



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

ПОЯСНЮВАЛЬНА ЗАПИСКА
до курсової роботи
з дисципліни
«ОСНОВИ ПРОГРАМУВАННЯ»

на тему
«Інформаційна система першості країни з футболу»

Викладач В.В. Хайдуров
Студент Дерменжи Ілля Сергійович
Група КП-13
Варіант 11

Завдання до курсової роботи

Варіант 11. Інформаційна система першості країни з футболу

БД повинна забезпечувати зберігання відомостей про командах, що беруть участь в першості, про гравців команд, про розклад зустрічей і їх результати, про ціну квитків на ігри.

Відомості про команду – назва команди, місто, де вона базується, ім'я головного тренера, місце в таблиці минулого сезону, розклад зустрічей. В один день команда може брати участь тільки в одній зустрічі.

Відомості про гравця включають в себе прізвище та ім'я гравця, його вік, номер і амплуа в команді. Відомості про стадіон, на якому відбувається зустріч містять місто, в якому він знаходиться, назва стадіону, і його місткість. Ціна квитка на матч залежить від місткості стадіону і положення команд, що зустрічаються в турнірній таблиці минулого сезону (найбільша - при грі трійки призерів, найменша – при грі трійки аутсайдерів). Організатори змагань повинні мати можливість внести зміни в дані про склад команд, перенести зустріч.

Їм можуть знадобитися такі відомості.

- Дати зустрічей зазначеної команди, її противники і рахунок.
- Номери та прізвища гравців команд, які брали участь у зустрічі, яка проходила в зазначений день в зазначеному місті.
- Ціна, квитка на матч між зазначеними командами.
- Гравець, який забив у турнірі найбільшу кількість м'ячів.
- Команди, які мають найкращу і найгіршу різницю забитих і пропущених м'ячів.
- Наймолодший учасник турніру.
- Команди, які посіли призові місця.
- Розклад ігор по стадіонах.

За результатами турніру повинен бути представлений звіт з результатами кожної гри. Для кожної гри вказується місце і час її проведення, команди – учасниці, рахунок, гравці, які виходили на поле, і гравці, які забили м'ячі (вказати на якій хвилині). У звіті повинні бути вказані призери турніру і команда, що посіла останнє місце.

Окрім цього, були розширені таблиці про гравців (країна походження, коротка статистика ігор та умови контракту з клубом), ігри (погодні умови, час початку-кінця, ім'я рефері, кількість відвідувачів), таблиця результатів та створена окрема таблиця обліку клубного бюджету.

ЗМІСТ

Завдання до курсової роботи.....	1
Вступ	2
1. Аналіз предметної області.....	4
2. Функціональне моделювання інформаційної системи національної першості з футболу з використанням DFD-діаграм.....	5
3. Концептуальна модель бази даних	7
4. Фізична модель бази даних системи інформаційної служби національної першості з футболу	10
5. Опис реалізації завдань відповідно до обраного варіанту.....	14
5.1. Створення та структура таблиць бази даних інформаційної системи національної першості з футболу.....	14
5.2 Формування запитів додавання даних до створеної бази даних	17
5.3 Формування запитів пошуку даних у створеній базі даних національної першості з футболу	25
Висновок	31
Список використаних джерел	32
Додаток А.....	33
Додаток Б	37

Вступ

Актуальність досліджень. Сучасне суспільство існує в Інформаційну еру, що визначається провідною роллю повсякденної взаємодії з різноманітною інформацією. Необхідність зберігання та організації даних, що складають інформаційний простір зумовила розвиток всеможливих баз даних та застосунків для роботи з ними. Очевидно, що жодна організація, що оперує інформацією, не може обійтись без використання тієї чи іншої бази даних. Зокрема, спортивні організації національного чи міжнародного рівня користуються певними інформаційними ресурсами, організованими у відповідні структури. В контексті національних змагань база даних може знадобитись для створення таблиці результатів, організації інформації щодо конкретного матчу чи гравця.

Мета роботи. Метою роботи є побудова працюючої системи обліку команд, гравців, стадіонів та матчів національної першості країни з можливістю пошуку, обробки та модифікації даних через зручності користувацького інтерфейсу.

Завдання дослідження. Завдання дослідження цієї інформаційної системи полягає в додаванні та редагуванні даних про: команди, гравців, бюджет, стадіони, таблицю результатів тощо. Програмний застосунок передбачає можливість як прямого введення команд через командний рядок, що пов'язує проект мовою с# з базою даних, так і використання деяких типових запитів на пошук та модифікації даних футбольних змагань.

Порядок виконання завдання. Безпосередній розробці програмного застосунку передував глибокий аналіз області дослідження з розподілом важливої та другорядної інформації в контексті національних футбольних змагань з подальшим розширенням чи зменшенням таблиці даних шляхом додавання та вилучення певних полів. Проведено дослідження необхідних структурних зв'язків у відношення один-до-одного, один-до-багатьох та багато-до-багатьох з використанням первинних та зовнішніх ключів. Після аналізу предметної області відбулось безпосереднє створення бази даних та необхідних таблиць до неї у визначеній СУБД з поєднанням таблиць через ключі з'єднання. До створених таблиць були додані випадково

згенеровані дані, коректність яких визначається спеціальною програмою заповнення та обробки інформації. Для виконання цих запитів було створено програмний застосунок з користувацьким інтерфейсом, що переводить відповідні запити до бази даних. У процесі заповнення таблиці також виконувались операції пошуку, видалення та оновлення даних, що представлені відповідним графічним інтерфейсом (зокрема, до операції вилучення було додано набір типових команд). Створена база даних, програмний застосунок генерації даних та користувацький інтерфейс становлять кінцевий результат виконання дослідницької роботи.

Засоби реалізації завдань проектування. До розробки інформаційної системи національної першості з футболу було залучено два основні програмні застосунки. В якості СУБД було обрано програму PostgreSQL (середовище роботи з реляційними базами даних), що використовується задля зберігання та організації даних. Для створення користувацького інтерфейсу та генерації даних до бази даних була використана мова програмування C# та XAML (мова розмітки WPF) у застосунку Microsoft Visual Studio 2022.

1. Аналіз предметної області

Інформаційна система національної першості з футболу містить певний набір даних різного типу (текстового, числового, date-тип тощо), що безпосередньо стосується команд, гравців, стадіонів та інших важливих для чемпіонату подробиць.

Відповідна база даних повинна:

1. Складатися з окремих таблиць
2. Містити ідентифікатор у якості обов'язкового поля для таблиці
3. Мати коректний тип даних, що відповідає інформації кожного поля
4. Поєднувати таблиці через первинні та зовнішні ключі
5. Відповідати нормам іменування та організації даних

Відповідний користувацький інтерфейс повинен:

1. Мати безпосередній доступ до середовища СУБД
2. Надавати можливість додавання, оновлення, видалення та пошуку даних
3. Містити типові запити для пошуку даних
4. Бути інтуїтивно зрозумілим
5. Відповідати нормам іменування, розмітки тощо

2. Функціональне моделювання інформаційної системи національної першості з футболу з використанням DFD-діаграм

DFM-діаграми передбачають створення графічної структури, що моделює систему обміну даних між джерелом та адресатом та використовується для аналізу потоків даних конкретної бази.

На рис. 2.1 зображено структуру потоків даних у базі даних національної першості з футболу. Стрілками показані зв'язки між адресатом та джерелом передачі (можлива як двостороння взаємодія, так і одностороння між різними частинами загальної інформаційної системи).

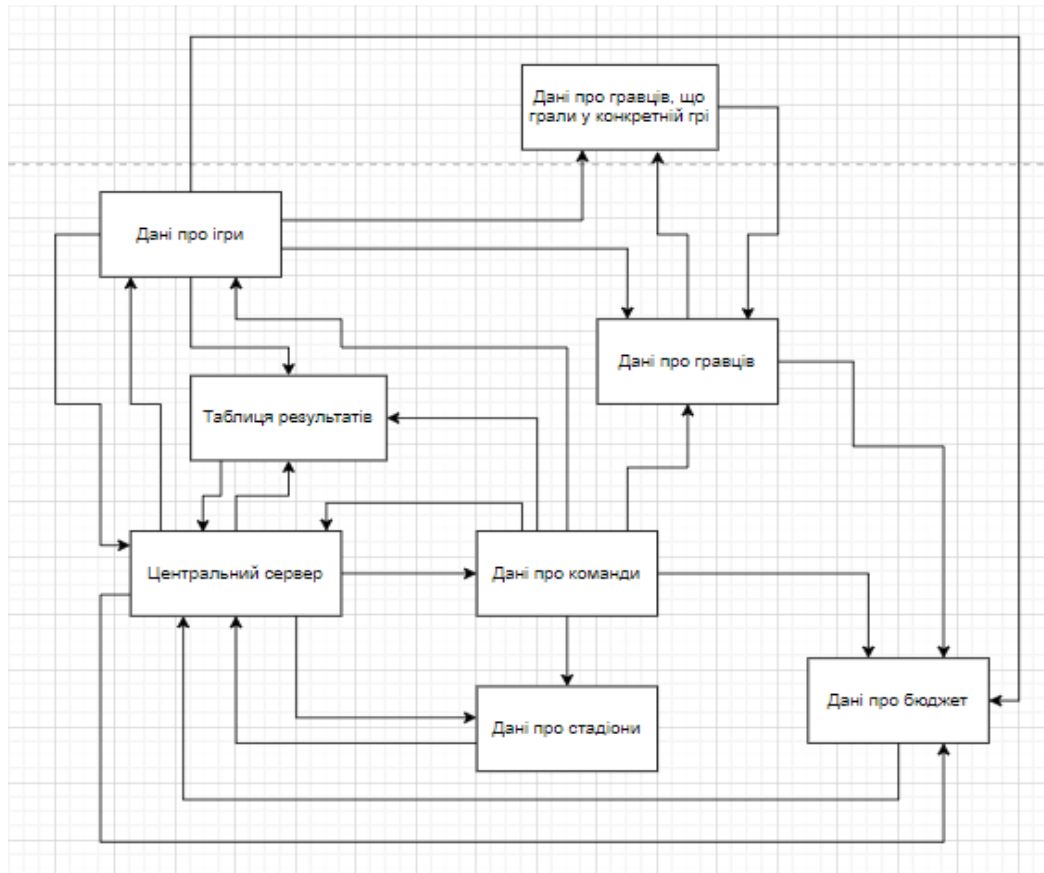


Рис. 2.1 Діаграма потоків даних інформаційної системи національної першості з футболу

Функціональність інформаційної системи передбачає:

- редагування даних про гравців, ігри, табличні результати команд, бюджет
- створення запитів у реальному часі на додавання, оновлення, видалення та пошук даних
- формування однозначних зв'язків один-до-одного, один-до-багатьох та багато-до-багатьох між таблицями
- динамічне оновлення даних про статистику ігор гравців, конкретні матчі та бюджет клубу

Як результат, для забезпечення функціональних можливостей системи національної першості з футболу згодиться будь-яка операційна система, що підтримує x32 та x64 розрядність. Системні вимоги до пам'яті не мають перевищувати 500 Mb операційної пам'яті та 2 Gb постійної.

3. Концептуальна модель бази даних

Концептуальна модель знань описує предметну область через набір концептів. Ця модель може бути подана у різних видах (текстовому, словесному, у вигляді діаграми тощо).

На рис. 3.1 продемонстрована ER-діаграма, що використовувалась при розробці бази даних інформаційної системи національної першості з футболу.

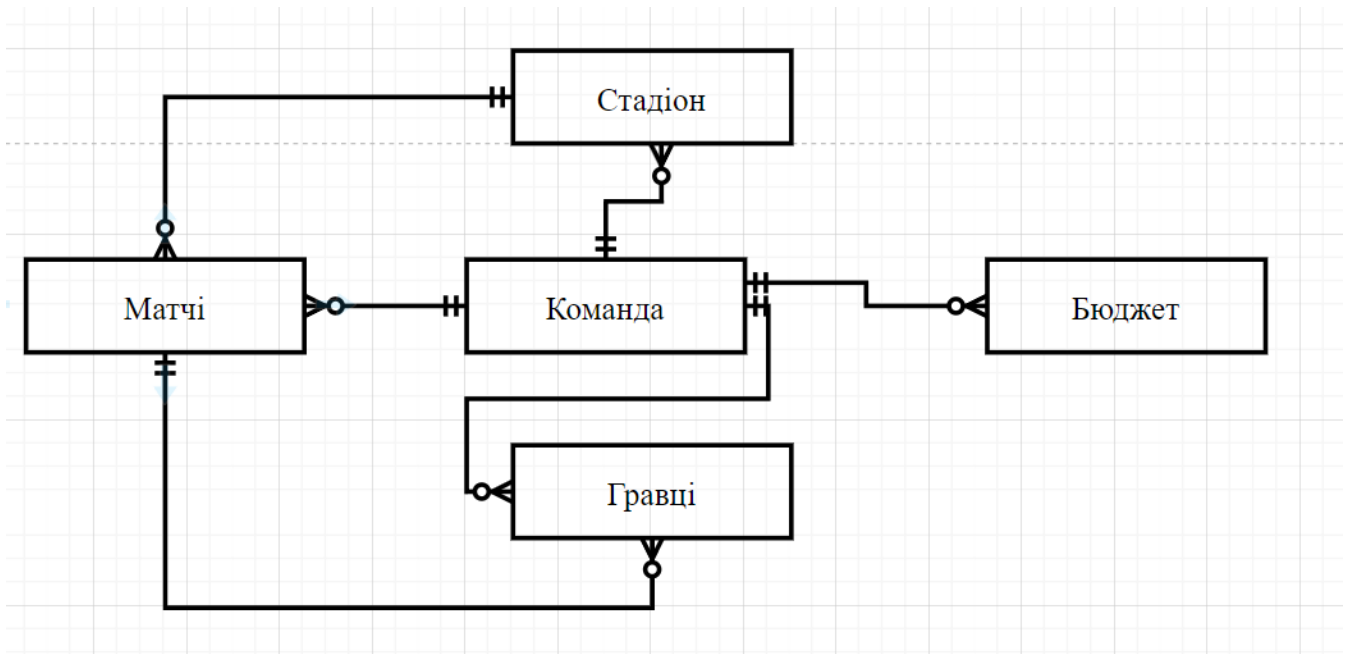


Рис. 3.1 ER-діаграма інформаційної системи бази даних національної першості з футболу

На рис. 3.2-3.4 продемонстрована загальна діаграма організації бази даних (логічне представлення) у середовищі PostgreSQL з відповідними зв'язками, що реалізовані через первинні та зовнішні ключі.

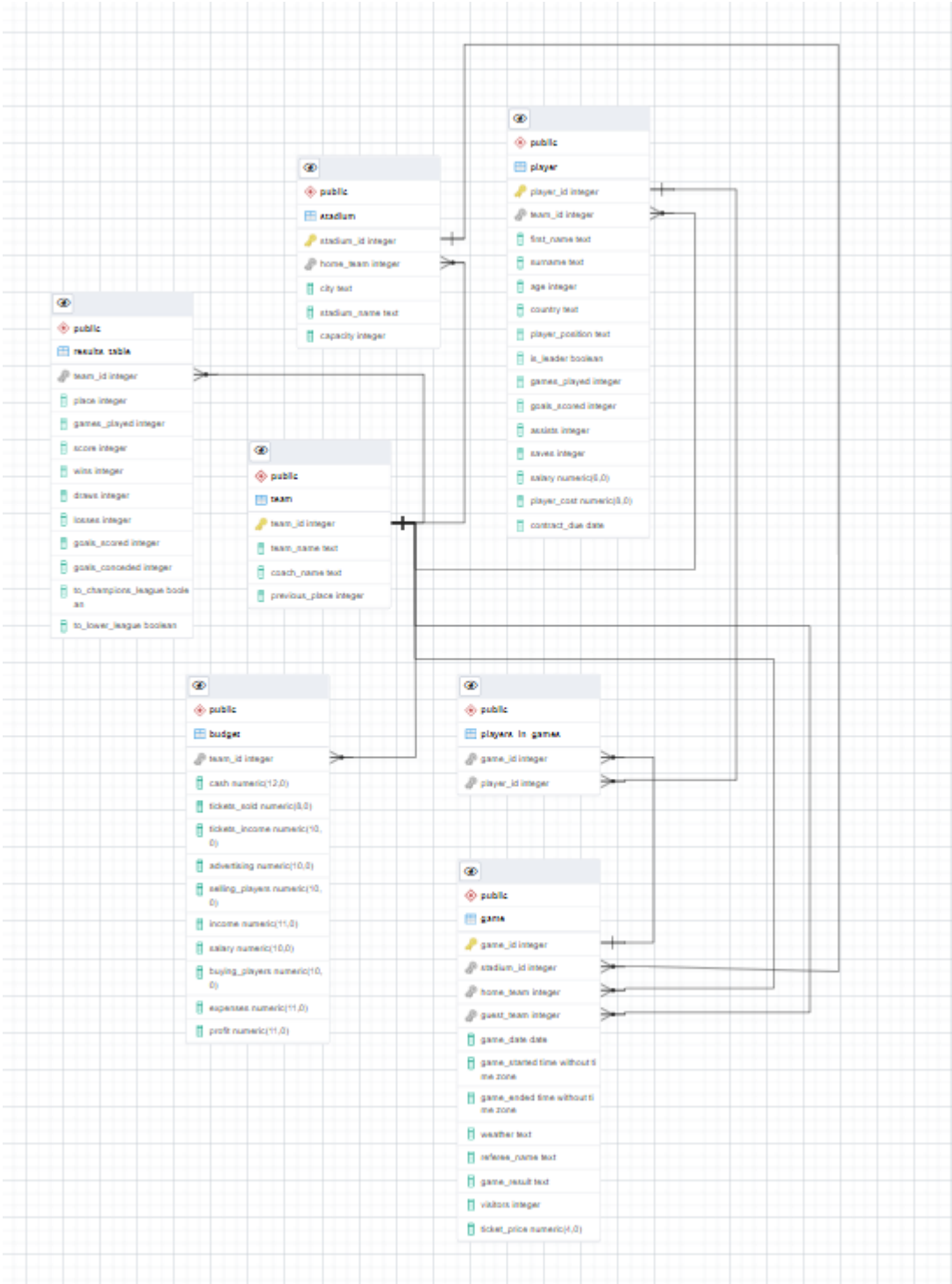


Рис. 3.2 Загальний план

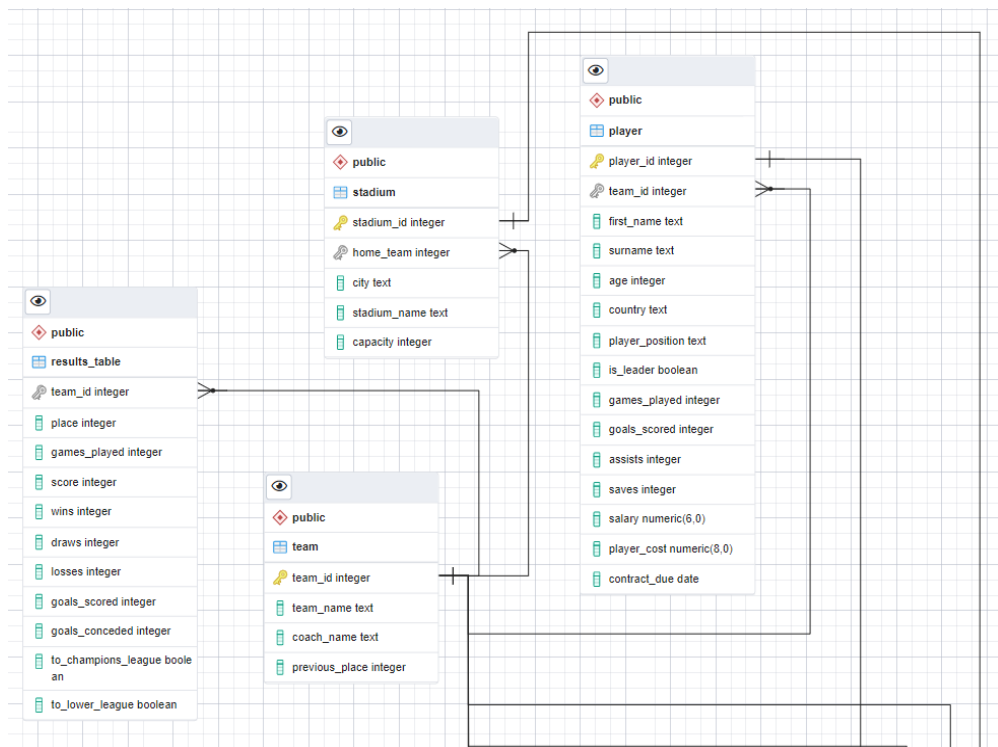


Рис. 3.3 загалый план, верхня частина

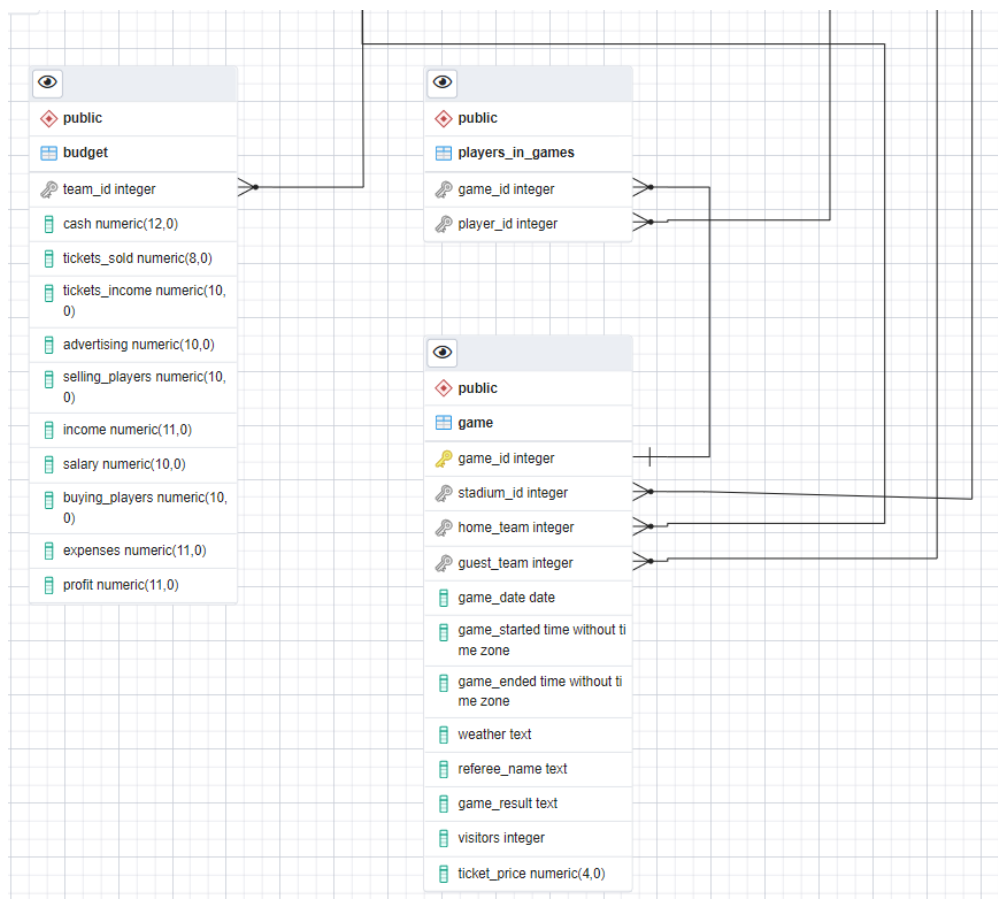


Рис. 3.4 загалый план, нижня частина

4. Фізична модель бази даних системи інформаційної служби національної першості з футболу

Фізична модель – логічна модель бази даних, виражена в термінах мови опису даних конкретної СКБД. Обмеження, наявні в логічній моделі даних, реалізуються різними засобами СКБД, наприклад, за допомогою індексів, декларативних обмежень цілісності, тригерів, процедур, що зберігаються. При цьому рішення, прийняті на рівні логічного моделювання, визначають деякі обмеження, у межах яких можна розвивати фізичну модель даних.

Каскадне видалення таблиць у випадку, якщо такі вже існують

```
DROP TABLE IF EXISTS team CASCADE;
DROP TABLE IF EXISTS budget CASCADE;
DROP TABLE IF EXISTS player CASCADE;
DROP TABLE IF EXISTS stadium CASCADE;
DROP TABLE IF EXISTS game CASCADE;
DROP TABLE IF EXISTS results_table CASCADE;
DROP TABLE IF EXISTS players_in_games CASCADE;
```

Створення таблиці «team»

```
CREATE TABLE team
(
    team_id serial PRIMARY KEY,

    team_name text UNIQUE,
    coach_name text UNIQUE,
    previous_place integer UNIQUE
);
```

Створення таблиці «player»

```
CREATE TABLE player
(
    player_id serial PRIMARY KEY,
    team_id serial NOT NULL
        REFERENCES team(team_id),

    first_name text NOT NULL,
    surname text NOT NULL,
    age integer NOT NULL,
    country text NOT NULL,

    player_position text NOT NULL,
    is_leader boolean NOT NULL,

    games_played integer,
```

```

goals_scored integer,
assists integer,
saves integer,

salary numeric (6, 0),
player_cost numeric(8, 0),
contract_due date
);

```

Створення таблиці «budget»

```

CREATE TABLE budget
(
    team_id serial NOT NULL
        REFERENCES team(team_id),

    cash numeric(12,0),
    tickets_sold numeric(8,0),
    tickets_income numeric(10,0),
    advertising numeric(10,0),
    selling_players numeric(10,0),
    income numeric(11,0),

    salary numeric(10,0),
    buying_players numeric(10,0),
    expenses numeric(11,0),

    profit numeric(11,0)
);

```

Створення таблиці «stadium»

```

CREATE TABLE stadium
(
    stadium_id serial PRIMARY KEY,
    home_team serial NOT NULL
        REFERENCES team(team_id),

    city text,
    stadium_name text NOT NULL,
    capacity integer
);

```

Створення таблиці «game»

```

CREATE TABLE game
(
    game_id serial PRIMARY KEY,
    stadium_id serial NOT NULL
        REFERENCES stadium(stadium_id),
    home_team serial NOT NULL
        REFERENCES team(team_id),
    guest_team serial NOT NULL

```

```
REFERENCES team(team_id),

game_date DATE,
game_started TIME,
game_ended TIME,
weather text,
referee_name text NOT NULL,
game_result text NOT NULL,
visitors integer,
ticket_price numeric(4,0)
);
```

Створення таблиці «results_table»

```
CREATE TABLE players_in_games
(
    game_id serial NOT NULL
        REFERENCES game(game_id),
    player_id serial NOT NULL
        REFERENCES player(player_id)
);
```

Слід зазначити, що дослідницькою роботою передбачена можливість імплементації даних команд як через аутентичний застосунок PostgreSQL СУБД (рис. 4.1), так за допомогою окремо розробленої програми з відповідним графічним користувацьким інтерфейсом (рис. 4.2)

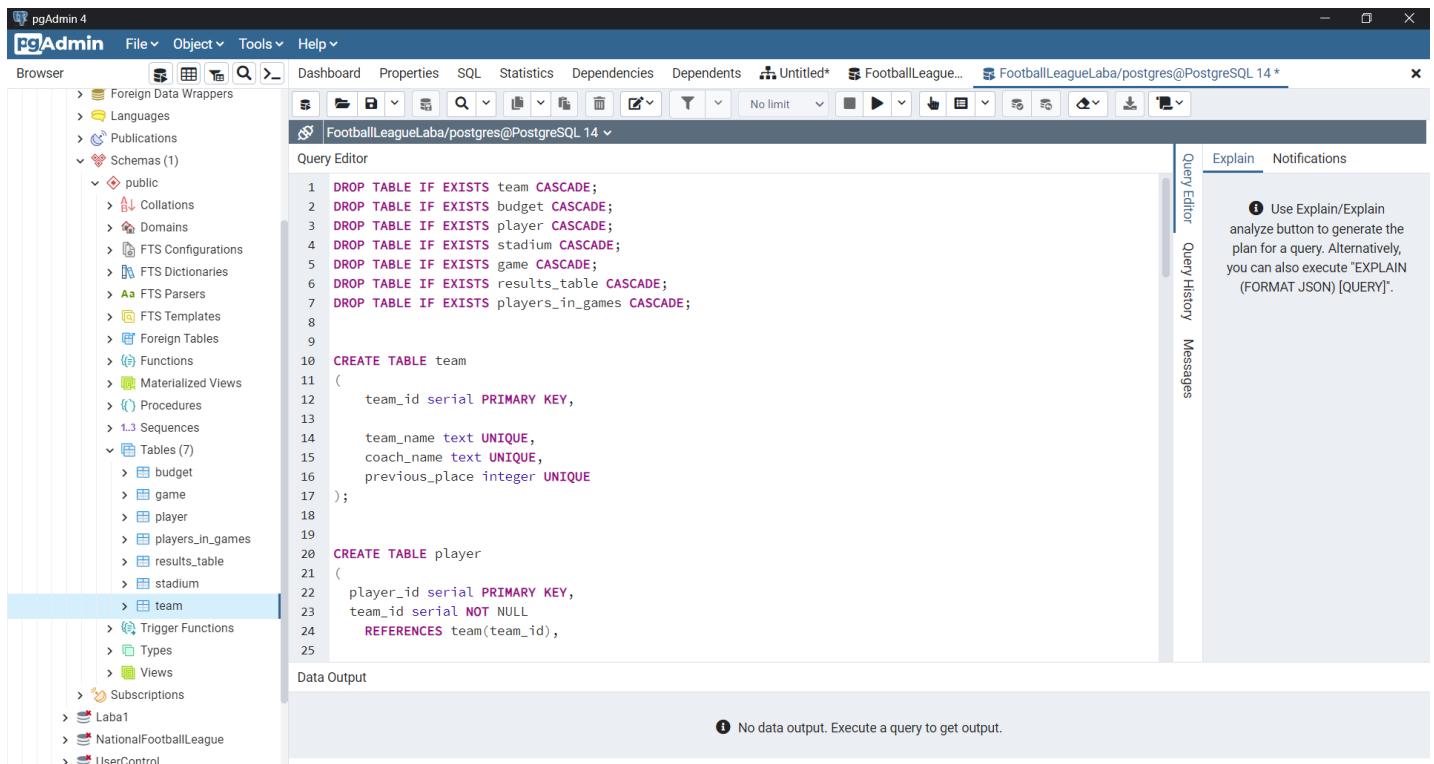


Рис. 4.1 створення таблиць до бази даних через застосунок PostgreSQL (pgAdmin 4)

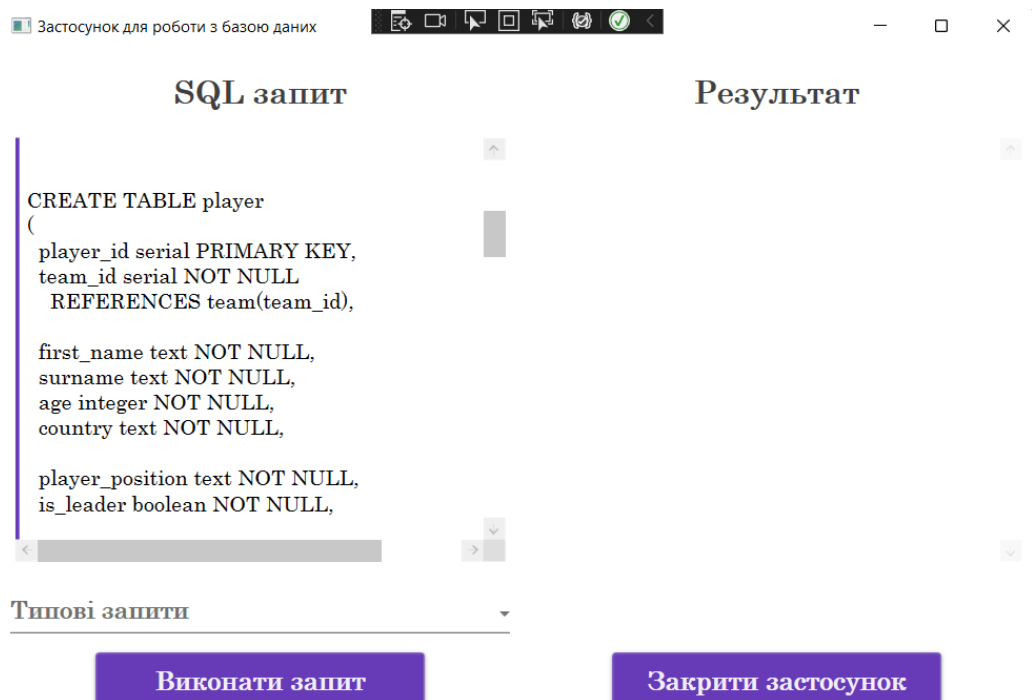


Рис. 4.2 Той самий SQL-запит зі застосуванням простої програми, написаної на мові C#

5. Опис реалізації завдань відповідно до обраного варіанту

5.1. Створення та структура таблиць бази даних інформаційної системи національної першості з футболу

Структури таблиць розробленої бази даних інформаційної системи національної першості з футболу показані у табл. 5.1–5.11.

Таблиця 5.1

Структура таблиці «team»

Назва стовпця	Тип даних	Унікальний	Первинний ключ	Зовнішній ключ
team_id	SERIAL	ТАК	ТАК	НІ
team_name	TEXT	ТАК	НІ	НІ
coach_name	TEXT	ТАК	НІ	НІ
previous_place	INTEGER	ТАК	НІ	НІ

Таблиця 5.2

Структура таблиці «player»

Назва стовпця	Тип даних	Унікальний	Первинний ключ	Зовнішній ключ
player_id	SERIAL	ТАК	ТАК	НІ
team_id	SERIAL	НІ	НІ	ТАК
first_name	TEXT	НІ	НІ	НІ
surname	TEXT	НІ	НІ	НІ
age	INTEGER	НІ	НІ	НІ
country	TEXT	НІ	НІ	НІ
player_position	TEXT	НІ	НІ	НІ
is_leader	BOOLEAN	НІ	НІ	НІ
games_played	INTEGER	НІ	НІ	НІ
goals_scored	INTEGER	НІ	НІ	НІ
assists	INTEGER	НІ	НІ	НІ
salary	NUMERIC (6, 0)	НІ	НІ	НІ
player_cost	NUMERIC (8,0)	НІ	НІ	НІ
contract_due	DATE	НІ	НІ	НІ

Структура таблиці «budget»

Назва стовпця	Тип даних	Унікальний	Первинний ключ	Зовнішній ключ
team_id	SERIAL	ТАК	НІ	ТАК
cash	NUMERIC(12,0)	НІ	НІ	НІ
tickets_sold	NUMERIC(12,0)	НІ	НІ	НІ
tickets_income	NUMERIC(10,0)	НІ	НІ	НІ
advertising	NUMERIC(10,0)	НІ	НІ	НІ
selling_players	NUMERIC(10,0)	НІ	НІ	НІ
income	NUMERIC(11,0)	НІ	НІ	НІ
salary	NUMERIC(10,0)	НІ	НІ	НІ
buying_players	NUMERIC(10,0)	НІ	НІ	НІ
expenses	NUMERIC(11,0)	НІ	НІ	НІ
profit	NUMERIC(11,0)	НІ	НІ	НІ

Структура таблиці «stadium»

Назва стовпця	Тип даних	Унікальний	Первинний ключ	Зовнішній ключ
stadium_id	SERIAL	ТАК	ТАК	НІ
home_team	SERIAL	ТАК	НІ	ТАК
city	TEXT	НІ	НІ	НІ
capacity	INTEGER	НІ	НІ	НІ

Структура таблиці «game»

Назва стовпця	Тип даних	Унікальний	Первинний ключ	Зовнішній ключ
game_id	SERIAL	ТАК	ТАК	НІ
stadium_id	SERIAL	НІ	НІ	ТАК
home_id	SERIAL	НІ	НІ	ТАК
guest_team	SERIAL	НІ	НІ	ТАК
game_date	DATE	НІ	НІ	НІ
game_started	TIME	НІ	НІ	НІ
game_ended	TIME	НІ	НІ	НІ
weather	TEXT	НІ	НІ	НІ
referee_name	TEXT	НІ	НІ	НІ
visitors	INTEGER	НІ	НІ	НІ
ticket_price	NUMERIC(4,0)	НІ	НІ	НІ

Структура таблиці «results_table»

Назва стовпця	Тип даних	Унікальний	Первинний ключ	Зовнішній ключ
team_id	SERIAL	ТАК	НІ	ТАК
place	INTEGER	ТАК	НІ	НІ
games_played	INTEGER	НІ	НІ	НІ
score	INTEGER	НІ	НІ	НІ
wins	INTEGER	НІ	НІ	НІ
draws	INTEGER	НІ	НІ	НІ
losses	INTEGER	НІ	НІ	НІ
goals_scored	INTEGER	НІ	НІ	НІ
goals_conceded	INTEGER	НІ	НІ	НІ
to_champions_league	BOOLEAN	НІ	НІ	НІ
to_lower_league	BOOLEAN	НІ	НІ	НІ

Структура таблиці «players_in_games»

Назва стовпця	Тип даних	Унікальний	Первинний ключ	Зовнішній ключ
game_id	SERIAL	НІ	НІ	ТАК
player_name	SERIAL	НІ	НІ	ТАК

5.2 Формування запитів додавання даних до створеної бази даних

У зв'язку з необхідністю відповідності даних між різними таблицями (наприклад, загальна кількість м'ячів у таблиці результатів не може різнитись від такої ж суми особистої статистики гравців) та здоровому глузду (неможливість двох ігор для однієї команди в один день, сніжна погода влітку, забиття голу голкіпером тощо) була створена окрема програма у застосунку Microsoft Visual Studio 2022 мовою С# (код до програми можна побачити у папці «Заповнення бази даних»), що знаходиться у тій же комірці, що і звіт.

Була здійснена валідація наступних полів:

Команда

- Унікальна назва команди/клубу (генерується зі списку назв 20 команд, що представлені в англійській Прем'єр-лізі, файл зі списком знаходиться у папці «матеріали» під назвою «teams.txt»)
- Унікальне ім'я для тренера (генерується зі списку типових європейських імен та прізвищ, що знаходяться у файлах «names.txt» та «surname.txt» відповідно).
- Унікальне положення у таблиці результатів попереднього сезону

Гравець

- Вік, що відповідає рамці мінімального для гри в Лізі та максимального для продовження професійної спортивної кар'єри
- Один лідер на усю команду
- Позиція на полі, що відповідає футбольним вимогам у 1 голкіпері та стандартним схемам гри, що визначають кількість захистників-напівзахистників-форвардів.

- Одна з реальних країн світу у полі «country» (файл зі списком країн, з яких робиться вибір знаходиться у папці «Матеріали»)

Бюджет

- Збалансовані статті доходу та видатків
- Цифри, що відповідають бюджетам реальних команд

Стадіон

- Унікальна назва (обирається поміж запропонованих назв реальних клубів, що знаходяться у файлі «stadiumNames.txt»)
- Одне з реальних англійських міст у полі «city»
- Місткість, що відповідає футбольним стандартам для стадіону

Гра

- Визначення пари домашня-гостьова команда як тих, що ще не грали між собою (або грали, але були по іншу сторону відношення «господар-гість»)
- Визначення дати гри як тієї, у яку ці дві команди не мають інших зустрічей (а також у попередній та наступний день)
- Визначення часу завершення матчу через час початку та додані хвилини
- Визначення погоди в залежності від пори року
- Генерація кількості глядачів, що залежить від ціни квитка, погоди, місткості стадіону та рейтингу команд
- Генерація ціни квитка в залежності від рейтингу команд та стадіону

Таблиця результатів

- Підрахунок загальної позиції у рейтингу у відповідності до кількості очок (в разі однакової кількості позиція визначається через різницю забити-пропущених м'ячів, особисті зустрічі команд тощо)
- Кількість очок підраховується за формулою $s = w * 3 + d$, де s – кількість очок, w – кількість перемог, d – кількість нічиїх

Таблиця, що поєднує гравців та ігри

- Містить по 22 гравця на кожному гру (по 11 з кожної сторони, без замінів)

SQL-запити на заповнення таблиць даними

Попереднє видалення таблиць, якщо такі існують

```
DELETE FROM players_in_games CASCADE;
DELETE FROM budget CASCADE;
DELETE FROM game CASCADE;
DELETE FROM stadium CASCADE;
DELETE FROM player CASCADE;
DELETE FROM results_table CASCADE;
DELETE FROM team CASCADE;
```

Додавання даних до таблиці team

```
INSERT INTO team (team_name, coach_name, previous_place)
VALUES (@team_name, @coach_name, @previous_place);
```

Додавання даних до таблиці stadium

```
INSERT INTO stadium
(
    home_team, city,
    stadium_name, capacity
)
VALUES
(
    (
        SELECT team_id
        FROM team
        WHERE team_name = @team_name
    ),
    @city, @stadium_name,
    @capacity
);
```

Аналогічно прописані запити на додавання даних до таблиць budget, game, players_in_games, player та results_table (детальніше див. Додаток А).

SQL-запити на оновлення таблиць з даними

Оновлення таблиці game

```
UPDATE game
SET stadium_id =
(
    SELECT stadium_id
    FROM stadium
    WHERE home_team =
        (
            SELECT team_id
            FROM team
            WHERE team_name = @home_team
        )
),
home_team =
(
    SELECT team_id
    FROM team
    WHERE team_name = @home_team
),
guest_team =
(
    SELECT team_id
    FROM team
    WHERE team_name = @guest_team
),
game_date = @game_date,
game_started = @game_started,
game_ended = @game_ended,
weather = @weather,
referee_name = @referee_name,
game_result = @game_result,
visitors = @visitors,
ticket_price = @ticket_price
WHERE home_team =
(
    SELECT team_id
    FROM team
    WHERE team_name = @home_team
)
AND guest_team =
(
    SELECT team_id
    FROM team
    WHERE team_name = @guest_team
);
```

Аналогічно прописані запити на оновлення даних інших таблиць (детальніше див. Додаток Б)

Результат запитів

Як результат, маємо заповнення бази даних (її таблиць) відповідними значеннями, що зображено на рис. 5.7 – 5.8

	team_id integer	cash numeric (12)	tickets_sold numeric (8)	tickets_income numeric (10)	advertising numeric (10)	selling_players numeric (10)	income numeric (11)	salary numeric (10)	buying_players numeric (10)	expenses numeric (11)	profit numeric (11)
1	1	55680255	287653	15346405	3269581	0	18615986	1352871	0	6764355	11851631
2	2	69041235	293185	16781042	4246846	0	21027888	985783	0	4928915	16098973
3	3	41943537	243889	12942754	3203221	0	16145975	1483031	0	7415155	8730820
4	4	22462653	156791	7073620	4211396	0	11285016	1499044	0	7495220	3789796
5	5	20587760	150793	7809378	2172489	0	9981867	1431002	0	7155010	2826857
6	6	37799262	243256	11681086	2731966	0	14413052	1360749	0	6803745	7609307
7	7	25240624	173881	8112317	3373455	0	11485772	1376762	0	6883810	4601962
8	8	37038014	251120	10213146	3668476	0	13881622	1242659	0	6213295	7668327
9	9	28243276	195993	9172444	3113607	0	12286051	1482886	0	7414430	4871621
10	10	54021321	336428	14747858	3107886	0	17855744	1178699	0	5893495	11962249
11	11	52246513	307265	12977275	3682670	0	16659945	1008377	0	5041885	11618060
12	12	12426980	128439	5015787	2648127	0	7663914	1221707	0	6108535	1555379
13	13	45596161	288588	11949623	3276353	0	15225976	1144161	0	5720805	9505171
14	14	20444851	176906	6966500	3497423	0	10463923	1381125	0	6905625	3558298
15	15	30902117	242317	9110261	3788143	0	12898404	1364704	0	6823520	6074884
16	16	15868310	151846	6176620	2873324	0	9049944	1404997	0	7024985	2024959
17	17	6063612	94654	3442880	2754530	0	6197410	1307079	0	6535395	-337985
18	18	36740322	248539	9037374	3688739	0	12726113	1056817	0	5284085	7442028

Рис. 5.7 Заповнення таблиці *budget*

	game_id [PK] integer	stadium_id integer	home_team integer	guest_team integer	game_date date	game_started time without time zone	game_ended time without time zone	weather text	referee_name text	game_result text
65	65	12	12	8	2022-10-08	13:15:00	15:08:00	Clear	Eric Martin	2:1
66	66	13	13	11	2022-10-09	18:00:00	19:49:00	Clear	Brian King	2:1
67	67	17	17	15	2022-10-08	20:00:00	21:49:00	Clear	Jacob Winston	1:0
68	68	7	7	17	2022-10-09	19:30:00	21:21:00	Clear	Leonardo Walsh	2:2
69	69	16	16	7	2022-10-08	15:45:00	17:34:00	Fog	Hugo Lee	3:2
70	70	19	19	13	2022-10-09	13:15:00	15:08:00	Clear	Mattia Johnson	1:3
71	71	14	14	18	2022-10-14	13:30:00	15:21:00	Clear	Michael Rossi	2:0
72	72	1	1	19	2022-10-13	20:15:00	22:06:00	Clear	Benas White	1:3
73	73	4	4	19	2022-10-13	17:00:00	18:51:00	Clear	Leo Rodriguez	1:0
74	74	18	18	7	2022-10-14	21:00:00	22:50:00	Rain	Daniel Petit	3:6
75	75	9	9	1	2022-10-12	19:00:00	20:51:00	Fog	Jan Williams	5:3
76	76	16	16	6	2022-10-14	14:15:00	16:07:00	Rain	Paul Taylor	1:1
77	77	11	11	15	2022-10-12	21:30:00	23:22:00	Clear	Christian Kowalski	2:2
78	78	13	13	13	2022-10-13	14:15:00	16:06:00	Fog	Antonio Petit	1:0
79	79	11	11	5	2022-10-12	14:30:00	16:20:00	Fog	Gary Moore	1:3
80	80	2	2	7	2022-10-13	20:00:00	21:50:00	Clear	Liam Harris	0:2
81	81	11	11	17	2022-10-18	20:00:00	21:51:00	Rain	Adam Taylor	5:2
82	82	3	3	5	2022-10-17	18:00:00	19:49:00	Clear	Arthur Turner	2:0

Рис. 5.8 Заповнення таблиці *game*

	player_id [PK] integer	team_id integer	first_name text	surname text	age integer	country text	player_position text	is_leader boolean	games_played integer	goals_scored integer	assists integer	saves integer
17	17	1	Muhammad	Torres	24	Portugal	Forward	false	12	5	3	0
18	18	1	James	Schmidt	34	Finland	Forward	false	8	7	2	0
19	19	1	Ronald	Hill	18	Latvia	Forward	false	9	6	2	0
20	20	1	Daniel	Nowak	29	Albania	Forward	false	10	3	0	0
21	21	2	Jan	Lewis	28	Australia	Goalkeeper	false	6	0	0	15
22	22	2	Andriy	Petit	28	Finland	Goalkeeper	false	5	0	0	14
23	23	2	Theodore	Taylor	26	Moldova	Goalkeeper	false	6	0	0	10
24	24	2	Antonio	Jones	24	Albania	Defender	false	13	0	0	6
25	25	2	Arthur	Murhy	31	New Zeland	Defender	false	14	0	2	2
26	26	2	Joseph	Hill	32	Luxembourg	Defender	false	10	0	0	3
27	27	2	Thomas	White	28	Sweden	Defender	false	10	0	0	2
28	28	2	Peter	Williams	24	Estonia	Defender	false	6	0	0	4
29	29	2	Riccardo	Wilson	32	Austria	Defender	false	15	0	0	4
30	30	2	Oscar	Clark	21	Russia	Midfielders	false	13	0	1	0
31	31	2	Eric	Lopez	27	Ireland	Midfielders	false	10	1	0	0
32	32	2	Jan	Bruno	19	Poland	Midfielders	false	12	1	1	1

Рис. 5.9 Заповнення таблиці player

	game_id integer	player_id integer
1	1	201
2	1	205
3	1	207
4	1	208
5	1	206
6	1	214
7	1	213
8	1	215
9	1	211
10	1	218
11	1	219
12	1	22
13	1	29
14	1	27
15	1	25
16	1	24
17	1	33
18	1	34

Рис. 5.10 Заповнення таблиці players_in_games

	team_id integer	place integer	games_played integer	score integer	wins integer	draws integer	losses integer	goals_scored integer	goals_conceded integer	to_champions_league boolean	to_lower_league boolean
1	9	1	19	39	11	6	2	34	17	true	false
2	7	2	22	36	10	6	6	40	34	true	false
3	1	3	24	35	10	5	9	38	39	true	false
4	3	4	19	31	9	4	6	26	21	false	false
5	10	5	19	31	9	4	6	25	24	false	false
6	15	6	19	29	8	5	6	27	26	false	false
7	8	7	18	27	7	6	5	22	16	false	false
8	11	8	22	27	8	3	11	32	32	false	false
9	13	9	21	27	8	3	10	22	24	false	false
10	19	10	20	26	7	5	8	29	25	false	false
11	5	11	20	26	7	5	8	30	31	false	false
12	12	12	22	26	6	8	8	26	30	false	false
13	4	13	16	24	6	6	4	17	16	false	false
14	16	14	15	23	6	5	4	21	22	false	false
15	14	15	19	21	6	3	10	27	34	false	false
16	18	16	17	20	5	5	7	24	27	false	false
17	6	17	20	20	6	2	12	17	25	false	false
18	20	18	16	19	5	4	7	23	26	false	true

Рис. 5.11 Заповнення таблиці results_table

	stadium_id [PK] integer	home_team integer	city text	stadium_name text	capacity integer
1	1	1	Brighton	Etihad Stadium	33412
2	2	2	Coventry	Stade de France	42510
3	3	3	Sunderland	Lusail Iconic	30920
4	4	4	Southampton	Stadio Olimpico	40235
5	5	5	Manchester	Groupama	24285
6	6	6	London	Emirates	39876
7	7	7	Leeds	Allianz Arena	25211
8	8	8	Madrid	Old Trafford	41123
9	9	9	Manchester	Santiago Bernabeu	51427
10	10	10	Birmingham	Esprit Arena	42930
11	11	11	Madrid	Olimpiyskiy	40944
12	12	12	Leicester	Veltins-Arena	23157
13	13	13	Manchester	Stade Velodrome	47249
14	14	14	Madrid	MetLife	27864
15	15	15	Bristol	Luzhniki	48231
16	16	16	Norwich	Estadio da Luz	47941
17	17	17	Nottingham	Rose Bowl	28213
18	18	18	Southampton	Commonwealth	52844

Рис. 5.12 Заповнення таблиці stadium

	 team_id [PK] integer 	team_name text 	coach_name text 	previous_place integer 
1	1	Southampton	Kevin Costa	1
2	2	Norwich	Henri Lopez	2
3	3	Everton	Henri Bernand	3
4	4	Liverpool	Ronald Rivera	4
5	5	Brentford	Oscar Harris	5
6	6	Wolverhampton	Lukas Lee	6
7	7	Man United	Peter Torres	7
8	8	Leeds	Luka Rossi	8
9	9	West Ham	Theodore Johnson	9
10	10	Tottenham	Mark Johnson	10
11	11	Aston Villa	Benjamin Meyer	11
12	12	Burnley	Richard Weber	12
13	13	Arsenal	Benjamin Williams	13
14	14	Chelsea	Leonardo Bruno	14
15	15	Leicester	Artem Sousa	15
16	16	Watford	Leo Silva	16
17	17	Crystal	Luka Martinez	17

Рис. 5.13 Заповнення таблиці team

5.3 Формування запитів пошуку даних у створеній базі даних національної першості з футболу

Користувацький інтерфейсів для роботи з запитами пошуку (SELECT)

Для зручної взаємодії з базою даних, виконання запитів та структуризації отриманих даних було створено інтуїтивно зрозумілий користувацький інтерфейс у застосунку, написаному на об'єктно-орієнтованій мові програмування C# та засобах графічної розмітки XAML(WPF).

На рис. 5.14 можна побачити програму, що складається з текст-блоку для введення необхідного SQL-запиту, панелі типових запитів, таблиці результатів (для запитів типу SELECT), кнопки виконання запиту та закриття застосунку (детальніше ознайомитись з особливостями XAML розмітки можна у файлі «Вікно застосунку для роботи з базою даних.xaml»).

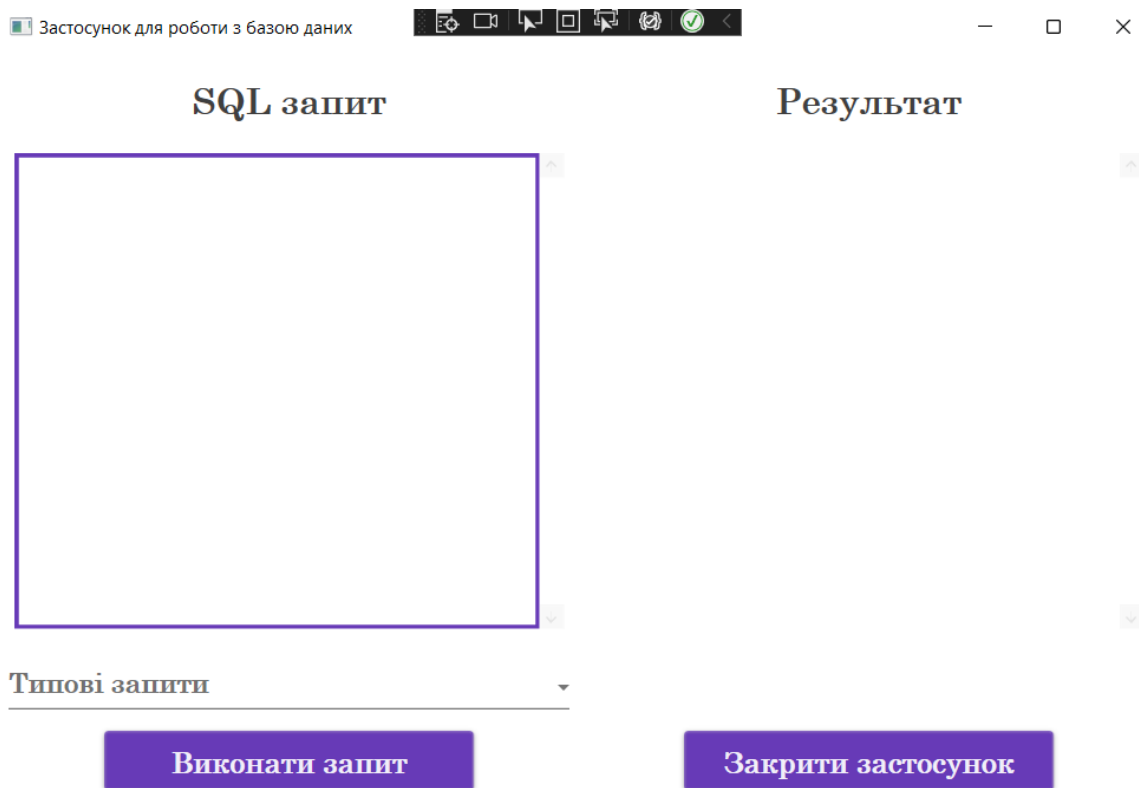


Рис. 5.14 Простий програмний застосунок для роботи з базою даних

Формування типових SQL запитів пошуку в інформаційній системі національної першості з футболу

Отримання інформації про ігри певної команди

```
SELECT *
FROM game
WHERE home_team =
(
SELECT team_id
FROM team
WHERE team_name = 'Уведіть назву команди'
)
OR guest_team =
(
SELECT team_id
FROM team
WHERE team_name = 'Уведіть назву команди ще раз'
);
```

Отримання інформації про гравців, що грали у певній грі

```
SELECT *
FROM player
WHERE player_id =
(
SELECT player_id
FROM players_in_games
WHERE game_id =
(
SELECT game_id
FROM game
WHERE home_team =
(
SELECT team_id
FROM team
WHERE team_name = 'Уведіть назву команди-хозяйки'
)
AND guest_team =
(
SELECT team_id
FROM team
WHERE team_name = 'Уведіть назву команди-гостя'
)
)
);
```

Визначення ціни квитка на певний матч

```
SELECT ticket_price
FROM game
WHERE home_team =
(
    SELECT team_id
    FROM team
    WHERE team_name = 'Уведіть назву команди-хозяйки'
)
AND guest_team =
(
    SELECT team_id
    FROM team
    WHERE team_name = 'Уведіть назву команди-гостя'
);
```

Інформація про гравця, що забив найбільше голів

```
SELECT *
FROM player
WHERE goals_scored =
(
    SELECT MAX(goals_scored)
    FROM player
);
```

Інформація про різницю забитих-пропущених м'ячів для певної команди

```
SELECT *
FROM team
WHERE team_id =
(
    SELECT team_id
    FROM results_table
    WHERE goals_scored-goals_conceded =
    (
        SELECT MAX(goals_scored-goals_conceded)
        FROM results_table
    )
);
```

Визначення наймолодших гравців Ліги

```
SELECT *
FROM player
WHERE age =
(
    SELECT MIN(age)
    FROM player
);
```

Інформація про команди, що проходять до Ліги Чемпіонів (перші 3 команди у таблиці)

```
SELECT *
FROM team
WHERE team_id =
(
    SELECT team_id
    FROM results_table
    WHERE to_champions_league = true
);
```

Визначення команди з найбільшим прибутком

```
SELECT *
FROM team
WHERE team_id =
(
    SELECT team_id
    FROM budget
    WHERE profit =
        (
            SELECT MAX(profit)
            FROM budget
        )
);
```

Слід зазначити, що хоча певні SQL-запити типу SELECT не потребують додаткового введення даних, серед інших є ті, у які необхідно вписати інформацію для уточнення (зазвичай, це назва команди). Ці пропуски помічені одинарними лапками та українським текстом всередині та починаються зі слова «Уведіть».

Результат виконання певних команд запиту SELECT

На рис. 5.15 – 5.18 показані результати виконання деяких типових запитів до бази даних національної першості країн з футболу

SQL запит

```
SELECT *  
FROM player  
WHERE age =  
(  
  SELECT MIN(age)  
  FROM player  
);
```

Результат

firstname	surname
Ronald	Hill
Louis	Lopez
Jan	Petit
Peter	Petit
Thomas	Wright
Artem	Turner
Oliver	Petit
Kevin	Brown
Peter	Costa

Наймолодші гравці

Виконати запит

Закрити застосунок

Рис. 5.15 Результат виконання типової команди пошуку наймолодших гравців

SQL запит

```
SELECT *  
FROM team  
WHERE team_id =  
(  
  SELECT team_id  
  FROM budget  
  WHERE profit =  
(  
    SELECT MAX(profit)  
    FROM budget  
  )  
)
```

Результат

teamid	teamname
2	Norwich

Команда з найбільшим прибутком

Виконати запит

Закрити застосунок

Рис. 5.16 Результат виконання типового запиту пошуку команди з найбільшим загальним прибутком

Застосунок для роботи з базою даних

SQL запит

SELECT ticket_price
FROM game
WHERE home_team =
(
SELECT team_id
FROM team
WHERE team_name = 'Leeds'
)
AND guest_team =
(
SELECT team_id
FROM team
WHERE team_name = 'Chelsea'
);

Результат

ticketprice
39

Ціна квитка на певний матч

Виконати запит

Закрити застосунок

Рис. 5.17 Результат виконання типового запиту визначення ціни на квиток на певний матч (матч визначається через назву команди-господаря та команди-гостя)

Застосунок для роботи з базою даних

SQL запит

SELECT *
FROM game
WHERE home_team =
(
SELECT team_id
FROM team
WHERE team_name = 'Leeds'
)
OR guest_team =
(
SELECT team_id
FROM team
WHERE team_name = 'Chelsea'
);

Результат

gamedate	gamestarted
9/7/2022 12:00:00 AM	14:00:00
9/10/2022 12:00:00 AM	19:15:00
9/15/2022 12:00:00 AM	20:00:00
9/15/2022 12:00:00 AM	21:00:00
9/17/2022 12:00:00 AM	14:45:00
9/24/2022 12:00:00 AM	18:15:00
9/26/2022 12:00:00 AM	21:45:00
10/2/2022 12:00:00 AM	18:45:00
10/18/2022 12:00:00 AM	15:00:00

Інформація про ігри певної команди

Виконати запит

Закрити застосунок

Рис. 5.18 Результат виконання типового запиту на визначення загальної інформації гри між певними двома командами

Висновок

Курсова робота з теми «Інформаційна система першості країн з футболу» демонструє можливість створення бази даних, її наповнення, модифікацію та обробку. Сама база даних була розроблена під реляційну СУБД PostgreSQL (що виступає середовищем її зберігання та організації) на мові запитів SQL. Окрім цього, дослідницька робота передбачає розроблення програмного застосунку з користувацьким інтерфейсом на мові програмування C# та XAML у середовищі програмної розробки Visual Studio 2022 від Microsoft.

Подібна база даних може мати прикладне призначення як для організації відповідних даних стосовно національної першості з футболу, так використовуватись у вигляді скелету для різних ігрових симуляторів футболу.

Для коректного заповнення бази (INSERT та UPDATE) даних відбувається її попередня валідація у спеціальному програмному застосунку.

Для виконання запитів пошуку (SELECT) в інформаційній системі національної першості з футболу було створено окрему користувацьку програму з графічним інтерфейсом та типовими запитам.

Усі функціональні компоненти проекту протестовані та працюють відповідно до завдання курсової роботи.

Список використаних джерел

1. Microsoft C# documentation, [<https://docs.microsoft.com/en-us/dotnet/csharp/>]
2. Microsoft SQL documentation, [<https://docs.microsoft.com/en-us/sql/?view=sql-server-ver16>]
3. Beaulieu, Alan Learning. М.: O'Reilly Media, 2020. SQL 384 с.
4. Sells, Chris Programming WPF: Building Windows UI with Windows Presentation Foundation. М.: O'Reilly Media, 2007. 864 с.

Додаток А

Повний перелік запитів на заповнення таблиць бази даних інформаційної системи національної першості з футболу

Додавання даних до таблиці team

```
INSERT INTO team
(
    team_name,
    coach_name,
    previous_place
)
VALUES
(
    @team_name,
    @coach_name,
    @previous_place
);
```

Додавання даних до таблиці stadium

```
INSERT INTO stadium
(
    home_team, city,
    stadium_name, capacity
)
VALUES
(
    (
        SELECT team_id
        FROM team
        WHERE team_name = @team_name
    ),
    @city, @stadium_name,
    @capacity
);
```

Додавання даних до таблиці budget

```
INSERT INTO budget
(
    team_id,
    cash, advertising,
    tickets_sold,
    tickets_income,
    selling_players, income,
    salary, buying_players,
    expenses, profit
)
```

```

VALUES
(
(
SELECT team_id
FROM team
WHERE team_name = @team_name
),
@cash, @advertising,
@tickets_sold, @tickets_income,
@selling_players, @income,
@salary, @buying_players,
@expenses, @profit
);

```

Додавання даних до таблиці game

```

INSERT INTO game
(
stadium_id,
home_team,
guest_team, weather,
game_date, game_started, game_ended,
referee_name, game_result,
visitors, ticket_price
)
VALUES
(
(
SELECT stadium_id
FROM stadium
WHERE home_team =
(
SELECT team_id
FROM team
WHERE team_name = @home_team
)
),
(
SELECT team_id
FROM team
WHERE team_name = @home_team
),
(
SELECT team_id
FROM team
WHERE team_name = @guest_team
),
@weather,
@game_date, @game_started, @game_ended,
@referee_name, @game_result,
@visitors, @ticket_price
);

```

Додавання даних до таблиці players_in_games

```
INSERT INTO players_in_games
(
    game_id,
    player_id
)
VALUES
(
    (
        SELECT game_id
        FROM game
        WHERE home_team =
        (
            SELECT team_id
            FROM team
            WHERE team_name = @home_team
        )
        AND guest_team =
        (
            SELECT team_id
            FROM team
            WHERE team_name = @guest_team
        )
    ),
    @player_id
);
```

Додавання даних до таблиці player

```
INSERT INTO player
(
    team_id, first_name, surname, age,
    country, player_position, is_leader,
    games_played, goals_scored, assists, saves,
    salary, player_cost, contract_due
)
VALUES
(
    (
        SELECT team_id
        FROM team
        WHERE team_name = @team_name
    ),
    @first_name, @surname, @age,
    @country, @player_position, @is_leader,
    @games_played, @goals_scored, @assists, @saves,
    @salary, @player_cost,
    CAST(@contract_due as DATE)
);
```

Додавання даних до таблиці results_table

```
INSERT INTO results_table
(
    team_id, place, score, games_played,
    wins, draws, losses,
    goals_scored, goals_conceded,
    to_champions_league, to_lower_league
)
VALUES
(
    (
        SELECT team_id
        FROM team
        WHERE team_name = @team_name
    ),
    @place, @score, @games_played,
    @wins, @draws, @losses,
    @goals_scored, @goals_conceded,
    @to_champions_league, @to_lower_league
);
```

Додаток Б

Повний перелік запитів на оновлення таблиць бази даних інформаційної системи національної першості з футболу

Оновлення даних таблиці game

```
UPDATE game
SET stadium_id =
(
  SELECT stadium_id
  FROM stadium
  WHERE home_team =
    (
      SELECT team_id
      FROM team
      WHERE team_name = @home_team
    )
),
home_team =
(
  SELECT team_id
  FROM team
  WHERE team_name = @home_team
),
guest_team =
(
  SELECT team_id
  FROM team
  WHERE team_name = @guest_team
),
game_date = @game_date,
game_started = @game_started,
game_ended = @game_ended,
weather = @weather,
referee_name = @referee_name,
game_result = @game_result,
visitors = @visitors,
ticket_price = @ticket_price
WHERE home_team =
(
  SELECT team_id
  FROM team
  WHERE team_name = @home_team
)
AND guest_team =
(
  SELECT team_id
  FROM team
  WHERE team_name = @guest_team
);
```


Оновлення даних таблиці budget

```
UPDATE budget
SET team_id =
(
    SELECT team_id
    FROM team
    WHERE team_name = @team_name
),
cash = @cash,
tickets_sold = @tickets_sold,
tickets_income = @tickets_income,
advertising = @advertising,
selling_players = @selling_players,
salary = @salary,
buying_players = @buying_players,
expenses = @expenses,
profit = @profit,
income = @income
WHERE team_id =
(
    SELECT team_id
    FROM team
    WHERE team_name = @team_name
);
```

Оновлення даних таблиці results_table

```
UPDATE results_table
SET team_id =
(
    SELECT team_id
    FROM team
    WHERE team_name = @team_name
),
place = @place,
score = @score,
games_played = @games_played,
wins = @wins,
draws = @draws,
losses = @losses,
goals_scored = @goals_scored,
goals_conceded = @goals_conceded,
to_champions_league = @to_champions_league,
to_lower_league = @to_lower_league
WHERE team_id =
(
    SELECT team_id
    FROM team
    WHERE team_name = @team_name
);
```

Оновлення даних таблиці player

```
UPDATE player
SET games_played =
(
  SELECT games_played
  FROM player
  WHERE player_id = @player_id
) + 1
WHERE player_id = @player_id;
```

Оновлення даних таблиці player

```
UPDATE player
SET goals_scored =
(
  SELECT goals_scored
  FROM player
  WHERE player_id = @player_id
) + 1
WHERE player_id = @player_id;
```

Оновлення даних таблиці player

```
UPDATE player
SET assists =
(
  SELECT assists
  FROM player
  WHERE player_id = @player_id
) + 1
WHERE player_id = @player_id;
```

Оновлення даних таблиці player

```
UPDATE player
SET saves =
(
  SELECT saves
  FROM player
  WHERE player_id = @player_id
) + 1
WHERE player_id = @player_id;
```