

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих комп’ютерних систем**

**Лабораторна робота №1**

з дисципліни **Бази даних і засоби управління**

*на тему: “Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL”*

Виконав:

студент ІII курсу

групи КВ-93

Осика І.О.

Перевірив:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2021

*Метою роботи* є здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

*Завдання* роботи полягає у наступному:

1. Розробити модель «сутність-зв’язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

**Модель «сутність-зв’язок» кіберспортивного турніру**

Згідно цієї області для побудови бази даних було виділено наступні сутності:

1. Гравець, з атрибутами: ім’я, вік, нік, національність. Учасник турніру, який змагається у певній кібер-спортивній дисципліні.
2. Тренер, з атрибутами: ім’я, вік, нік, національність. Учасник турніру, який підтримує та може обговорити подальші дії під час тактичних пауз.
3. Команда, з атрибутами: країна\регіон, назва. Група в яку входить певна кількість гравців та 1-2 тренера.
4. Матч, з атрибутами: дата\час, ID матчу, рахунок. Один з матчів кібер-спортивного турніру, під час якого змагаються дві команди.

**Опис зв’язків**

Один або декілька тренерів можуть тренувати багатьох гравців тому між ними зв’язок M:N. Тобто, один тренер може тренувати багатьох гравців, і один гравець може тренуватися у багатьох тренерів

І тренер(и) і гравці є частиною однієї команди, тому зв’язок 1:N присутній між командою-тренером і командою-гравцем.

І так як це турнір, тому багато команд грають якусь кількість матчів одна проти одної, тож до сутності «матч» від сутності «команда» йдуть два зв’язки 1:N

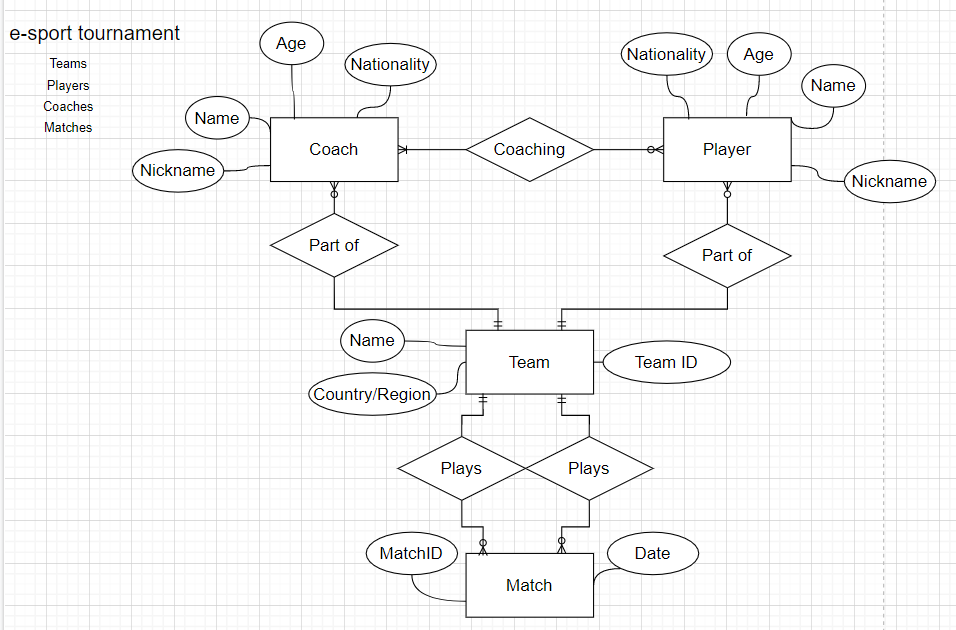


Рисунок 1 - ER-діаграма, побудована за нотацією Чена

**Перетворення моделі у схему бази даних**

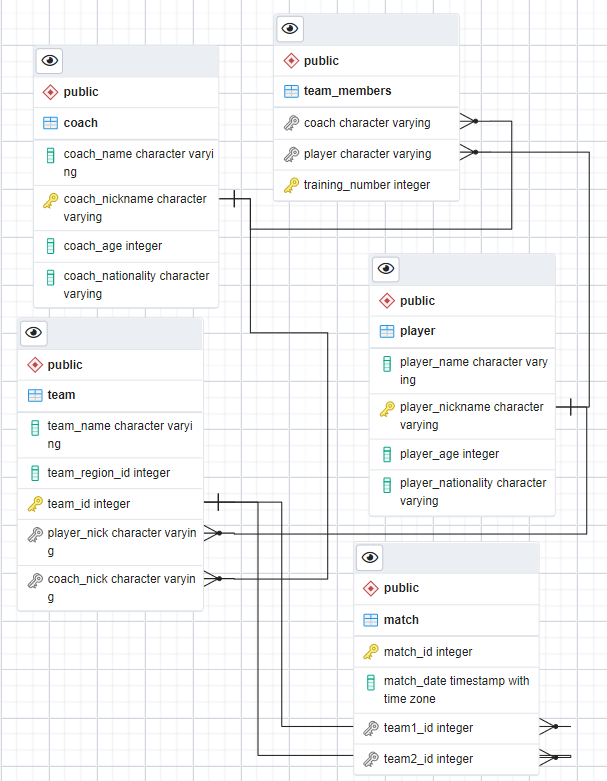


Рисунок 2 - Схема бази даних у pgAdmin 4

**Опис перетворень**

Сутність “Team” було перетворено у таблицю “Team”, а зв’язки 1:N цієї сутності із сутностями “Coach” і “Player” зумовили появу у ній зовнішніх ключів coach\_nickname, player\_nickname.

Сутності “Player” та “Coach” були перетворені у відповідні таблиці, а зв’язок М:М між ними утворив таблицю “Team\_member”, що відображає id тренувань які проводилися тренерами з гравцями.

Сутність “Match” була перетворена у відповідну таблицю, з двома зв’язками 1:N від таблиці “Team”

Опис структури БД

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **player** - *містить дані про гравця* | **Player\_name** *– ПІБ гравця*  **Player\_nickname** *– нік гравця*  **Player\_age** *– вік гравця*  **Player\_nationality** *– його національність* | **character varying** *(рядок)*  **character varying** *(рядок)*  **integer** *(числовий)*  **character varying** *(рядок)* |
| **coach** - *містить дані про тренера* | **coach\_name** *– ПІБ тренера*  **coach\_nickname** *– нік тренера*  **coach\_age** *– вік тренера*  **coach\_nationality** *– його національність* | **character varying** *(рядок)*  **character varying** *(рядок)*  **integer** *(числовий)*  **character varying** *(рядок)* |
| **team** – *дані про команду* | **Team\_name** *– назва команди*  **Team\_region\_id** *– ідентифікатор регіону команди*  **Team\_id** *– унікальний ідентифікатор команди* | **character varying** *(рядок)*  **integer** *(числовий)*  **integer** *(числовий)* |
| **match** – *інформація про матч* | **Match\_id** *- унікальний ідентифікатор матчу*  **Match\_date** *– дата-час матчу* | **integer** *(числовий)*  **timestamp** *(мітка часу)* |

**Нормалізація**

Схема відповідає 1НФ, тому що:

1) Кожна таблиця має основний ключ (Primary key, PK) та мінімальний набір атрибутів, які ідентифікують запис;

2) Кожен атрибут повинен має лише одне значення, а не множину значень.

Схема відповідає 2НФ, тому що:

1) Вона відповідає вимогам 1НФ;

2) Кожен не ключовий атрибут функціонально залежить від основного атрибуту.

Схема відповідає 3НФ, а саме:

1) Вона відповідає вимогам 2НФ;

2) Дані в таблиці залежать винятково від основного ключа

**Изображение выглядит как текст

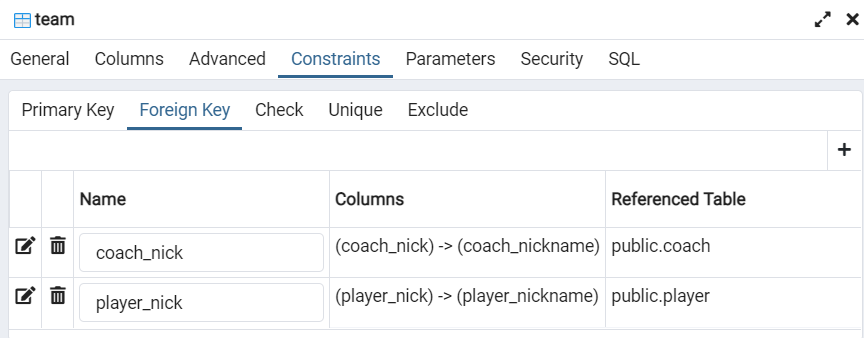
Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описаниеТаблиці бази даних у pgAdmin4**

**Изображение выглядит как стол

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описание**

**Таблиці заповнені даними**

coach

**Изображение выглядит как стол

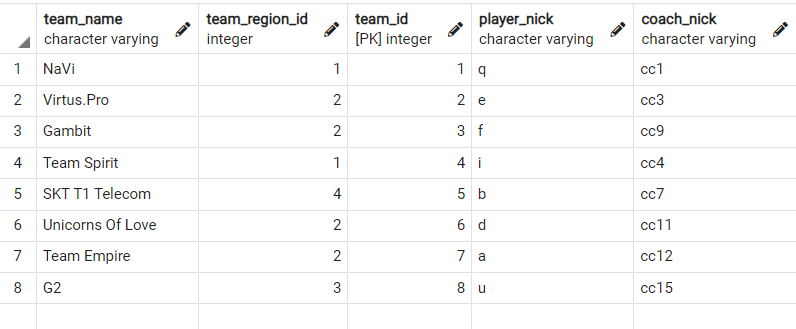
Автоматически созданное описание**

player

**Изображение выглядит как стол

Автоматически созданное описание**

team



Team\_members

Изображение выглядит как стол

Автоматически созданное описание

Match

Изображение выглядит как стол

Автоматически созданное описание

**SQL-опис схеми БД**

-- This script was generated by a beta version of the ERD tool in pgAdmin 4.

-- Please log an issue at https://redmine.postgresql.org/projects/pgadmin4/issues/new if you find any bugs, including reproduction steps.

BEGIN;

CREATE TABLE IF NOT EXISTS public.coach

(

coach\_name character varying NOT NULL,

coach\_nickname character varying NOT NULL,

coach\_age integer NOT NULL,

coach\_nationality character varying NOT NULL,

PRIMARY KEY (coach\_nickname)

);

CREATE TABLE IF NOT EXISTS public.match

(

match\_id integer NOT NULL,

match\_date timestamp with time zone NOT NULL,

PRIMARY KEY (match\_id)

);

CREATE TABLE IF NOT EXISTS public.matches\_journal

(

team\_id integer NOT NULL,

match\_id integer NOT NULL,

team\_score integer NOT NULL,

PRIMARY KEY (team\_score)

);

CREATE TABLE IF NOT EXISTS public.player

(

player\_name character varying NOT NULL,

player\_nickname character varying NOT NULL,

player\_age integer NOT NULL,

player\_nationality character varying NOT NULL,

PRIMARY KEY (player\_nickname)

);

CREATE TABLE IF NOT EXISTS public.team

(

team\_name character varying NOT NULL,

team\_region\_id integer NOT NULL,

team\_id integer NOT NULL,

player\_nick character varying,

coach\_nick character varying,

PRIMARY KEY (team\_id)

);

CREATE TABLE IF NOT EXISTS public.team\_members

(

coach character varying NOT NULL,

player character varying NOT NULL,

training\_number integer NOT NULL,

PRIMARY KEY (training\_number)

);

ALTER TABLE public.matches\_journal

ADD FOREIGN KEY (match\_id)

REFERENCES public.match (match\_id)

NOT VALID;

ALTER TABLE public.matches\_journal

ADD FOREIGN KEY (team\_id)

REFERENCES public.team (team\_id)

NOT VALID;

ALTER TABLE public.team

ADD FOREIGN KEY (coach\_nick)

REFERENCES public.coach (coach\_nickname)

NOT VALID;

ALTER TABLE public.team

ADD FOREIGN KEY (player\_nick)

REFERENCES public.player (player\_nickname)

NOT VALID;

ALTER TABLE public.team\_members

ADD FOREIGN KEY (coach)

REFERENCES public.coach (coach\_nickname)

NOT VALID;

ALTER TABLE public.team\_members

ADD FOREIGN KEY (player)

REFERENCES public.player (player\_nickname)

NOT VALID;

END;