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 № 3

по теме:

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

по дисциплине: Проектирование и реализация баз данных

Проверила:
Говорова М.М.
Дата:_____2023 г.
Оценка _____

Выполнила:
студент группы К3241
Борисова Э. Е.

Санкт-Петербург 2023/2024

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 16. БД "Спортивный клуб"

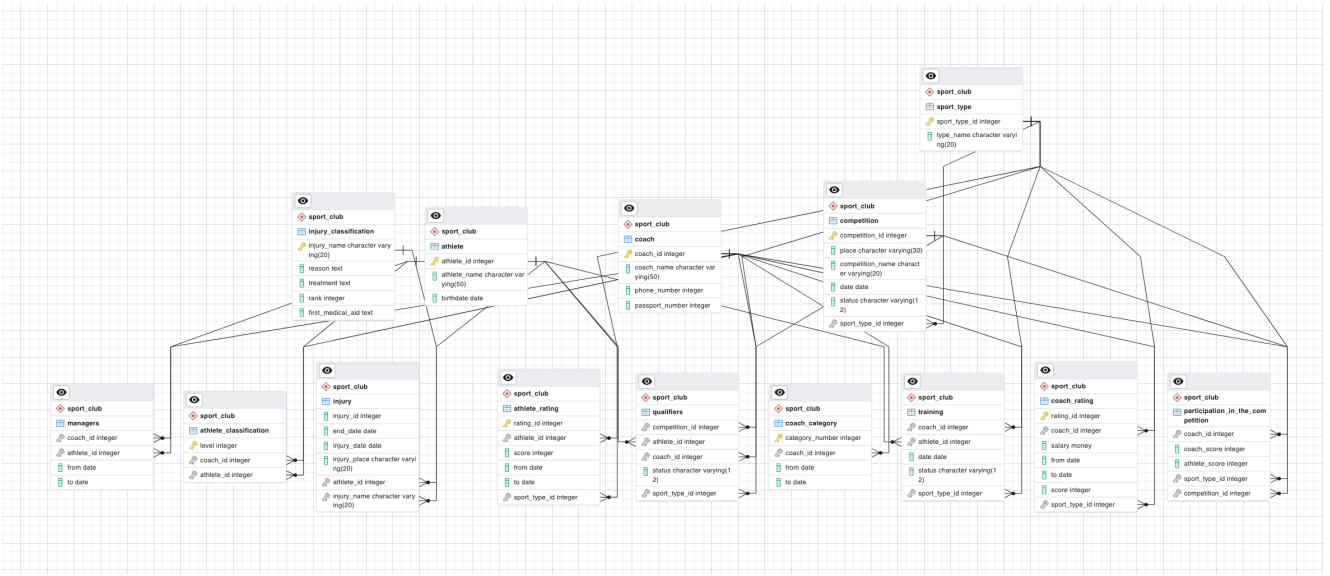

Рисунок 1 – ERD базы данных

Выполнение работы:

1. Создание схемы

```
CREATE SCHEMA IF NOT EXISTS sport_club
    AUTHORIZATION postgres;
```

2. Создание таблиц (вместе с ограничениями)

```
CREATE TABLE IF NOT EXISTS sport_club.athlete
(
    athlete_id integer NOT NULL,
    athlete_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
    birthdate date NOT NULL,
    CONSTRAINT athlete_pkey PRIMARY KEY (athlete_id),
    CONSTRAINT chk_ath_id CHECK (athlete_id > 0) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.athlete
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.athlete_classification
(
    level integer NOT NULL,
    coach_id integer NOT NULL,
    athlete_id integer NOT NULL,
    CONSTRAINT athlete_classification_pkey PRIMARY KEY (level),
    CONSTRAINT athlete_classification_athlete_id_fkey FOREIGN KEY (athlete_id)
        REFERENCES sport_club.athlete (athlete_id) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT athlete_classification_coach_id_fkey FOREIGN KEY (coach_id)
        REFERENCES sport_club.coach (coach_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT chk_class CHECK (level > 0) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.athlete_classification
    OWNER to postgres;

CREATE TABLE IF NOT EXISTS sport_club.athlete_rating
(
    rating_id integer NOT NULL,
    athlete_id integer NOT NULL,
    score integer NOT NULL,
    "from" date NOT NULL,
    "to" date NOT NULL,
    sport_type_id integer NOT NULL,
    CONSTRAINT athlete_rating_pkey PRIMARY KEY (rating_id),
    CONSTRAINT athlete_rating_athlete_id_fkey FOREIGN KEY (athlete_id)
        REFERENCES sport_club.athlete (athlete_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT athlete_rating_sport_type_id_fkey FOREIGN KEY (sport_type_id)
        REFERENCES sport_club.sport_type (sport_type_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.athlete_rating
    OWNER to postgres;

CREATE TABLE IF NOT EXISTS sport_club.coach
(
    coach_id integer NOT NULL,
    coach_name character varying(50) COLLATE pg_catalog."default" NOT NULL,
    phone_number integer NOT NULL,
    passport_number integer NOT NULL,
    CONSTRAINT coach_pkey PRIMARY KEY (coach_id),
    CONSTRAINT chk_coach_id CHECK (coach_id > 0) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.coach
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.coach_category
(
    category_number integer NOT NULL,
    coach_id integer NOT NULL,
    "from" date NOT NULL,
    "to" date NOT NULL,
    CONSTRAINT coach_category_pkey PRIMARY KEY (category_number),
    CONSTRAINT coach_id FOREIGN KEY (coach_id)
        REFERENCES sport_club.coach (coach_id) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.coach_category
    OWNER to postgres;

CREATE TABLE IF NOT EXISTS sport_club.coach_rating
(
    rating_id integer NOT NULL,
    coach_id integer NOT NULL,
    salary money NOT NULL,
    "from" date NOT NULL,
    "to" date NOT NULL,
    score integer NOT NULL,
    sport_type_id integer NOT NULL,
    CONSTRAINT coach_rating_pkey PRIMARY KEY (rating_id),
    CONSTRAINT coach_id FOREIGN KEY (coach_id)
        REFERENCES sport_club.coach (coach_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT coach_rating_sport_type_id_fkey FOREIGN KEY (sport_type_id)
        REFERENCES sport_club.sport_type (sport_type_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.coach_rating
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.competition
(
    competition_id integer NOT NULL,
    place character varying(30) COLLATE pg_catalog."default" NOT NULL,
    competition_name character varying(20) COLLATE pg_catalog."default" NOT NULL,
    date date NOT NULL,
    status character varying(12) COLLATE pg_catalog."default" NOT NULL,
    sport_type_id integer NOT NULL,
    CONSTRAINT competition_pkey PRIMARY KEY (competition_id),
    CONSTRAINT competition_sport_type_id_fkey FOREIGN KEY (sport_type_id)
        REFERENCES sport_club.sport_type (sport_type_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT chk_comp_id CHECK (competition_id > 0) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.competition
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.injury
(
    injury_id integer NOT NULL,
    end_date date NOT NULL,
    injury_date date NOT NULL,
```

```
    injury_place character varying(20) COLLATE pg_catalog."default" NOT NULL,
    athlete_id integer NOT NULL,
    injury_name character varying(20) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT injury_athlete_id_fkey FOREIGN KEY (athlete_id)
        REFERENCES sport_club.athlete (athlete_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT injury_injury_name_fkey FOREIGN KEY (injury_name)
        REFERENCES sport_club.injury_classification (injury_name) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT chk_inj_id CHECK (injury_id > 0) NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.injury
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.injury_classification
(
    injury_name character varying(20) COLLATE pg_catalog."default" NOT NULL,
    reason text COLLATE pg_catalog."default" NOT NULL,
    treatment text COLLATE pg_catalog."default" NOT NULL,
    rank integer NOT NULL,
    first_medical_aid text COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT injury_classification_pkey PRIMARY KEY (injury_name)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.injury_classification
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.managers
(
    coach_id integer NOT NULL,
    athlete_id integer NOT NULL,
    "from" date NOT NULL,
    "to" date NOT NULL,
    CONSTRAINT managers_athlete_id_fkey FOREIGN KEY (athlete_id)
        REFERENCES sport_club.athlete (athlete_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT managers_coach_id_fkey FOREIGN KEY (coach_id)
        REFERENCES sport_club.coach (coach_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.managers
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.participation_in_the_competition
(
    coach_id integer NOT NULL,
    coach_score integer NOT NULL,
```

```
    athlete_score integer NOT NULL,
    sport_type_id integer NOT NULL,
    competition_id integer NOT NULL,
    CONSTRAINT coach_id FOREIGN KEY (coach_id)
        REFERENCES sport_club.coach (coach_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT participation_in_the_competition_competition_id_fkey FOREIGN KEY (competition_id)
        REFERENCES sport_club.competition (competition_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT participation_in_the_competition_sport_type_id_fkey FOREIGN KEY (sport_type_id)
        REFERENCES sport_club.sport_type (sport_type_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.participation_in_the_competition
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.qualifiers
(
    competition_id integer NOT NULL,
    athlete_id integer NOT NULL,
    coach_id integer NOT NULL,
    status character varying(12) COLLATE pg_catalog."default" NOT NULL,
    sport_type_id integer NOT NULL,
    CONSTRAINT qualifiers_athlete_id_fkey FOREIGN KEY (athlete_id)
        REFERENCES sport_club.athlete (athlete_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT qualifiers_coach_id_fkey FOREIGN KEY (coach_id)
        REFERENCES sport_club.coach (coach_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT qualifiers_competition_id_fkey FOREIGN KEY (competition_id)
        REFERENCES sport_club.competition (competition_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT qualifiers_sport_type_id_fkey FOREIGN KEY (sport_type_id)
        REFERENCES sport_club.sport_type (sport_type_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.qualifiers
    OWNER to postgres;

CREATE TABLE IF NOT EXISTS sport_club.sport_type
(
    sport_type_id integer NOT NULL,
    type_name character varying(20) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT sport_type_pkey PRIMARY KEY (sport_type_id),
    CONSTRAINT chk_sptp_id CHECK (sport_type_id > 0) NOT VALID
)

TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS sport_club.sport_type
    OWNER to postgres;


CREATE TABLE IF NOT EXISTS sport_club.training
(
    coach_id integer NOT NULL,
    athlete_id integer NOT NULL,
    date date NOT NULL,
    status character varying(12) COLLATE pg_catalog."default" NOT NULL,
    sport_type_id integer NOT NULL,
    CONSTRAINT training_athlete_id_fkey FOREIGN KEY (athlete_id)
        REFERENCES sport_club.athlete (athlete_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT training_coach_id_fkey FOREIGN KEY (coach_id)
        REFERENCES sport_club.coach (coach_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT training_sport_type_id_fkey FOREIGN KEY (sport_type_id)
        REFERENCES sport_club.sport_type (sport_type_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS sport_club.training
    OWNER to postgres;
```

Выводы:

В данной работе изучались ключевые категории данных, применяемые при формировании таблиц, а также разнообразные ограничения и инструменты резервного копирования для баз данных. Работа по интеграции таблиц с рабочими данными была успешно проведена. Это требовало использования техники импорта данных из внешних ресурсов.

В ходе выполнения практической работы получены знания и умения по созданию таблиц в среде баз данных POSTGRESQL и их заполнению.