

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ»
(УНИВЕРСИТЕТ ИТМО)
Факультет среднего профессионального образования

ОТЧЁТ
О ПРАКТИЧЕСКОЙ РАБОТЕ № 6

по теме: Анализ данных. Создание таблиц базы данных PostgreSQL. Заполнение таблиц рабочими данными.

по дисциплине: Основы проектирования баз данных

Специальность 09.02.07 Информационные системы и программирование

Проверил:

_____ Говоров А.И.

Дата: « ____ » _____ 2020г.

Оценка _____

Выполнил:

студент группы Y2336

_____ Морозов В.А.

ПОСТАНОВКА ЗАДАЧИ

Цель лабораторной работы №6: овладеть практическими навыками создания таблиц базы данных PostgreSQL 12, заполнения их рабочими данными, резервного копирования и восстановления баз данных.

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 рабочими данными.

5. Создать резервную копию базы данных.

6. Восстановить базу данных на другом ПК.

ВЫПОЛНЕНИЕ

Dump, содержащий скрипты работы БД, представлен ниже:

```
CREATE DATABASE "sminc" WITH TEMPLATE = template0 ENCODING =  
'UTF8' LC_COLLATE = 'Russian_Russia.1251' LC_CTYPE = 'Russian_Rus-  
sia.1251';
```

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

```
\connect "sminc"
```

```
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."Classroom"  
(  
  id integer NOT NULL,  
  "number" integer NOT NULL,  
  teacher_id integer NOT NULL,  
  CONSTRAINT "Classroom_pkey" PRIMARY KEY (id),  
  CONSTRAINT teacher_id_ FOREIGN KEY (teacher_id)  
  REFERENCES public."Teacher" (id) MATCH SIMPLE  
  ON UPDATE CASCADE  
  ON DELETE SET NULL  
  NOT VALID  
)
```

```
CREATE TABLE public."Group"  
(  
  id integer NOT NULL,
```

```
"number" text COLLATE pg_catalog."default" NOT NULL,  
CONSTRAINT un_id PRIMARY KEY (id)  
)
```

```
CREATE TABLE public."Lesson"  
(  
id integer NOT NULL,  
group_id integer NOT NULL,  
teacher_id integer NOT NULL,  
subject_id integer NOT NULL,  
lesson n_lesson NOT NULL,  
day weekday,  
CONSTRAINT un_lesson_id PRIMARY KEY (id),  
CONSTRAINT fk_lesson_group_id FOREIGN KEY (group_id)  
REFERENCES public."Group" (id) MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE,  
CONSTRAINT fk_lesson_subject_id FOREIGN KEY (subject_id)  
REFERENCES public."Subject" (id) MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE,  
CONSTRAINT fk_lesson_teacher_id FOREIGN KEY (teacher_id)  
REFERENCES public."Teacher" (id) MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE  
)
```

```
CREATE TABLE public."Student"  
(  
id integer NOT NULL,  
name text COLLATE pg_catalog."default" NOT NULL,  
surname text COLLATE pg_catalog."default" NOT NULL,  
group_id integer NOT NULL,  
CONSTRAINT un_student PRIMARY KEY (id),  
CONSTRAINT fk_student FOREIGN KEY (group_id)  
REFERENCES public."Group" (id) MATCH SIMPLE  
ON UPDATE CASCADE  
ON DELETE CASCADE  
)
```

```
CREATE TABLE public."Subject"  
(  
name text COLLATE pg_catalog."default" NOT NULL,  
id integer NOT NULL,  
CONSTRAINT "Subject_pkey" PRIMARY KEY (id),
```

```
CONSTRAINT un_subject UNIQUE (id)
)
```

```
CREATE TABLE public."Teacher"
(
  id integer NOT NULL,
  name character varying COLLATE pg_catalog."default" NOT NULL,
  surname character varying COLLATE pg_catalog."default" NOT NULL,
  CONSTRAINT un_teacher PRIMARY KEY (id)
)
```

```
CREATE TABLE public."TeachersGroups"
(
  id integer NOT NULL,
  teacher_id integer NOT NULL,
  group_id integer NOT NULL,
  CONSTRAINT un_teachers_groups_id PRIMARY KEY (id),
  CONSTRAINT fk_teachers_groups_group_id FOREIGN KEY (group_id)
  REFERENCES public."Group" (id) MATCH SIMPLE
  ON UPDATE CASCADE
  ON DELETE CASCADE,
  CONSTRAINT fk_teachers_groups_teacher_id FOREIGN KEY (teacher_id)
  REFERENCES public."Teacher" (id) MATCH SIMPLE
  ON UPDATE CASCADE
  ON DELETE CASCADE
)
```

```
CREATE TABLE public."TeachersSubjects"
(
  teacher_id integer NOT NULL,
  id integer NOT NULL,
  subject_id integer NOT NULL,
  CONSTRAINT un_teachers_subjects_id PRIMARY KEY (id),
  CONSTRAINT fk_subject FOREIGN KEY (subject_id)
  REFERENCES public."Subject" (id) MATCH SIMPLE
  ON UPDATE CASCADE
  ON DELETE CASCADE
  NOT VALID,
  CONSTRAINT fk_teacher FOREIGN KEY (teacher_id)
  REFERENCES public."Teacher" (id) MATCH SIMPLE
  ON UPDATE CASCADE
  ON DELETE CASCADE
)
```

```
CREATE TABLE public."Grade"
```

```
(
id integer NOT NULL,
subject_id integer NOT NULL,
student_id integer NOT NULL,
grade integer NOT NULL,
date date NOT NULL,
CONSTRAINT un_grade PRIMARY KEY (id),
CONSTRAINT fk_grade_student FOREIGN KEY (student_id)
REFERENCES public."Student" (id) MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE,
CONSTRAINT fk_grade_subject FOREIGN KEY (subject_id)
REFERENCES public."Subject" (id) MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
)
```

```
INSERT INTO public."Classroom"(
id, "number", teacher_id)
VALUES (1, 101, 1);
INSERT INTO public."Classroom"(
id, "number", teacher_id)
VALUES (2, 102, 2);
INSERT INTO public."Classroom"(
id, "number", teacher_id)
VALUES (3, 103, 3);
INSERT INTO public."Classroom"(
id, "number", teacher_id)
VALUES (4, 104, 4);
```

```
INSERT INTO public."Grade"(
id, subject_id, student_id, grade, date)
VALUES (1, 1, 1, 5, '07-02-2020');
INSERT INTO public."Grade"(
id, subject_id, student_id, grade, date)
VALUES (2, 2, 2, 4, '06-02-2020');
INSERT INTO public."Grade"(
id, subject_id, student_id, grade, date)
VALUES (3, 3, 3, 4, '01-02-2020');
INSERT INTO public."Grade"(
id, subject_id, student_id, grade, date)
VALUES (4, 4, 4, 3, '09-02-2020');
INSERT INTO public."Grade"(
id, subject_id, student_id, grade, date)
VALUES (5, 1, 1, 4, '12-02-2020');
```

```

INSERT INTO public."Group"(
id, "number")
VALUES (1, 'Y2121');
INSERT INTO public."Group"(
id, "number")
VALUES (2, 'Y2122');
INSERT INTO public."Group"(
id, "number")
VALUES (3, 'Y2123');
INSERT INTO public."Group"(
id, "number")
VALUES (4, 'Y2124');

```

```

INSERT INTO public."Lesson"(
id, group_id, teacher_id, subject_id, lesson, day)
VALUES (2, 2, 2, 2, 'II', 'Monday');
INSERT INTO public."Lesson"(
id, group_id, teacher_id, subject_id, lesson, day)
VALUES (3, 3, 3, 3, 'III', 'Wednesday');
INSERT INTO public."Lesson"(
id, group_id, teacher_id, subject_id, lesson, day)
VALUES (4, 4, 4, 4, 'IV', 'Friday');
INSERT INTO public."Lesson"(
id, group_id, teacher_id, subject_id, lesson, day)
VALUES (5, 1, 1, 1, 'I', 'Saturday');
INSERT INTO public."Lesson"(
id, group_id, teacher_id, subject_id, lesson, day)
VALUES (6, 1, 1, 1, 'II', 'Monday');

```

```

INSERT INTO public."Student"(
id, name, surname, group_id)
VALUES (1, 'Ivan', 'Ivanov', 1);
INSERT INTO public."Student"(
id, name, surname, group_id)
VALUES (2, 'Venay', 'Shinobi', 1);
INSERT INTO public."Student"(
id, name, surname, group_id)
VALUES (3, 'Tolik', 'Geniaus', 1);
INSERT INTO public."Student"(
id, name, surname, group_id)
VALUES (4, 'Ann', 'Vetrow', 1);

```

```

INSERT INTO public."Subject"(
name, id)

```



```
VALUES ('Math', 1);
INSERT INTO public."Subject"(
name, id)
VALUES ('History', 2);
INSERT INTO public."Subject"(
name, id)
VALUES ('English', 3);
INSERT INTO public."Subject"(
name, id)
VALUES ('Programming', 4);
```

```
INSERT INTO public."Teacher"(
id, name, surname)
VALUES (1, 'Inna', 'Sverdlavo');
INSERT INTO public."Teacher"(
id, name, surname)
VALUES (2, 'Andrey', 'Luggage');
INSERT INTO public."Teacher"(
id, name, surname)
VALUES (3, 'Alex', 'McLaggen');
```

```
INSERT INTO public."TeachersGroups"(
id, teacher_id, group_id)
VALUES (1, 1, 1);
INSERT INTO public."TeachersGroups"(
id, teacher_id, group_id)
VALUES (2, 1, 2);
INSERT INTO public."TeachersGroups"(
id, teacher_id, group_id)
VALUES (3, 1, 3);
INSERT INTO public."TeachersGroups"(
id, teacher_id, group_id)
VALUES (4, 2, 1);
INSERT INTO public."TeachersGroups"(
id, teacher_id, group_id)
VALUES (5, 2, 2);
INSERT INTO public."TeachersGroups"(
id, teacher_id, group_id)
VALUES (6, 3, 1);
```

```
INSERT INTO public."TeachersSubjects"(
teacher_id, id, subject_id)
VALUES (1, 1, 1);
INSERT INTO public."TeachersSubjects"(
teacher_id, id, subject_id)
```

```
VALUES (2, 2, 1);
INSERT INTO public."TeachersSubjects"(
teacher_id, id, subject_id)
VALUES (3, 3, 2);
INSERT INTO public."TeachersSubjects"(
teacher_id, id, subject_id)
VALUES (4, 4, 2);
INSERT INTO public."TeachersSubjects"(
teacher_id, id, subject_id)
VALUES (1, 5, 3);
INSERT INTO public."TeachersSubjects"(
teacher_id, id, subject_id)
VALUES (2, 6, 3);
INSERT INTO public."TeachersSubjects"(
teacher_id, id, subject_id)
VALUES (3, 7, 4);
```

ВЫВОД

В практической работе №6 были получены практические навыки создания таблиц базы данных PostgreSQL 12, заполнения их рабочими данными, резервного копирования и восстановления баз данных.