Специальность 09.02.07«Информационные системы и программирование»

Дисциплина «Архитектура аппаратных средств»

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИТМО»

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

«СОЗДАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ POSTGRESQL. ЗАПОЛНЕНИЕ ТАБЛИЦ РАБОЧИМИ ДАННЫМИ»

Проверил:
Говоров А.И.
«__»_____20__г.
Оценка: _____

Выполнил:
студент группы Y2339
Волошин П.С.

Санкт-Петербург
2019/2020

Цель работы: овладеть практическими навыками создания таблиц базы данных PostgreSQL 10 (11), заполнения их рабочими данными, резервного копирования и восстановления БД.

Порядок выполнения работ. Познакомиться с инструкциями по работе с отладчиком, написать программу по полученному заданию и, выполняя ее по командам, заполнить таблицы в отчете. Операции умножения и деления выполнять через сложение и вычитание в виде отдельных процедур. Вызов и возврат из процедуры выполнять через команды переходов. При делении учитывать остаток.

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. Заполнить таблицы БД рабочими данными.
5. Создать резервную копию БД.
6. Восстановить БД на другом ПК.

Dump, содержащий скрипт работы:

```
--

-- NOTE:

--

-- File paths need to be edited. Search for $$PATH$$ and

-- replace it with the path to the directory containing

-- the extracted data files.

--

--

-- PostgreSQL database dump

--

-- Dumped from database version 11.7

-- Dumped by pg_dump version 11.7

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;


DROP DATABASE "Bibl";

--

-- Name: Bibl; Type: DATABASE; Schema: -; Owner: postgres

--


CREATE DATABASE "Bibl" WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE = 'Russian_Russia.1251' LC_CTYPE =
'Russian_Russia.1251';
```

```
ALTER DATABASE "Bibl" OWNER TO postgres;

\connect "Bibl"

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_with_oids = false;

--

-- Name: Biblioteka; Type: TABLE; Schema: public; Owner: postgres

--


CREATE TABLE public."Biblioteka" (

    "Adress_b" "char" NOT NULL,

    date_op date NOT NULL,

    date_cl date NOT NULL,

    "ID_rad_hall" integer NOT NULL,

    "ID_bibl" integer NOT NULL,

    id_b integer NOT NULL

);



ALTER TABLE public."Biblioteka" OWNER TO postgres;


--

-- Name: Bibliotekar; Type: TABLE; Schema: public; Owner: postgres
```

```
--


CREATE TABLE public."Bibliotekar" (
    "FIO_bibl" text NOT NULL,
    schedul text NOT NULL,
    phone_bibl integer NOT NULL,
    passport_num integer NOT NULL,
    date_birth_bibl date NOT NULL,
    education_bibl text,
    "ID_bibliotekar" integer NOT NULL
);



ALTER TABLE public."Bibliotekar" OWNER TO postgres;

--
-- Name: book; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.book (
    "ID_book" integer NOT NULL,
    name text NOT NULL,
    year_edition date NOT NULL,
    authors text NOT NULL,
    number_instance integer NOT NULL,
    date_create date NOT NULL
);



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

```
--
-- Name: book_intance; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.book_intance (
    state text NOT NULL,
    "ID_book" integer NOT NULL,
    "Tabel_number" integer,
    "Bibl_number" integer NOT NULL
);


ALTER TABLE public.book_intance OWNER TO postgres;

--
-- Name: getting_a_book; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.getting_a_book (
    "Bib_number" integer NOT NULL,
    "Number_pasport" integer NOT NULL
);


ALTER TABLE public.getting_a_book OWNER TO postgres;

--
-- Name: read_write; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.read_write (
```

```
    number_pasport integer NOT NULL,
    id_read_hall integer NOT NULL
);


ALTER TABLE public.read_write OWNER TO postgres;

--
-- Name: reader; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.reader (
    number_ticket integer NOT NULL,
    "FIO" text NOT NULL,
    "Adress" text NOT NULL,
    number_phone integer NOT NULL,
    date_birh date NOT NULL,
    "Education" text,
    number_pasport integer NOT NULL
);
ALTER TABLE public.reader OWNER TO postgres;
--
-- Name: reading_hall; Type: TABLE; Schema: public; Owner: postgres
--
CREATE TABLE public.reading_hall (
    date_opening date NOT NULL,
    date_closing date NOT NULL,
    "ID_read_hall" integer NOT NULL
);
ALTER TABLE public.reading_hall OWNER TO postgres;
--
```

```
-- Name: registration; Type: TABLE; Schema: public; Owner: postgres

--

CREATE TABLE public.registration (

    number_pasport integer NOT NULL,

    "ID_Bibliotek" integer NOT NULL,

    date_record date NOT NULL,

    date_statement date NOT NULL

);


ALTER TABLE public.registration OWNER TO postgres;

--

-- Data for Name: Biblioteka; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public."Biblioteka" ("Adress_b", date_op, date_cl, "ID_rad_hall", "ID_bibl", id_b) FROM stdin;

\.

COPY public."Biblioteka" ("Adress_b", date_op, date_cl, "ID_rad_hall", "ID_bibl", id_b) FROM '$$PATH$$/2867.dat';

--

-- Data for Name: Bibliotekar; Type: TABLE DATA; Schema: public; Owner: postgres

--


COPY public."Bibliotekar" ("FIO_bibl", schedul, phone_bibl, passport_num, date_birth_bibl, education_bibl, "ID_bibliotekar") FROM stdin;

\.

COPY public."Bibliotekar" ("FIO_bibl", schedul, phone_bibl, passport_num, date_birth_bibl, education_bibl, "ID_bibliotekar") FROM '$$PATH$$/2868.dat';


--

-- Data for Name: book; Type: TABLE DATA; Schema: public; Owner: postgres

--


COPY public.book ("ID_book", name, year_edition, authors, number_instance, date_create) FROM stdin;

\.

COPY public.book ("ID_book", name, year_edition, authors, number_instance, date_create) FROM '$$PATH$$/2869.dat'
```

```
--

-- Data for Name: book_intance; Type: TABLE DATA; Schema: public; Owner: postgres

--


COPY public.book_intance (state, "ID_book", "Tabel_number", "Bibl_number") FROM stdin;

\.

COPY public.book_intance (state, "ID_book", "Tabel_number", "Bibl_number") FROM '$$PATH$$/2870.dat';


-

-- Data for Name: getting_a_book; Type: TABLE DATA; Schema: public; Owner: postgres

--


COPY public.getting_a_book ("Bib_number", "Number_pasport") FROM stdin;

\.

COPY public.getting_a_book ("Bib_number", "Number_pasport") FROM '$$PATH$$/2871.dat';

--

-- Data for Name: read_write; Type: TABLE DATA; Schema: public; Owner: postgres

--


COPY public.read_write (number_pasport, id_read_hall) FROM stdin;

\.

COPY public.read_write (number_pasport, id_read_hall) FROM '$$PATH$$/2872.dat';


--

-- Data for Name: reader; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.reader (number_ticket, "FIO", "Adress", number_phone, date_birh, "Education", number_pasport) FROM stdin;

\.

COPY public.reader (number_ticket, "FIO", "Adress", number_phone, date_birh, "Education", number_pasport) FROM '$$PATH$$/2873.dat';


--
```

```
-- Data for Name: reading_hall; Type: TABLE DATA; Schema: public; Owner: postgres

--


COPY public.reading_hall (date_opening, date_closing, "ID_read_hall") FROM stdin;

\.

COPY public.reading_hall (date_opening, date_closing, "ID_read_hall") FROM '$$PATH$$/2874.dat';


--

-- Data for Name: registration; Type: TABLE DATA; Schema: public; Owner: postgres

--


COPY public.registration (number_pasport, "ID_Bibliotek", date_record, date_statement) FROM stdin;

\.

COPY public.registration (number_pasport, "ID_Bibliotek", date_record, date_statement) FROM '$$PATH$$/2875.dat';


--

-- Name: Biblioteka Biblioteka_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public."Biblioteka"

    ADD CONSTRAINT "Biblioteka_pkey" PRIMARY KEY (id_b);


--

-- Name: Bibliotekar Bibliotekar_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--

ALTER TABLE ONLY public."Bibliotekar"

    ADD CONSTRAINT "Bibliotekar_pkey" PRIMARY KEY ("ID_bibliotekar");


--

-- Name: book_intance book_intance_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--
```

```
ALTER TABLE ONLY public.book_intance
    ADD CONSTRAINT book_intance_pkey PRIMARY KEY ("Bibl_number");

--
-- Name: book book_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.book
    ADD CONSTRAINT book_pkey PRIMARY KEY ("ID_book");

--
-- Name: getting_a_book getting_a_book_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--
ALTER TABLE ONLY public.getting_a_book
    ADD CONSTRAINT getting_a_book_pkey PRIMARY KEY ("Bib_number", "Number_pasport");

--
-- Name: read_write read_write_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.read_write
    ADD CONSTRAINT read_write_pkey PRIMARY KEY (number_pasport, id_read_hall);

--
-- Name: reader reader_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.reader
    ADD CONSTRAINT reader_pkey PRIMARY KEY (number_pasport);
```

```
--

-- Name: reading_hall reading_hall_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public.reading_hall

    ADD CONSTRAINT reading_hall_pkey PRIMARY KEY ("ID_read_hall");


--

-- Name: registration registration_pkey; Type: CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public.registration

    ADD CONSTRAINT registration_pkey PRIMARY KEY (number_pasport, "ID_Bibliotek");


--

-- Name: getting_a_book Bibl_number; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public.getting_a_book

    ADD CONSTRAINT "Bibl_number" FOREIGN KEY ("Bib_number") REFERENCES public.book_intance("Bibl_number");


--

-- Name: registration ID_Biblioteki; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public.registration

    ADD CONSTRAINT "ID_Biblioteki" FOREIGN KEY ("ID_Bibliotek") REFERENCES public."Biblioteka"(id_b);


--

-- Name: Biblioteka ID_bibl; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--
```

```
ALTER TABLE ONLY public."Biblioteka"
    ADD CONSTRAINT "ID_bibl" FOREIGN KEY ("ID_bibl") REFERENCES public."Bibliotekar"("ID_bibliotekar");

--
-- Name: book_intance ID_book; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.book_intance
    ADD CONSTRAINT "ID_book" FOREIGN KEY ("ID_book") REFERENCES public.book("ID_book");

--
-- Name: read_write ID_read_hall; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.read_write
    ADD CONSTRAINT "ID_read_hall" FOREIGN KEY (id_read_hall) REFERENCES public.reading_hall("ID_read_hall");

--
-- Name: Biblioteka ID_read_hall; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Biblioteka"
    ADD CONSTRAINT "ID_read_hall" FOREIGN KEY ("ID_rad_hall") REFERENCES public.reading_hall("ID_read_hall");

--
-- Name: getting_a_book number_pasport; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public.getting_a_book
    ADD CONSTRAINT number_pasport FOREIGN KEY ("Number_pasport") REFERENCES public.reader(number_pasport);
```

```
--

-- Name: read_write number_pasport; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public.read_write

    ADD CONSTRAINT number_pasport FOREIGN KEY (number_pasport) REFERENCES public.reader(number_pasport);


--

-- Name: registration number_passport; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--


ALTER TABLE ONLY public.registration

    ADD CONSTRAINT number_passport FOREIGN KEY (number_pasport) REFERENCES public.reader(number_pasport);


--

-- PostgreSQL database dump complete

--
```

Вывод: в ходе работы были приобретены умения создания таблиц базы данных PostgreSQL 10 (11), заполнения их рабочими данными, резервного копирования и восстановления БД.