	Утилизация	10
7	5	2023-03-15	300	Потеря	Перепродажа	25
8	1	2023-01-05	700	Повреждение	Перепродажа	40
9	9	2023-02-18	1000	Потеря	Утилизация	52
10	12	2022-11-30	600	Повреждение	Перепродажа	6
11	7	2023-04-01	800	Повреждение	Утилизация	15
12	2	2023-03-20	400	Потеря	Перепродажа	30
13	8	2023-01-28	500	Повреждение	Утилизация	45
14	14	2022-12-10	700	Потеря	Перепродажа	57
15	4	2023-02-05	600	Повреждение	Утилизация	9
16	6	2022-11-20	800	Повреждение	Перепродажа	24
17	10	2023-04-16	1000	Потеря	Утилизация	39
18	13	2023-01-02	400	Повреждение	Перепродажа	53
19	7	2023-03-01	500	Повреждение	Утилизация	7
20	11	2022-12-15	900	Потеря	Перепродажа	26
21	5	2023-02-12	600	Повреждение	Утилизация	41
22	15	2023-04-20	700	Повреждение	Перепродажа	55
23	1	2023-03-25	800	Потеря	Утилизация	10
24	9	2023-01-10	900	Повреждение	Перепродажа	25
25	12	2022-12-05	1000	Повреждение	Утилизация	40

5. Обновление значений по умолчанию для первичных ключей

```
SELECT pg_catalog.setval('library.affiliate_affiliate_id_seq', 114, true);
SELECT pg_catalog.setval('library.book_book_id_seq', 30, true);
SELECT pg_catalog.setval('library.copy_copy_id_seq', 60, true);
SELECT pg_catalog.setval('library.copy_storage_copy_storage_id_seq', 30, true);
SELECT pg_catalog.setval('library.employee_employee_id_seq', 15, true);
SELECT pg_catalog.setval('library.incoming_incoming_id_seq', 19, true);
SELECT pg_catalog.setval('library.publisher_publisher_id_seq', 21, true);
SELECT pg_catalog.setval('library.reader_reader_id_seq', 10, true);
SELECT pg_catalog.setval('library.registered_reader_registered_reader_id_seq', 28,
true);
SELECT pg_catalog.setval('library.registering_registering_id_seq', 30, true);
SELECT pg_catalog.setval('library.replace_document_replace_document_id_seq', 15, true);
SELECT pg_catalog.setval('library.storage_storage_id_seq', 90, true);
SELECT pg_catalog.setval('library.subscription_subscription_id_seq', 30, true);
SELECT pg_catalog.setval('library.supplier_invoice_supplier_invoice_id_seq', 10, true);
SELECT pg_catalog.setval('library.write_off_write_off_id_seq', 25, true);
SELECT pg_catalog.setval('library."card_Index_card_Index_id_seq"', 19, true);
```

6. Создание ограничений

```
ALTER TABLE ONLY library.affiliate
    ADD CONSTRAINT affiliate_pkey PRIMARY KEY (affiliate_id);

ALTER TABLE ONLY library.book
    ADD CONSTRAINT book_pkey PRIMARY KEY (book_id);

ALTER TABLE library.card_index
    ADD CONSTRAINT ch_copies_count CHECK ((copies_count >= 0)) NOT VALID;

ALTER TABLE library.incoming
    ADD CONSTRAINT ch_date CHECK ((date <= now())) NOT VALID;

ALTER TABLE library.registering
    ADD CONSTRAINT ch_date CHECK ((date <= now())) NOT VALID;

ALTER TABLE library.replace_document
    ADD CONSTRAINT ch_date CHECK ((date <= now())) NOT VALID;

ALTER TABLE library.supplier_invoice
    ADD CONSTRAINT ch_date CHECK ((date <= now())) NOT VALID;

ALTER TABLE library.write_off
    ADD CONSTRAINT ch_date CHECK ((date <= now())) NOT VALID;

ALTER TABLE library.registered_reader
    ADD CONSTRAINT ch_date_end CHECK ((date_end > date_start)) NOT VALID;

ALTER TABLE library.copy_storage
    ADD CONSTRAINT ch_date_end CHECK ((date_end > date_start)) NOT VALID;

ALTER TABLE library.subscription
    ADD CONSTRAINT ch_date_real_return CHECK (((date_real_return IS NULL) OR
(date_real_return > date_subscribed))) NOT VALID;

ALTER TABLE library.subscription
    ADD CONSTRAINT ch_date_return CHECK ((date_return > date_subscribed)) NOT VALID;

ALTER TABLE library.registered_reader
    ADD CONSTRAINT ch_date_start CHECK ((date_start <= now())) NOT VALID;

ALTER TABLE library.copy_storage
    ADD CONSTRAINT ch_date_start CHECK ((date_start <= now())) NOT VALID;

ALTER TABLE library.subscription
    ADD CONSTRAINT ch_date_subscribed CHECK ((date_subscribed <= now())) NOT VALID;

ALTER TABLE library.write_off
    ADD CONSTRAINT ch_full_price CHECK ((full_price > 0)) NOT VALID;

ALTER TABLE library.reader
    ADD CONSTRAINT ch_phone_number CHECK ((phone_number > 0)) NOT VALID;

ALTER TABLE library.employee
    ADD CONSTRAINT ch_phone_number CHECK ((phone_number > 0)) NOT VALID;
```

```
ALTER TABLE library.card_index
    ADD CONSTRAINT ch_publish_year CHECK ((publish_year > 0)) NOT VALID;

ALTER TABLE library.card_index
    ADD CONSTRAINT ch_publisher_type CHECK (((publisher_type)::text = ANY
((ARRAY['Электронный'::character varying, 'Печатный'::character varying])::text[]))) NOT
VALID;

ALTER TABLE library.storage
    ADD CONSTRAINT ch_rack_number CHECK ((rack_number > 0)) NOT VALID;

ALTER TABLE library.storage
    ADD CONSTRAINT ch_room_number CHECK ((room_number > 0)) NOT VALID;

ALTER TABLE library.storage
    ADD CONSTRAINT ch_shelf_number CHECK ((shelf_number > 0)) NOT VALID;

ALTER TABLE library.copy
    ADD CONSTRAINT ch_status CHECK (((status)::text = ANY ((ARRAY['выданный'::character
varying, 'возвращенный'::character varying, 'просроченный'::character
varying])::text[]))) NOT VALID;

ALTER TABLE library.copy_storage
    ADD CONSTRAINT ch_status CHECK (((status)::text = ANY (ARRAY[(('выданный'::character
varying)::text, ('возвращенный'::character varying)::text, ('просроченный'::character
varying)::text]))) NOT VALID;

ALTER TABLE library.subscription
    ADD CONSTRAINT ch_status CHECK (((status)::text = ANY ((ARRAY['активный'::character
varying, 'неактивный'::character varying, 'истекший'::character varying])::text[]))) NOT
VALID;

ALTER TABLE library.book
    ADD CONSTRAINT ch_volume_number CHECK ((volume_number > 0)) NOT VALID;

ALTER TABLE ONLY library.copy
    ADD CONSTRAINT copy_pkey PRIMARY KEY (copy_id);

ALTER TABLE ONLY library.copy_storage
    ADD CONSTRAINT copy_storage_pkey PRIMARY KEY (copy_storage_id);

ALTER TABLE ONLY library.employee
    ADD CONSTRAINT employee_pkey PRIMARY KEY (employee_id);

ALTER TABLE ONLY library.incoming
    ADD CONSTRAINT incoming_pkey PRIMARY KEY (incoming_id);

ALTER TABLE ONLY library.publisher
    ADD CONSTRAINT publisher_pkey PRIMARY KEY (publisher_id);

ALTER TABLE ONLY library.reader
    ADD CONSTRAINT reader_pkey PRIMARY KEY (reader_id);

ALTER TABLE ONLY library.registered_reader
```

```

    ADD CONSTRAINT registered_reader_pkey PRIMARY KEY (registered_reader_id);

ALTER TABLE ONLY library.registering
    ADD CONSTRAINT registering_pkey PRIMARY KEY (registering_id);

ALTER TABLE ONLY library.replace_document
    ADD CONSTRAINT replace_document_pkey PRIMARY KEY (replace_document_id);

ALTER TABLE ONLY library.storage
    ADD CONSTRAINT storage_pkey PRIMARY KEY (storage_id);

ALTER TABLE ONLY library.subscription
    ADD CONSTRAINT subscription_pkey PRIMARY KEY (subscription_id);

ALTER TABLE ONLY library.supplier_invoice
    ADD CONSTRAINT supplier_invoice_pkey PRIMARY KEY (supplier_invoice_id);

ALTER TABLE ONLY library.book
    ADD CONSTRAINT un_book UNIQUE (title, volume_number);

ALTER TABLE ONLY library.storage
    ADD CONSTRAINT un_storage UNIQUE (shelf_number, rack_number, room_number);

ALTER TABLE ONLY library.write_off
    ADD CONSTRAINT write_off_pkey PRIMARY KEY (write_off_id);

ALTER TABLE ONLY library.card_index
    ADD CONSTRAINT "card_Index_pkey" PRIMARY KEY (card_index_id);

ALTER TABLE ONLY library.employee
    ADD CONSTRAINT affiliate_id_fkey FOREIGN KEY (affiliate_id) REFERENCES
library.affiliate(affiliate_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.card_index
    ADD CONSTRAINT book_id_fkey FOREIGN KEY (book_id) REFERENCES library.book(book_id)
ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.incoming
    ADD CONSTRAINT copy_id_fkey FOREIGN KEY (copy_id) REFERENCES library.copy(copy_id)
ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.copy_storage
    ADD CONSTRAINT copy_id_fkey FOREIGN KEY (copy_id) REFERENCES library.copy(copy_id)
ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.subscription
    ADD CONSTRAINT copy_id_fkey FOREIGN KEY (copy_id) REFERENCES library.copy(copy_id)
ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.write_off
    ADD CONSTRAINT copy_id_fkey FOREIGN KEY (copy_id) REFERENCES library.copy(copy_id)
ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.registering
    ADD CONSTRAINT copy_id_fkey FOREIGN KEY (copy_id) REFERENCES library.copy(copy_id)
ON UPDATE CASCADE NOT VALID;

```

```
ALTER TABLE ONLY library.replace_document
    ADD CONSTRAINT copy_id_fkey FOREIGN KEY (copy_id) REFERENCES library.copy(copy_id)
ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.subscription
    ADD CONSTRAINT employee_id_fkey FOREIGN KEY (employee_id) REFERENCES
library.employee(employee_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.replace_document
    ADD CONSTRAINT employee_id_fkey FOREIGN KEY (employee_id) REFERENCES
library.employee(employee_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.write_off
    ADD CONSTRAINT employee_id_fkey FOREIGN KEY (employee_id) REFERENCES
library.employee(employee_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.registering
    ADD CONSTRAINT employee_id_fkey FOREIGN KEY (employee_id) REFERENCES
library.employee(employee_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.card_index
    ADD CONSTRAINT incoming_id_fkey FOREIGN KEY (incoming_id) REFERENCES
library.incoming(incoming_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.card_index
    ADD CONSTRAINT publisher_id_fkey FOREIGN KEY (publisher_id) REFERENCES
library.publisher(publisher_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.registered_reader
    ADD CONSTRAINT reader_id_fkey FOREIGN KEY (reader_id) REFERENCES
library.reader(reader_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.copy_storage
    ADD CONSTRAINT storage_id_fkey FOREIGN KEY (storage_id) REFERENCES
library.storage(storage_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.registered_reader
    ADD CONSTRAINT subscription_id_fkey FOREIGN KEY (subscription_id) REFERENCES
library.subscription(subscription_id) ON UPDATE CASCADE NOT VALID;

ALTER TABLE ONLY library.incoming
    ADD CONSTRAINT supplier_invoice_id_fkey FOREIGN KEY (supplier_invoice_id) REFERENCES
library.supplier_invoice(supplier_invoice_id) ON UPDATE CASCADE NOT VALID;
```

Выводы: В ходе выполнения данной лабораторной работы была изучена процедура создания таблиц в базе данных PostgreSQL с помощью инструмента pgAdmin.

Были рассмотрены основные типы данных, используемые при создании таблиц, а также различные ограничения и backup tool базы данных. Также была проведена работа по заполнению таблиц рабочими данными. Для этого был использован метод импорта данных из внешних источников. Большинство данных были сгенерированы.

В результате выполнения лабораторной работы было получено практическое представление о том, как создавать таблицы в базе данных PostgreSQL и заполнять их данными. Эти навыки будут полезны при проектировании и реализации баз данных в реальных проектах.