

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.2
тема: Создание таблиц базы данных POSTGRESQL.
Заполнение таблиц рабочими данными
дисциплина: «ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗ
ДАННЫХ»

Выполнила:
студентка II курса ИКТ группы К3241
Лорс Хава Абуевна

Проверила:
Говорова Марина Михайловна

Санкт-Петербург
2022

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.
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;
- при создании резервных копий БД настроить параметры *Dump options* для *Type of objects* и *Queries*.

7. Восстановить БД.

Вариант 16. БД "Спортивный клуб"

Описание предметной области: БД должна осуществлять ведение списков спортсменов и тренеров. Тренеры разделены по категориям. При достижении спортсменами определенного рейтинга категория тренера повышается; учет проводимых соревнований (с ведением их архива); учет травм, полученных спортсменами. Предусмотреть: возможность перехода спортсмена от одного тренера к другому; составление рейтингов спортсменов; составление рейтингов тренеров; выдачу информации по соревнованиям; выдачу информации по конкретному спортсмену; подбор возможных кандидатур на участие в соревнованиях (соответствующего уровня мастерства, возраста и без травм).

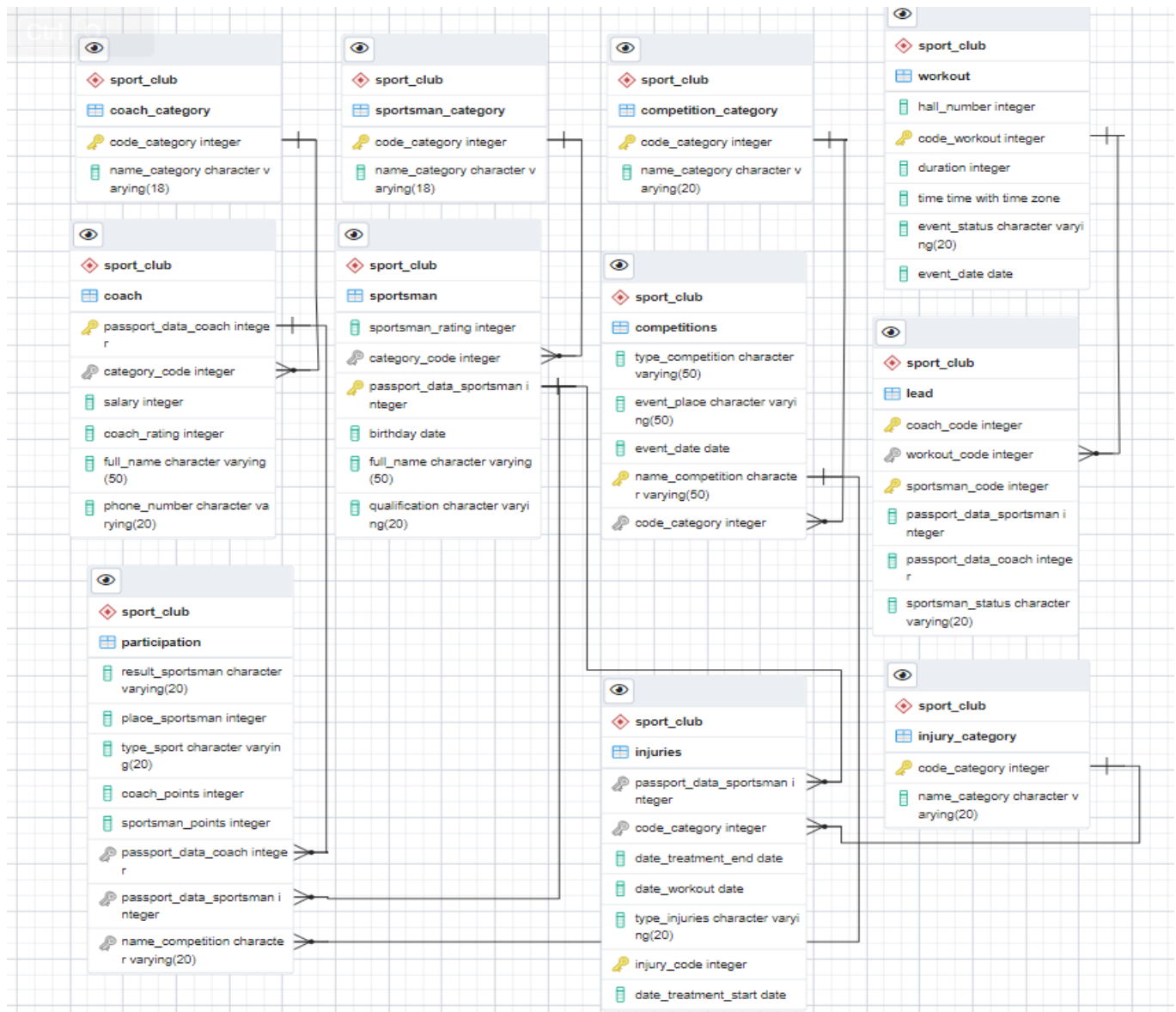
БД должна содержать следующий минимальный набор сведений: ФИО тренера. № телефона тренера. Паспортные данные тренера. Категория тренера. Оклад тренера. Название соревнования. Вид соревнования. Категория соревнования. Место проведения соревнования. Дата проведения соревнования. Фамилия спортсмена. Имя спортсмена. Отчество спортсмена. Дата рождения спортсмена. Паспортные данные тренера. Квалификация спортсмена. Результат спортсмена. Место, которое занял спортсмен. Количество баллов спортсмены за место. Количество баллов тренера за место. Категория спортсмена. Рейтинг спортсмена. Код травмы. Вид

травмы. Тяжесть травмы. Дата получения травмы. Дата окончания лечения. Даты тренировки (перед соревнованием).

Выполнение:

1) Наименование БД: sport_club

2) Схема логической модели базы данных, сгенерированная в Generate ERD:



3) dump, содержащий скрипты работы с БД:

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

-- Dumped from database version 12.10

-- Dumped by pg_dump version 12.10

-- Started on 2022-03-03 23:57:31

// Создание базы данных:

```
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;
```

--

-- TOC entry 6 (class 2615 OID 16413)

-- Name: sport_club; Type: SCHEMA; Schema: -; Owner: postgres

--

// Создание схемы:

```
CREATE SCHEMA sport_club;
```

```
ALTER SCHEMA sport_club OWNER TO postgres;
```

```
SET default_tablespace = '';
```

```
SET default_table_access_method = heap;
```

--

-- TOC entry 203 (class 1259 OID 16424)

-- Name: coach; Type: TABLE; Schema: sport_club; Owner: postgres

--

// Создание таблицы coach:

```
CREATE TABLE sport_club.coach (
    passport_data_coach integer NOT NULL,
    category_code integer NOT NULL,
    salary integer NOT NULL,
    coach_rating integer NOT NULL,
    full_name character varying(50) NOT NULL,
    phone_number character varying(20) NOT NULL
);
```

```
ALTER TABLE sport_club.coach OWNER TO postgres;
```

```
--  
-- TOC entry 211 (class 1259 OID 16521)  
-- Name: coach_category; Type: TABLE; Schema: sport_club; Owner: postgres  
--
```

```
// Создание таблицы coach_category:
```

```
CREATE TABLE sport_club.coach_category (  
    code_category integer NOT NULL,  
    name_category character varying(18) NOT NULL  
);
```

```
ALTER TABLE sport_club.coach_category OWNER TO postgres;
```

```
--  
-- TOC entry 213 (class 1259 OID 16531)  
-- Name: competition_category; Type: TABLE; Schema: sport_club; Owner:  
postgres  
--
```

```
// Создание таблицы competition_category:
```

```
CREATE TABLE sport_club.competition_category (  
    code_category integer NOT NULL,  
    name_category character varying(20) NOT NULL  
);
```

```
ALTER TABLE sport_club.competition_category OWNER TO postgres;
```

```
--  
-- TOC entry 205 (class 1259 OID 16438)  
-- Name: competitions; Type: TABLE; Schema: sport_club; Owner: postgres  
--
```

```
// Создание таблицы competitions
```

```
CREATE TABLE sport_club.competitions (  
    type_competition character varying(50) NOT NULL,  
    event_place character varying(50) NOT NULL,  
    event_date date NOT NULL,  
    name_competition character varying(50) NOT NULL,  
    code_category integer NOT NULL  
);
```

```
ALTER TABLE sport_club.competitions OWNER TO postgres;
```

```
--  
-- TOC entry 207 (class 1259 OID 16463)  
-- Name: injuries; Type: TABLE; Schema: sport_club; Owner: postgres  
--
```

```
// Создание таблицы injuries:
```

```
CREATE TABLE sport_club.injuries (  
    passport_data_sportsman integer NOT NULL,  
    code_category integer NOT NULL,  
    date_treatment_end date NOT NULL,  
    date_workout date NOT NULL,  
    type_injuries character varying(20) NOT NULL,  
    injury_code integer NOT NULL,  
    date_treatment_start date NOT NULL  
);
```

```
ALTER TABLE sport_club.injuries OWNER TO postgres;
```

```
--  
-- TOC entry 212 (class 1259 OID 16526)  
-- Name: injury_category; Type: TABLE; Schema: sport_club; Owner: postgres  
--
```

```
// Создание таблицы injuries_category:
```

```
CREATE TABLE sport_club.injury_category (  
    code_category integer NOT NULL,  
    name_category character varying(20) NOT NULL  
);
```

```
ALTER TABLE sport_club.injury_category OWNER TO postgres;
```

```
--  
-- TOC entry 209 (class 1259 OID 16501)  
-- Name: lead; Type: TABLE; Schema: sport_club; Owner: postgres  
--
```

```
// Создание таблицы lead:
```

```
CREATE TABLE sport_club.lead (  
    coach_code integer NOT NULL,  
    workout_code integer NOT NULL,  
    sportsman_code integer NOT NULL,  
    passport_data_sportsman integer NOT NULL,
```

```
passport_data_coach integer NOT NULL,  
sportsman_status character varying(20) NOT NULL  
);
```

```
ALTER TABLE sport_club.lead OWNER TO postgres;
```

```
--  
-- TOC entry 208 (class 1259 OID 16478)  
-- Name: participation; Type: TABLE; Schema: sport_club; Owner: postgres  
--
```

```
// Создание таблицы participation:
```

```
CREATE TABLE sport_club.participation (  
    result_sportsman character varying(20) NOT NULL,  
    place_sportsman integer NOT NULL,  
    type_sport character varying(20) NOT NULL,  
    coach_points integer NOT NULL,  
    sportsman_points integer NOT NULL,  
    passport_data_coach integer NOT NULL,  
    passport_data_sportsman integer NOT NULL,  
    name_competition character varying(20) NOT NULL  
);
```

```
ALTER TABLE sport_club.participation OWNER TO postgres;
```

```
--  
-- TOC entry 204 (class 1259 OID 16433)  
-- Name: sportsman; Type: TABLE; Schema: sport_club; Owner: postgres  
--
```

```
// Создание таблицы sportsman:
```

```
CREATE TABLE sport_club.sportsman (  
    sportsman_rating integer NOT NULL,  
    category_code integer NOT NULL,  
    passport_data_sportsman integer NOT NULL,  
    birthday date NOT NULL,  
    full_name character varying(50) NOT NULL,  
    qualification character varying(20) NOT NULL  
);
```

```
ALTER TABLE sport_club.sportsman OWNER TO postgres;
```

```
--  
-- TOC entry 210 (class 1259 OID 16513)
```

```
-- Name: sportsman_category; Type: TABLE; Schema: sport_club; Owner:
postgres
--
```

```
// Создание таблицы sportsman_category:
```

```
CREATE TABLE sport_club.sportsman_category (
    code_category integer NOT NULL,
    name_category character varying(18) NOT NULL
);
```

```
ALTER TABLE sport_club.sportsman_category OWNER TO postgres;
```

```
--
-- TOC entry 206 (class 1259 OID 16443)
-- Name: workout; Type: TABLE; Schema: sport_club; Owner: postgres
--
```

```
// Создание таблицы workout:
```

```
CREATE TABLE sport_club.workout (
    hall_number integer NOT NULL,
    code_workout integer NOT NULL,
    duration integer NOT NULL,
    "time" time with time zone NOT NULL,
    event_status character varying(20) NOT NULL,
    event_date date
);
```

```
ALTER TABLE sport_club.workout OWNER TO postgres;
```

```
// Заполнение таблицы coach рабочими данными:
```

```
COPY sport_club.coach (passport_data_coach, category_code, salary,
coach_rating, full_name, phone_number) FROM stdin;
1510082405 1 75000 1 ПОПОВ ДМИТРИЙ 936789595
1610082405 2 75000 1 ДУРОВ АНТОН 936787412
1101574677 3 75000 1 ЕРМОЛОВА ВАСИЛИСА 936581855\n
\.
```

```
// Заполнение таблицы coach_category рабочими данными:
```

```
COPY sport_club.coach_category (code_category, name_category) FROM stdin;
```



```
1    ПЛАВАНИЕ
2    ВОЛЕЙБОЛ
3    ФИГУРНОЕ КАТАНИЕ
\.
```

// Заполнение таблицы competition_category рабочими данными:

```
COPY sport_club.competition_category (code_category, name_category) FROM
stdin;
```

```
1    СПОРТИВНОЕ ПЛАВАНИЕ
2    ВОЛЕЙБОЛ
3    ФИГУРНОЕ КАТАНИЕ
\.
```

// Заполнение таблицы competitions рабочими данными:

```
COPY sport_club.competitions (type_competition, event_place, event_date,
name_competition, code_category) FROM stdin;
```

```
ОДИНОЧНОЕ      СК ГОРЬКОВСКАЯ 2022-03-24    ФИГУРНОЕ
КАТАНИЕ ПО СТРАНАМ 3
КОМАНДНОЕ      СК ВЯЗЕМСКИЙ   2022-03-25    МЕЖДУНАРОДНЫЕ
СОРЕВНОВАНИЯ ПО ВОЛЕЙБОЛУ 2
ОДИНОЧНОЕ      БАССЕЙН СКА    2022-07-26    ЧЕМПИОНАТ МИРА
ПО ПЛАВАНИЮ    1
\.
```

// Заполнение таблицы injuries рабочими данными:

```
COPY sport_club.injuries (passport_data_sportsman, code_category,
date_treatment_end, date_workout, type_injuries, injury_code,
date_treatment_start) FROM stdin;
```

```
1285277485 1    2014-04-22    2014-04-05    ЛЕГКИЙ        101
2014-04-05
1501256372 1    2018-01-17    2018-01-01    ЛЕГКИЙ        102
2018-01-01
1501256372 3    2019-09-09    2019-08-20    ЛЕГКИЙ        301
2019-08-20
\.
```

// Заполнение таблицы injury_category рабочими данными:

```
COPY sport_club.injury_category (code_category, name_category) FROM stdin;
```

```
1      УШИБ
2      ОЖОГ
3      ПЕРЕЛОМ
\.
```

// Заполнение таблицы lead рабочими данными:

```
COPY sport_club.lead (coach_code, workout_code, sportsman_code,
passport_data_sportsman, passport_data_coach, sportsman_status) FROM stdin;
10102203      1200  1060809      1285277485  1101574677
ПРИСУТСТВОВАЛ
10103206      1201  1060910      1501256372  1610082405
ПРИСУТСТВОВАЛ
10104208      1203  1060165      1701574456  1510082405
ПРИСУТСТВОВАЛ
\.
```

// Заполнение таблицы participation рабочими данными:

```
COPY sport_club.participation (result_sportsman, place_sportsman, type_sport,
coach_points, sportsman_points, passport_data_coach, passport_data_sportsman,
name_competition) FROM stdin;
\.
```

// Заполнение таблицы sportsman рабочими данными:

```
COPY sport_club.sportsman (sportsman_rating, category_code,
passport_data_sportsman, birthday, full_name, qualification) FROM stdin;
14      1      1701574456  2000-01-01  СТЕПАНОВ АНДРЕЙ
ПРОФЕССИОНАЛ
7       2      1501256372  1998-08-25  КИРИЛЕНКО ДАНИИЛ
ПРОФЕССИОНАЛ
5       3      1285277485  2001-05-03  ДОБРЮЛЮБОВА АННА
ВЫСОКИЙ КЛАСС
\.
```

// Заполнение таблицы sportsman_category рабочими данными:

```
COPY sport_club.sportsman_category (code_category, name_category) FROM
stdin;
1      ПЛАВАНИЕ
2      ВОЛЕЙБОЛ
```

3 ФИГУРНОЕ КАТАНИЕ

\.

// Заполнение таблицы workout рабочими данными:

```
COPY sport_club.workout (hall_number, code_workout, duration, "time",  
event_status, event_date) FROM stdin;
```

```
501 1200 150 10:00:00+03 ПРОВЕДЕНА 2022-02-15
```

```
243 1201 150 09:00:00+03 НЕ ПРОВЕДЕНА 2022-03-18
```

```
112 1203 150 10:00:00+03 ПРОВЕДЕНА 2022-03-27
```

\.

// Ограничения на значения полей:

```
ALTER TABLE ONLY sport_club.lead  
ADD CONSTRAINT coach_code UNIQUE (coach_code);
```

```
ALTER TABLE ONLY sport_club.coach  
ADD CONSTRAINT coach_pkey PRIMARY KEY (passport_data_coach);
```

```
ALTER TABLE ONLY sport_club.sportsman_category  
ADD CONSTRAINT code_category PRIMARY KEY (code_category);
```

```
ALTER TABLE ONLY sport_club.coach_category  
ADD CONSTRAINT code_category_coach PRIMARY KEY (code_category);
```

```
ALTER TABLE ONLY sport_club.coach_category  
ADD CONSTRAINT code_category_coach_unique UNIQUE (code_category);
```

```
ALTER TABLE ONLY sport_club.competition_category  
ADD CONSTRAINT code_category_competition PRIMARY KEY  
(code_category);
```

```
ALTER TABLE ONLY sport_club.competition_category  
ADD CONSTRAINT code_category_competition_unique UNIQUE  
(code_category);
```

```
ALTER TABLE ONLY sport_club.injury_category  
ADD CONSTRAINT code_category_injury PRIMARY KEY (code_category);
```

```
ALTER TABLE ONLY sport_club.injury_category
  ADD CONSTRAINT code_category_injury_unique UNIQUE (code_category);
```

```
ALTER TABLE ONLY sport_club.sportsman_category
  ADD CONSTRAINT code_category_sportsman_unique UNIQUE
(code_category);
```

```
ALTER TABLE ONLY sport_club.lead
  ADD CONSTRAINT code_lead PRIMARY KEY (coach_code,
sportsman_code);
```

```
ALTER TABLE ONLY sport_club.workout
  ADD CONSTRAINT code_workout_unique UNIQUE (code_workout);
```

```
ALTER TABLE ONLY sport_club.injuries
  ADD CONSTRAINT injury PRIMARY KEY (injury_code);
```

```
ALTER TABLE ONLY sport_club.injuries
  ADD CONSTRAINT injury_unique UNIQUE (injury_code);
```

```
ALTER TABLE ONLY sport_club.competitions
  ADD CONSTRAINT name PRIMARY KEY (name_competition);
```

```
ALTER TABLE ONLY sport_club.coach
  ADD CONSTRAINT pasaport_data_unique UNIQUE (passport_data_coach);
```

```
ALTER TABLE sport_club.participation
  ADD CONSTRAINT place CHECK (((place_sportsman >= 1) AND
(place_sportsman <= 3))) NOT VALID;
```

```
ALTER TABLE sport_club.coach
```

```
ADD CONSTRAINT rating CHECK (((coach_rating >= 1) AND (coach_rating
<= 10))) NOT VALID;
```

```
ALTER TABLE sport_club.sportsman
ADD CONSTRAINT rating CHECK (((sportsman_rating >= 1) AND
(sportsman_rating <= 100))) NOT VALID;
```

```
ALTER TABLE ONLY sport_club.lead
ADD CONSTRAINT sportsman_code UNIQUE (sportsman_code);
```

```
ALTER TABLE ONLY sport_club.sportsman
ADD CONSTRAINT sportsman_passport_data_unique UNIQUE
(passport_data_sportsman);
```

```
ALTER TABLE ONLY sport_club.sportsman
ADD CONSTRAINT sportsman_pkey PRIMARY KEY
(passport_data_sportsman);
```

```
ALTER TABLE ONLY sport_club.workout
ADD CONSTRAINT workout_pkey PRIMARY KEY (code_workout);
```

```
ALTER TABLE ONLY sport_club.coach
ADD CONSTRAINT category_code_coach FOREIGN KEY (category_code)
REFERENCES sport_club.coach_category(code_category) NOT VALID;
```

```
ALTER TABLE ONLY sport_club.competitions
ADD CONSTRAINT code_category_competition FOREIGN KEY
(code_category) REFERENCES sport_club.competition_category(code_category)
NOT VALID;
```

```
ALTER TABLE ONLY sport_club.injuries
ADD CONSTRAINT code_category_injuries FOREIGN KEY (code_category)
REFERENCES sport_club.injury_category(code_category) NOT VALID;
```

```
ALTER TABLE ONLY sport_club.sportsman
  ADD CONSTRAINT code_category_sportsman FOREIGN KEY
(category_code) REFERENCES sport_club.sportsman_category(code_category)
NOT VALID;
```

```
ALTER TABLE ONLY sport_club.lead
  ADD CONSTRAINT code_workout FOREIGN KEY (workout_code)
REFERENCES sport_club.workout(code_workout);
```

```
ALTER TABLE ONLY sport_club.participation
  ADD CONSTRAINT name_competition FOREIGN KEY (name_competition)
REFERENCES sport_club.competitions(name_competition) NOT VALID;
```

```
ALTER TABLE ONLY sport_club.participation
  ADD CONSTRAINT passport_data FOREIGN KEY (passport_data_sportsman)
REFERENCES sport_club.sportsman(passport_data_sportsman) NOT VALID;
```

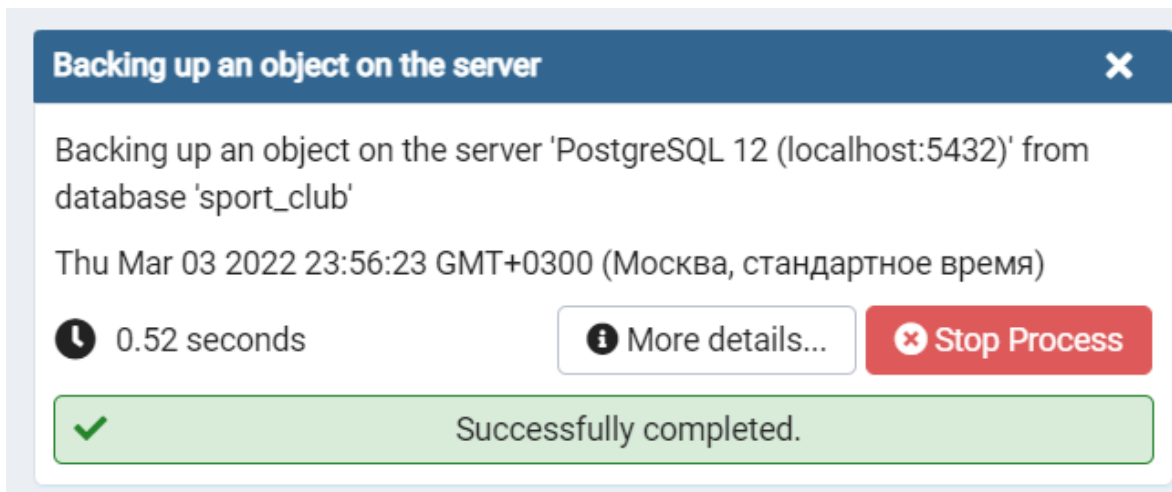
```
ALTER TABLE ONLY sport_club.injuries
  ADD CONSTRAINT passport_data_sportsman_injuries FOREIGN KEY
(passport_data_sportsman) REFERENCES
sport_club.sportsman(passport_data_sportsman) NOT VALID;
```

```
ALTER TABLE ONLY sport_club.participation
  ADD CONSTRAINT passport_date FOREIGN KEY (passport_data_coach)
REFERENCES sport_club.coach(passport_data_coach) NOT VALID;
```

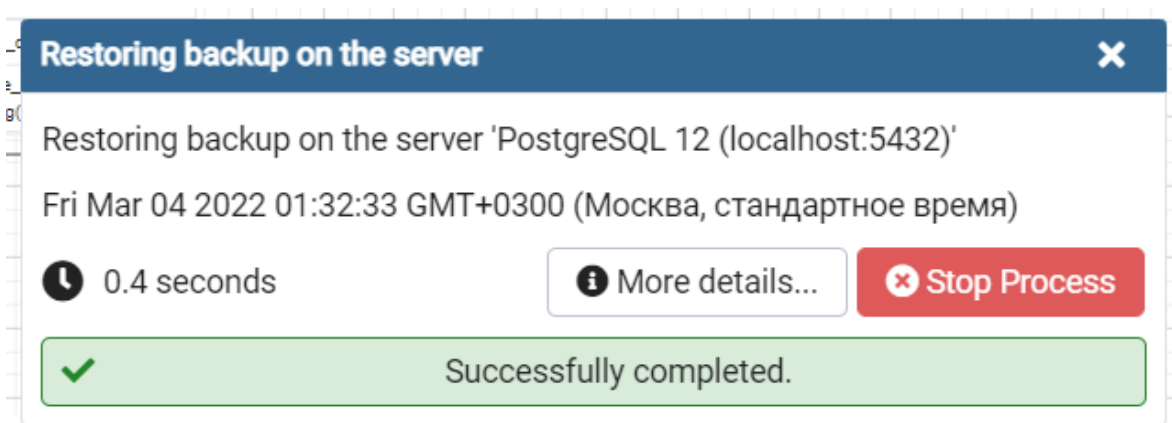
```
-- Completed on 2022-03-03 23:57:31
```

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

4) Резервная копия БД:



5) Восстановление БД:



Вывод:

В ходе выполнения работы были приобретены практические навыки создания базы данных в PostgreSQL 1X, создания схемы в составе БД, создании таблиц баз данных, заполнения их рабочими данными и установления ограничений на данные: Primary Key, Unique, Check, Foreign Key. Также были созданы две резервные копии с определенными расширениями.