

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**ОТЧЕТ**

по проектной работе

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

по дисциплине **«Проектирование и реализация баз данных»**

Автор:

Петросян Анна Мнацакановна

Факультет:

Инфокоммуникационных технологий

Группа:

К3242

Преподаватель: Говорова Марина Михайловна



**УНИВЕРСИТЕТ ИТМО**

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 (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.

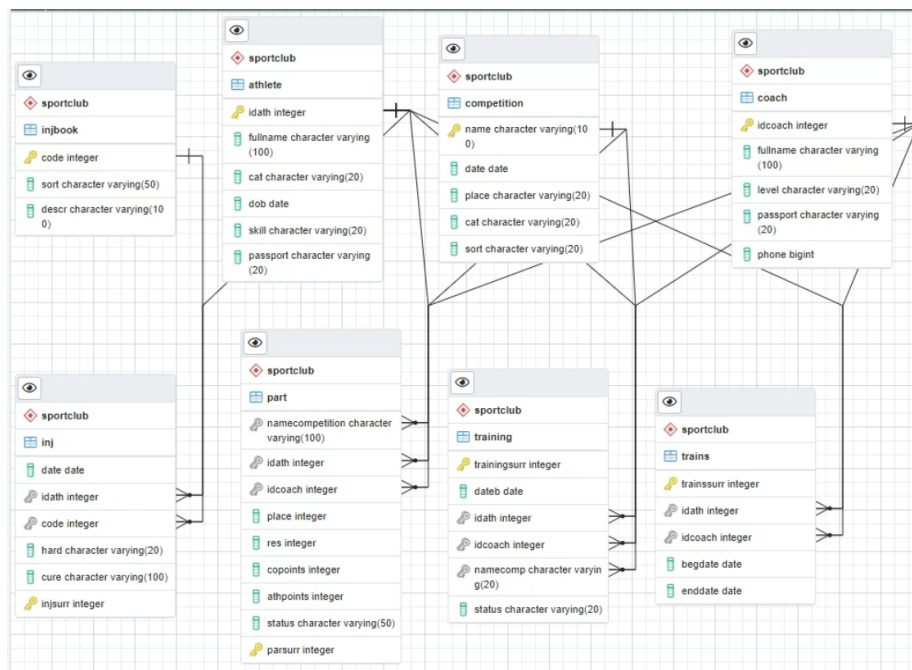
*Указание:*

*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.

**Выполнение:**

- наименование БД - Спортивный клуб
- схема логической модели базы данных, сгенерированная в Generate ERD:



- dump, содержащий скрипты работы с БД:

Создание и заполнение (представлено по одному запросу на заполнение каждой таблицы)

#### Создание базы данных

```
CREATE DATABASE postgres #создание базы данных
WITH
  OWNER = postgres
  ENCODING = 'UTF8' # установление кодировки
  LC_COLLATE = 'Russian_Russia.1251'
  LC_CTYPE = 'Russian_Russia.1251'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1;
```

```
COMMENT ON DATABASE postgres
IS 'default administrative connection database';
```

#### Создание схемы:

```
CREATE SCHEMA IF NOT EXISTS sportclub #создание схемы
AUTHORIZATION postgres; #принадлежность к бд
```

#### Создание таблиц:

```
CREATE TABLE IF NOT EXISTS sportclub.athlete #создание таблицы
(
  idath integer NOT NULL, #целочисленный тип данных, обязательность
  fullname character varying(100) COLLATE pg_catalog."default" NOT
NULL, #тд 100битная строка, об зн
  cat character varying(20) COLLATE pg_catalog."default" NOT NULL,
#тд 20тибитная строка, об зн
  dob date NOT NULL, # дата, обязательность значения
  skill character varying(20) COLLATE pg_catalog."default" NOT
NULL,#тд 20тибитная строка, об зн
  passport character varying(20) COLLATE pg_catalog."default" NOT
NULL,#тд 20тибитная строка, об зн
  CONSTRAINT athlete_pkey PRIMARY KEY (idath), #первичный ключ
  CONSTRAINT idath UNIQUE (idath), #ограничение на уникальное
  значение id-номеров
  CONSTRAINT passport UNIQUE (passport) #ограничение на
  уникальные номера паспортов
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS sportclub.athlete
  OWNER to postgres;
```

```
INSERT INTO sportclub.athlete (
idath, fullname, cat, dob, skill, passport) VALUES('2'::integer,
'Ясносолнышкин Юдзуру'::character varying, 'Вышая'::character varying,
'1994-12-07'::date, 'ЗМС'::character varying, '34JJ98'::character varying)
returning idath
```

```
CREATE TABLE IF NOT EXISTS sportclub.coach #создание таблицы
(
    idcoach integer NOT NULL, #тд целочисленный, обязательный атрибут
    fullname character varying(100) COLLATE pg_catalog."default" NOT NULL, #тд 100битная строка, об зн
    level character varying(20) COLLATE pg_catalog."default" NOT NULL, #тд 20тибитная строка, об зн
    passport character varying(20) COLLATE pg_catalog."default" NOT NULL, #тд 20тибитная строка, об зн
    phone bigint NOT NULL, #тд целочисленный с большим диапазоном, об зн
    CONSTRAINT idcoach PRIMARY KEY (idcoach), #назначение первичного ключа
    CONSTRAINT idcoachun UNIQUE (idcoach), #ограничение на id-номер тренера
    CONSTRAINT passportun UNIQUE (passport) #ограничение на уникальность номера паспорта
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS sportclub.coach
    OWNER to postgres;
INSERT INTO sportclub.coach (
idcoach, fullname, level, passport, phone) VALUES('137829'::integer,
'Весёлая Мизуки Андреевна'::character varying, 'Вышая'::character varying,
'4010897523'::character varying, '89215674030'::bigint)
returning idcoach
```

```
CREATE TABLE IF NOT EXISTS sportclub.competition #создание таблицы
(
    name character varying(100) COLLATE pg_catalog."default" NOT NULL, #100битная строка, об зн
    date date NOT NULL, #тд дата, об зн
    place character varying(20) COLLATE pg_catalog."default" NOT NULL, #20тибитная строка, об зн
    cat character varying(20) COLLATE pg_catalog."default" NOT NULL, #20тибитная строка, об зн
    sort character varying(20) COLLATE pg_catalog."default" NOT NULL, #20тибитная строка, об зн
    CONSTRAINT competition_pkey PRIMARY KEY (name) #назначение первичного ключа
```

)

**TABLESPACE** pg\_default;

**ALTER TABLE IF EXISTS** sportclub.competition

OWNER **to** postgres;

INSERT INTO sportclub.competition (  
name, date, place, cat, sort) VALUES('10-й этап Кубка Креветкина  
2017'::character varying, '2017-03-02'::date, 'Иркутск'::character varying,  
'Взрослые'::character varying, 'Региональное'::character varying)  
returning name

**CREATE TABLE IF NOT EXISTS** sportclub.inj #создание таблицы

(

**date date NOT NULL**, #тд дата, об зн

idath **integer NOT NULL**, #создание столбца для внешнего ключа,  
целочисленный тд, об зн

code **integer NOT NULL**, #создание столбца для внешнего ключа,  
целочисленный тд, об зн

hard **character varying(20) COLLATE** pg\_catalog."default" **NOT NULL**,  
#20тибитная строка, об зн

cure **character varying(100) COLLATE** pg\_catalog."default" **NOT  
NULL**, #100битная строка, об зн

injsurr **integer NOT NULL**, #суррогатный, целочисленный тд, об зн

**CONSTRAINT** inj\_pkey **PRIMARY KEY** (injsurr), #суррогатный ключ  
вместо составного

**CONSTRAINT** code **FOREIGN KEY** (code) #внешний ключ

**REFERENCES** sportclub.injbook (code) **MATCH** SIMPLE

#обращение к родительской таблице

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION**,

**CONSTRAINT** idath **FOREIGN KEY** (idath) #внешний ключ

**REFERENCES** sportclub.athlete (idath) **MATCH** SIMPLE

#обращение к родительской таблице

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION**

)

**TABLESPACE** pg\_default;

**ALTER TABLE IF EXISTS** sportclub.inj

OWNER **to** postgres;

INSERT INTO sportclub.inj (  
date, idath, code, hard, cure, injsurr) VALUES('2019-04-01'::date',  
'123456'::integer, '1'::integer, 'Средняя'::character varying, 'Покой, остеопат'::character  
varying, '1'::integer)  
returning injsurr

```

CREATE TABLE IF NOT EXISTS sportclub.injbook #создание таблицы
(
    code integer NOT NULL, #целочисленный тд, об зн
    sort character varying(50) COLLATE pg_catalog."default" NOT NULL,
#50тибитная строка, об зн
    descr character varying(100) COLLATE pg_catalog."default" NOT
NULL,#стобитная строка, об зн
    CONSTRAINT code PRIMARY KEY (code) #назначение первичного
ключа
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE IF EXISTS sportclub.injbook

```

```

    OWNER to postgres;

```

```

INSERT INTO sportclub.injbook(
    code, sort, descr) VALUES ('1'::integer, 'Разрыв ахиллова
сухожилия'::character varying, 'сопровождается звуком резкого щелчка, острой болью
над пяткой, сильным отеком'::character varying)
returning code

```

```

CREATE TABLE IF NOT EXISTS sportclub.part #создание таблицы

```

```

(
    namecompetition character varying(100) COLLATE pg_catalog."default"
NOT NULL, #100битная строка, об зн
    idath integer NOT NULL, #создается атрибут для внешнего ключа,
целочисленный тд, об зн
    idcoach integer NOT NULL, #создается атрибут для внешнего ключа,
целочисленный тд, об зн
    place integer NOT NULL, #целочисленный тд, об зн
    res integer NOT NULL, #целочисленный тд, об зн
    sorpoints integer NOT NULL, #целочисленный тд, об зн
    athpoints integer NOT NULL, #целочисленный тд, об зн
    status character varying(50) COLLATE pg_catalog."default" NOT
NULL, #50тибитная, об зн
    parsurr integer NOT NULL, #суррогатный ключ вместо составного
CONSTRAINT part_pkey PRIMARY KEY (parsurr), #назначение
суррогатного первичным
CONSTRAINT idath FOREIGN KEY (idath) #для внешнего ключа
REFERENCES sportclub.athlete (idath) MATCH SIMPLE
#обращение к род.табл.
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT idcoach FOREIGN KEY (idcoach) #для внешнего
ключа
REFERENCES sportclub.coach (idcoach) MATCH SIMPLE
#обращение к род.табл.
ON UPDATE NO ACTION

```

```

        ON DELETE NO ACTION,
        CONSTRAINT namecompet FOREIGN KEY (namecompetition) #для
внешнего ключа
        REFERENCES sportclub.competition (name) MATCH SIMPLE
#обращение к род.табл.
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
    )

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE IF EXISTS sportclub.part
OWNER to postgres;
INSERT INTO sportclub.part (
    namecompetition, idath, idcoach, place, res, copoints, sthpoints, status,
    parsuss) VALUES ('Первенство России по каратэ 2019'::character
varying, '234456'::integer, '197295'::integer, '2'::integer,
'6'::integer, '8'::athpoints, '8'::copoints, 'Призёр'::character
varying, '1'::integer)
returning parsurr

```

```

CREATE TABLE IF NOT EXISTS sportclub.training #создание таблицы
(
    trainingsurr integer NOT NULL, #сурпготатный вместо составного,
целочисл, об зн
    dateb date NOT NULL, #дата, об зн
    idath integer NOT NULL, #для внешнего ключа, целочисл, об зн
    idcoach integer NOT NULL, #для внешнего ключа, целочисл, об зн
    namecomp character varying(100) COLLATE pg_catalog."default" NOT
NULL, #для внешнего ключа, 20тибитная строка, об зн
    status character varying(20) COLLATE pg_catalog."default" NOT
NULL, #20битная строка, об зн
    CONSTRAINT trngsurr PRIMARY KEY (trainingsurr), #назначение
суррогатного ключа первичным
    CONSTRAINT idath FOREIGN KEY (idath) #внешний ключ
REFERENCES sportclub.athlete (idath) MATCH SIMPLE #ссылка на
род.табл.
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
    CONSTRAINT idcoach FOREIGN KEY (idcoach) #внешний ключ

```

**REFERENCES** sportclub.coach (idcoach) **MATCH SIMPLE** #ссылка на род.табл.

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION,**

**CONSTRAINT** namecomp **FOREIGN KEY** (namecomp) #внешний ключ

**REFERENCES** sportclub.competition (name) **MATCH SIMPLE** #ссылка на род.табл.

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION**

)

**TABLESPACE** pg\_default;

**ALTER TABLE IF EXISTS** sportclub.training

**OWNER to** postgres;

**INSERT INTO** sportclub.training (

trainingsurr, dateb, status, idath, idcoach, namecomp) **VALUES** (

'1'::integer, '2019-03-03'::date, 'Участник'::character varying,

'123456'::integer, '197295'::integer, 'Контрольный Заплыв 2019'::character

varying)

**returning** trainingsurr;

**CREATE TABLE IF NOT EXISTS** sportclub.trains #создание таблицы

(

trainssurr **integer NOT NULL**, #суррогатный ключ вместо составного, об зн

idath **integer NOT NULL**, #для внешнего, целочис, об зн

idcoach **integer NOT NULL**, #для внешнего, целочис, об зн

begdate **date NOT NULL**, #дата, об зн

enddate **date NOT NULL**, #дата, об зн

**CONSTRAINT** trnsurr **PRIMARY KEY** (trainssurr), #назначение суррогатного первичным

**CONSTRAINT** idath **FOREIGN KEY** (idath) #внешний ключ

**REFERENCES** sportclub.athlete (idath) **MATCH SIMPLE** #обр к род табл

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION,**

**CONSTRAINT** idcoach **FOREIGN KEY** (idcoach) #внешний ключ

**REFERENCES** sportclub.coach (idcoach) **MATCH SIMPLE** #обр к род табл

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION,**

**CONSTRAINT** dates **CHECK** (begdate <= enddate) #проверка на правильность введения дат

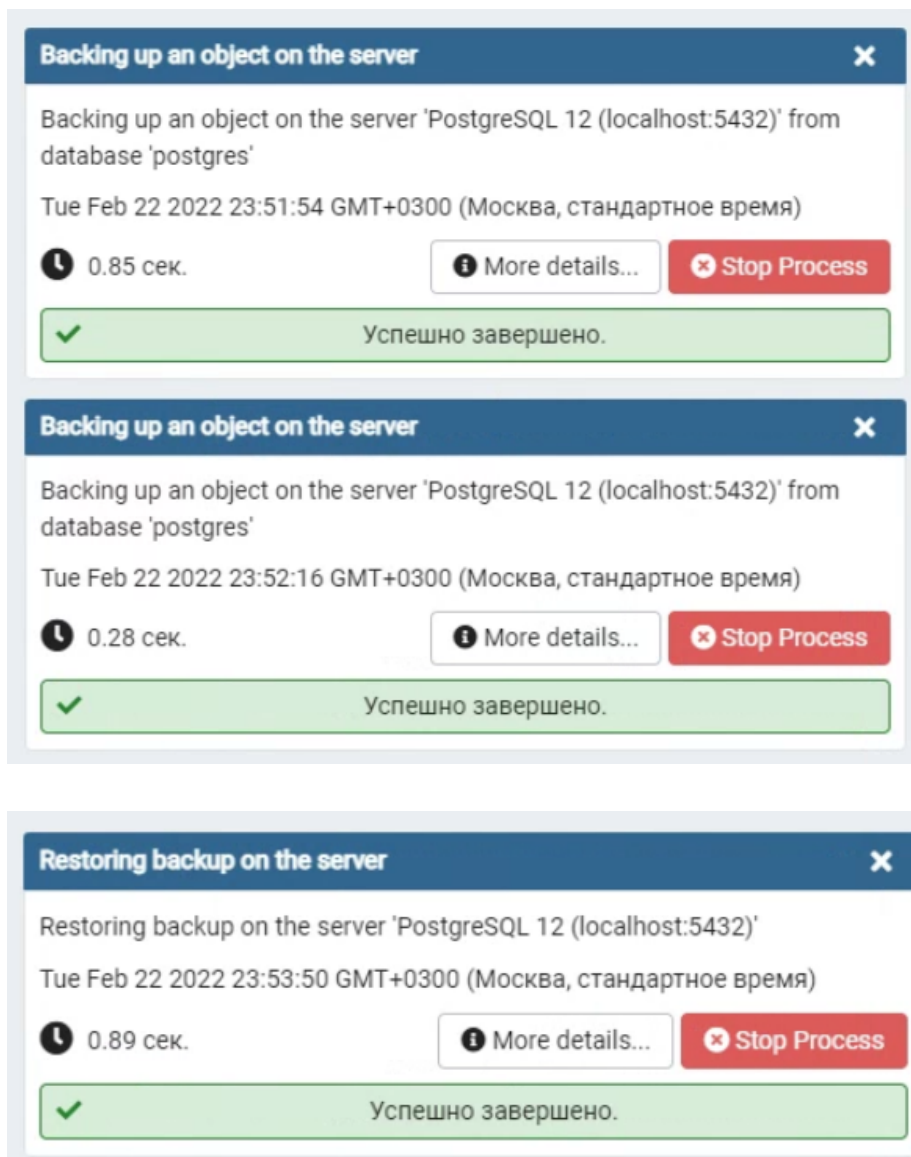
)

**TABLESPACE** pg\_default;



```
ALTER TABLE IF EXISTS sportclub.trains
OWNER to postgres;
INSERT INTO sportclub.trains (
trainssurr, idath, idcoach, begdate, enddate) VALUES (
'1'::integer, '2'::integer, '137829'::integer, '2012-03-01'::date,
'2022-09-06'::date)
returning trainssurr;
```

– Резервная копия и восстановление



**Выводы:** с помощью pgadmin создала базу данных и заполнила её, установила ограничения на таблицы (первичный ключ, проверка значений и уникальности), заменила составные ключи на суррогатные. Построена схема, аналогичная erwin, сделана резервная копия для восстановления и листинга кода.