

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

Київ – 2022

**ЗМІСТ**

[**Завдання до курсової роботи** 3](#_Toc106877156)

[**Вступ** 5](#_Toc106877157)

[**1. Аналіз предметної області** 7](#_Toc106877158)

[**2. Функціональне моделювання інформаційної системи роботи** 8](#_Toc106877159)

[**кінотеатрів міста з використанням DFD-діаграм** 8](#_Toc106877160)

[**3. Концептуальна модель бази даних** 10](#_Toc106877161)

[**4. Фізична модель бази даних системи інформаційної служби національної першості з футболу** 13](#_Toc106877162)

[**5. Опис реалізації завдань відповідно до обраного варіанту** 18](#_Toc106877163)

[**5.1. Створення та структура таблиць бази даних інформаційної** 18](#_Toc106877164)

[**системи національної першості з футболу** 18](#_Toc106877165)

[**5.2 Формування запитів додавання даних до створеної бази даних** 21](#_Toc106877166)

[**5.3 Формування запитів пошуку даних у створеній базі даних національної першості з футболу** 30](#_Toc106877167)

[**Висновок** 37](#_Toc106877168)

[**Список використаних джерел** 38](#_Toc106877169)

# **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 (поточний баланс, доходи, видатки та кінцевий профіт, що рахуються з результатів матчу).

# **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 c# 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 первнних 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 роботи**

# **кінотеатрів міста 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).

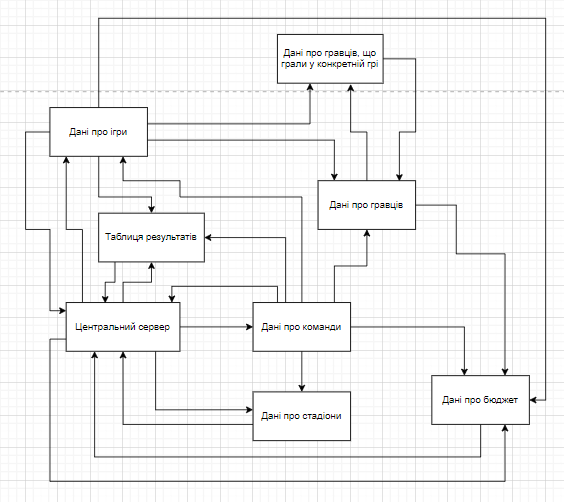


Рис. 2.1 Діаграма потоків даних інформаційної системи національної першості з футболу

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 ігр гравців, конкретні матчі та бюджет клубу

Як результат, для забезпечення функціональних можливостей системи національної першості з футболу згодиться будь-яка операційна система, що підтримує х32 та х64 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.

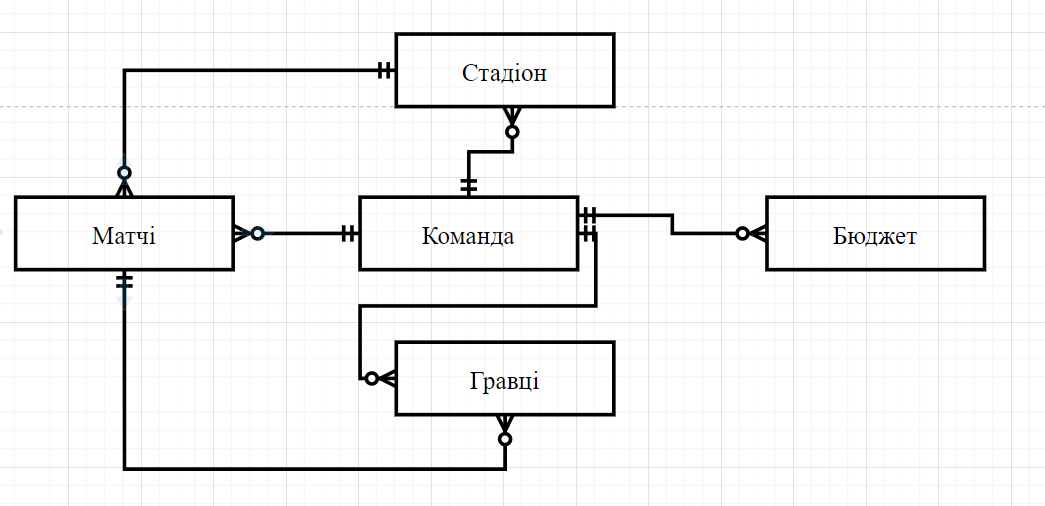


Рис. 3.1 ER-діаграма інформаціної системи бази даних національної першості з футболу

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

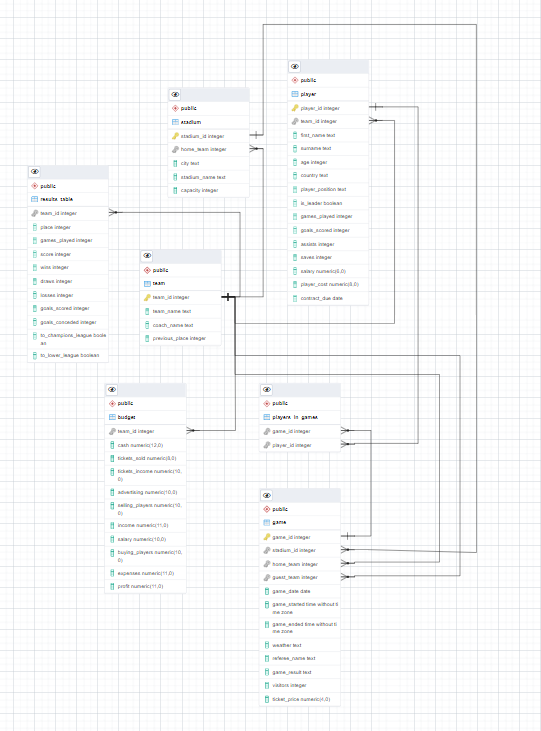
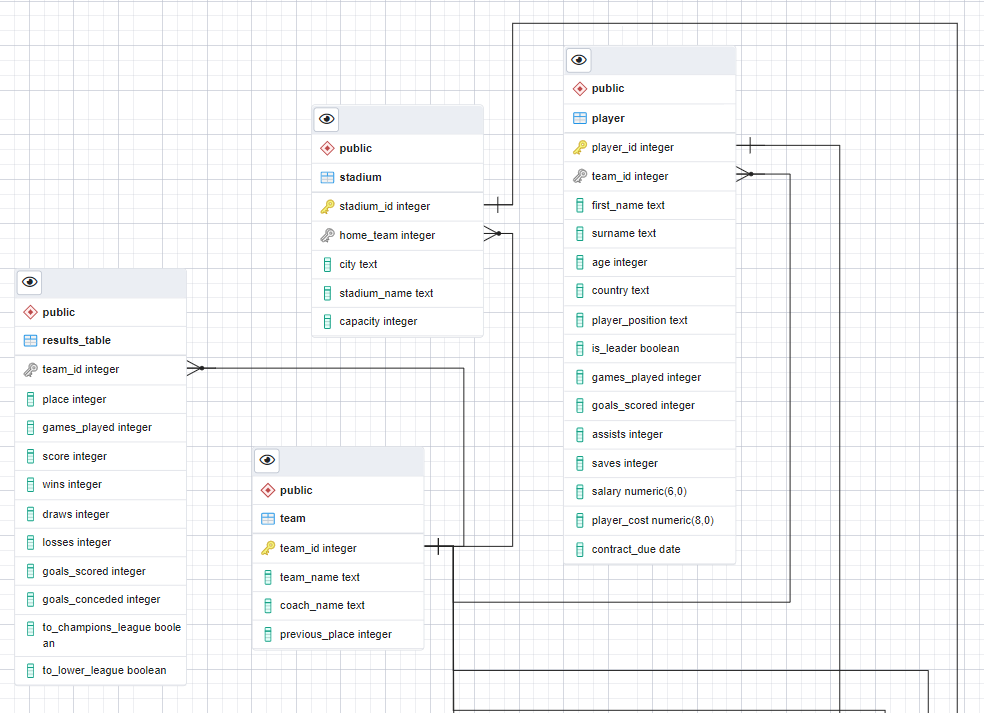
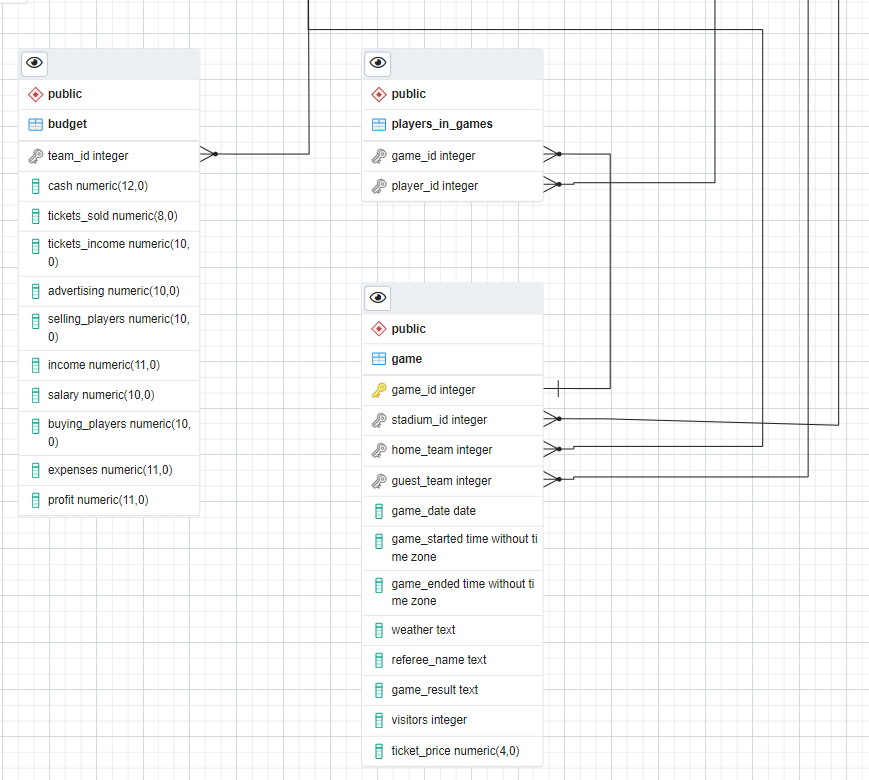


Рис. 3.2 Загальний план



*Рис. 3.3 загальний план, верхня частина*



*Рис. 3.4 загальний 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**

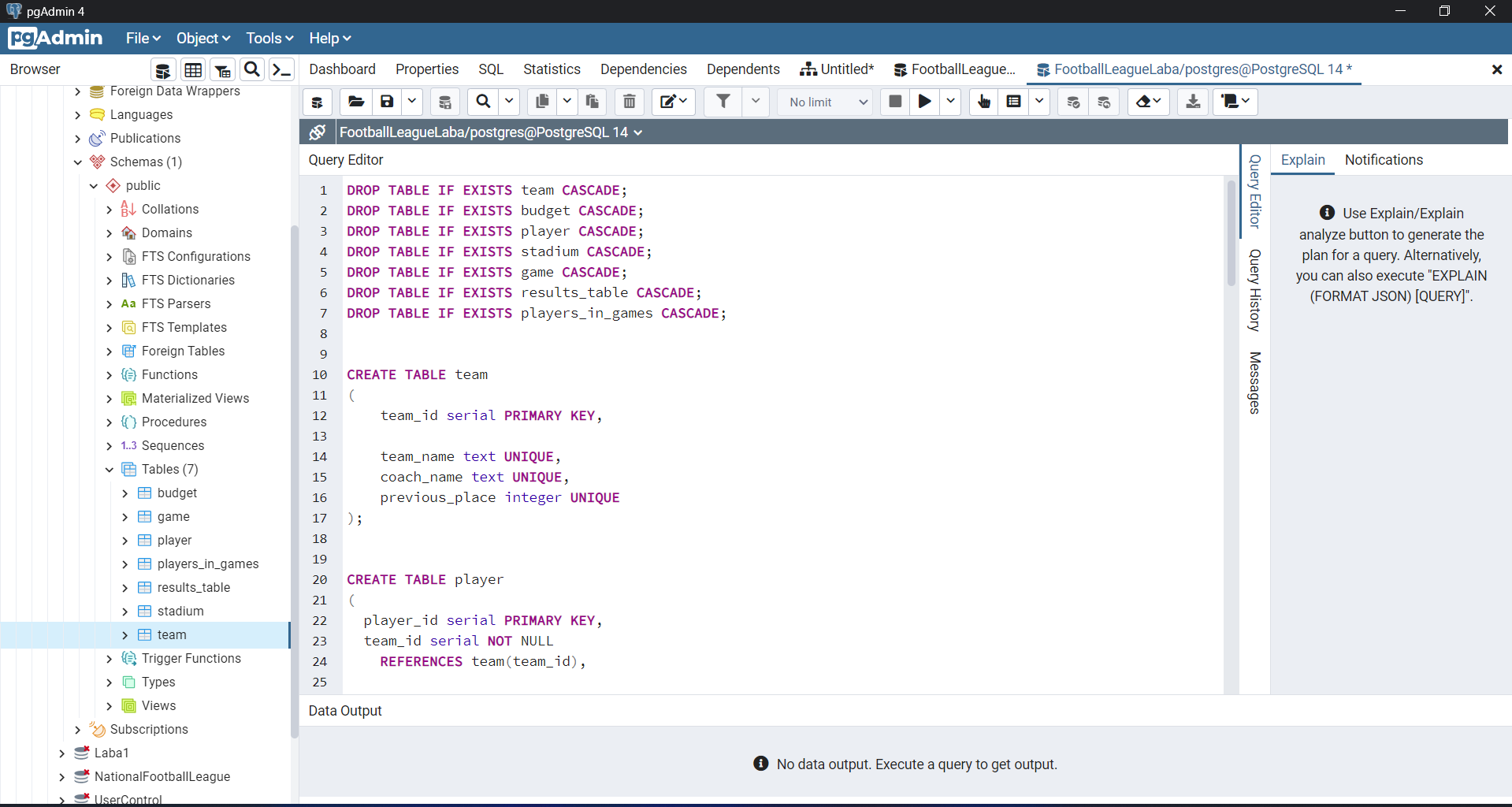
**REFERENCES game(game\_id),**

**player\_id serial NOT NULL**

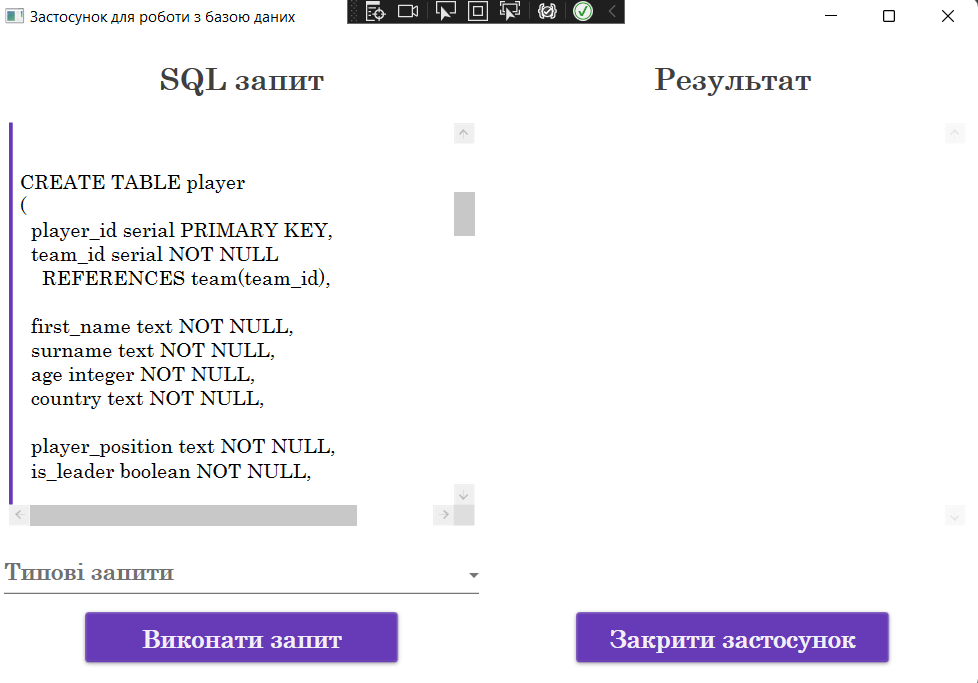
**REFERENCES player(player\_id)**

**);**

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#(WPF)*

# **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 довідкової

служби роботи кінотеатрів 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 |
| saves | 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 |

Таблиця 5.3

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 |

Таблиця 5.4

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 |

Таблиця 5.5

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 |

Таблиця 5.6

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 |

Таблиця 5.7

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 (на приклад, 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 C# (код до програми можна побачити у папці «Заповнення бази даних), що знаходиться у тій же комірці, що і звіт.

Була здійснена валідаці 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 кількісті 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 , де – кількість очок, w - кількість перемог, d – кількість нічиїх

**Таблиця, що поєднує гравців та ігри**

* Містить по 22 гравця на кожну гру (по 11 з кожної сторони, без замін)

**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

);

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

);

Аналогічно прописані запити на додавання даних до таблиць game, players\_in\_games, player та results\_table.

**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

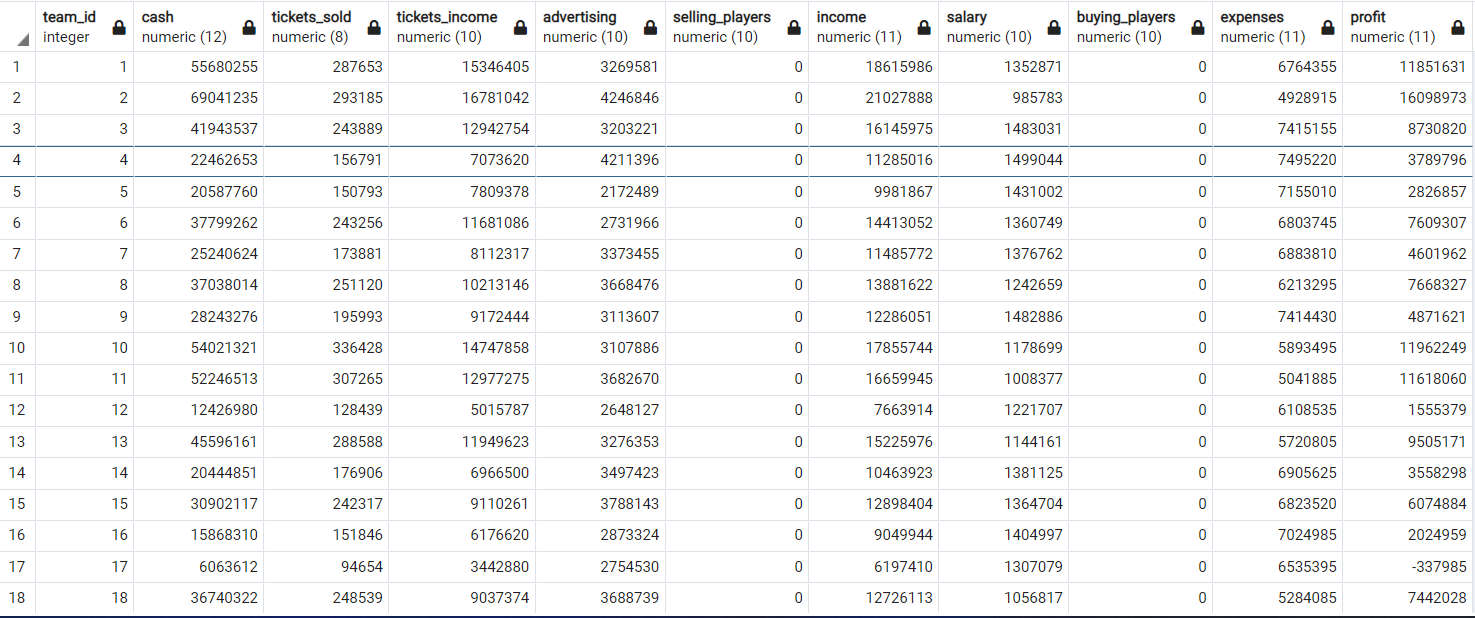
WHERE team\_name = @guest\_team

);

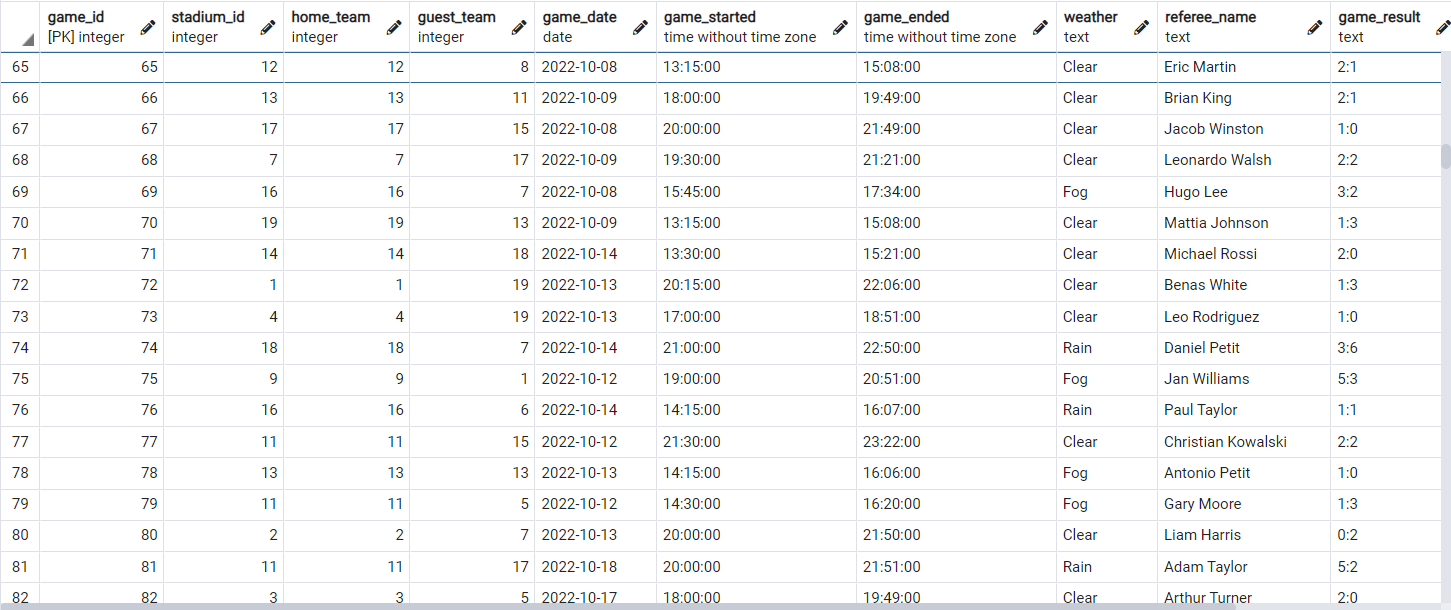
Аналогічно прописані запити на оновлення даних інших таблиць.

**Результат запитів**

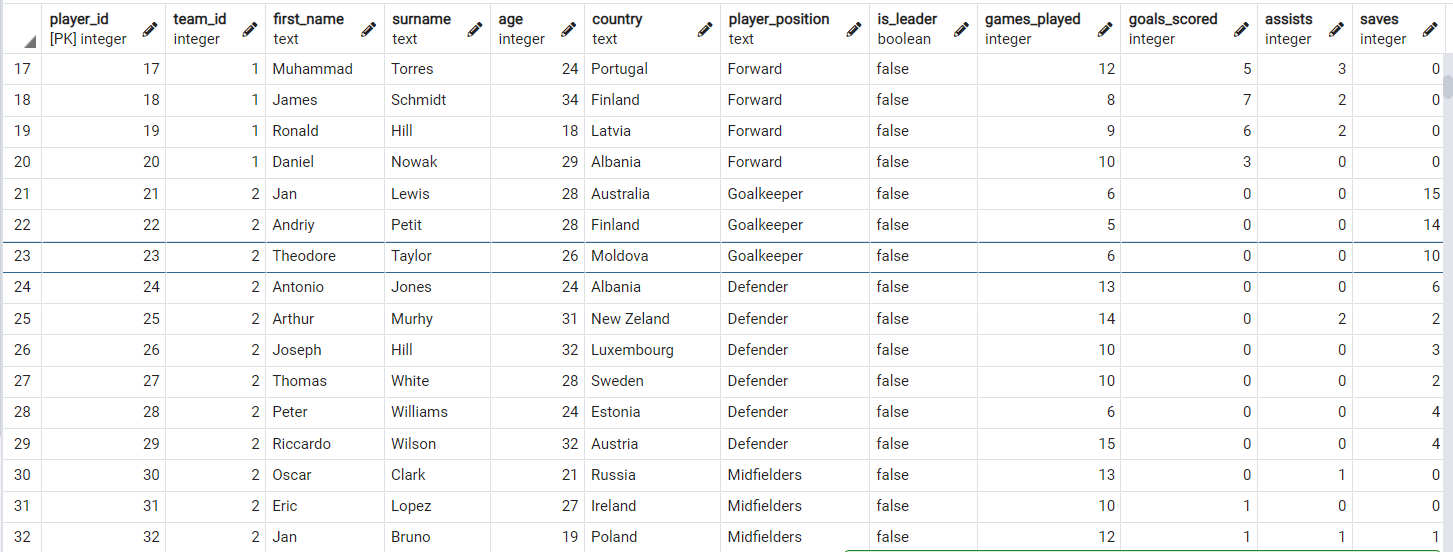
Як результат, маємо заповнення бази даних (її таблиць) відповідними значеннями, що зображено на рис. 5.2.1 – 5.2.



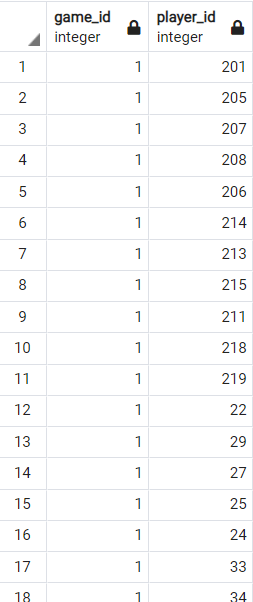
*Рис. 5.2.1 Заповнення таблиці budget*



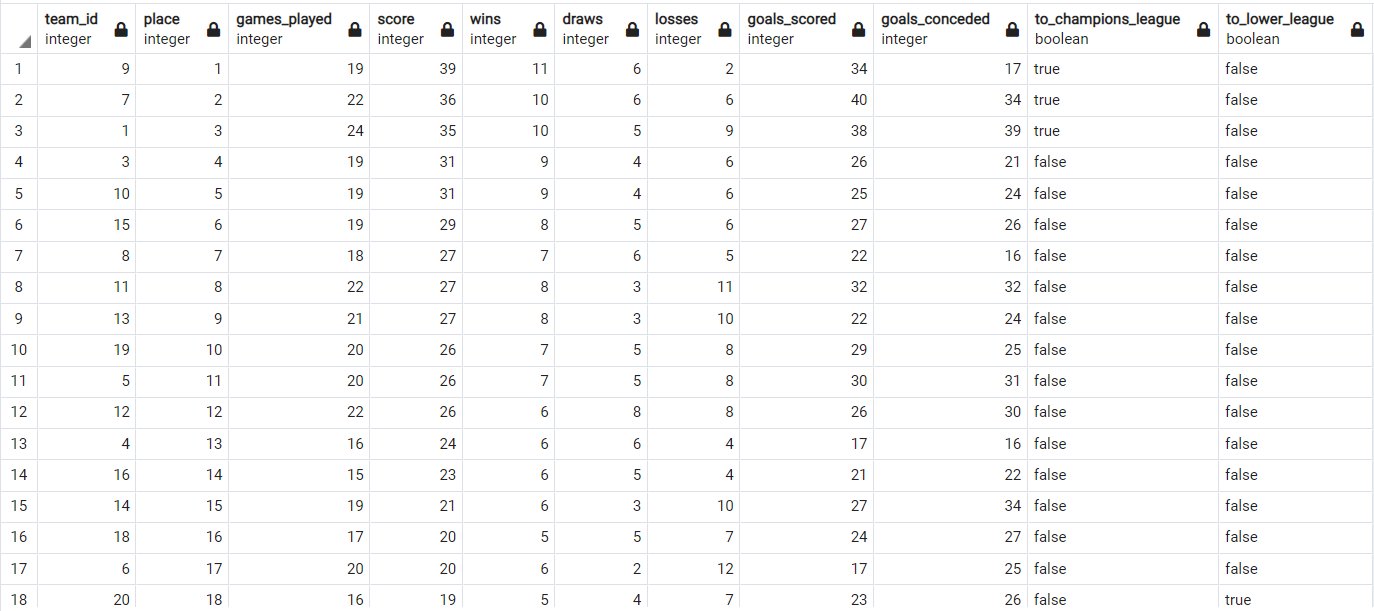
*Рис. 5.2.2 Заповнення таблиці game*



*Рис. 5.2.3 Заповнення таблиці player*



*Рис. 5.2.4 Заповнення таблиці players\_in\_games*



*Рис. 5.2.5 Заповнення таблиці results\_table*



*Рис. 5.2.6 Заповнення таблиці stadium*



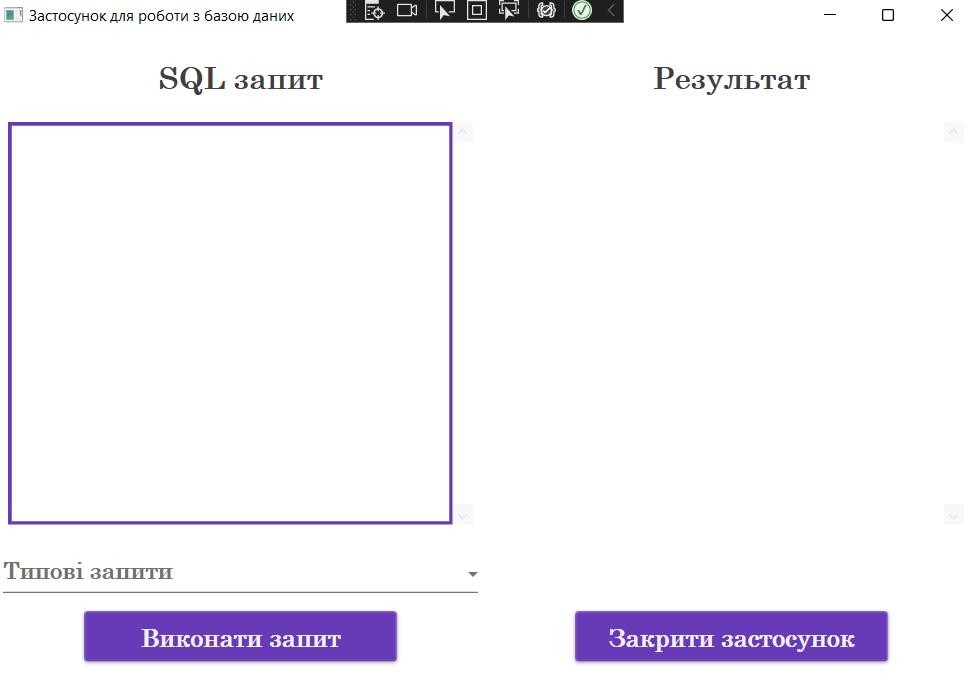
*Рис. 5.2.7 Заповнення таблиці team*

## **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 структуризуції отриманих даних було створено інтуітивно зрозумілий користувацький інтерфейс у застосунку, написаному на об’єктно-орієнтованій мові програмування C# та засобах графічної розмітки XAML(WPF).

На рис. 5.3.1 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. 5.3.1 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

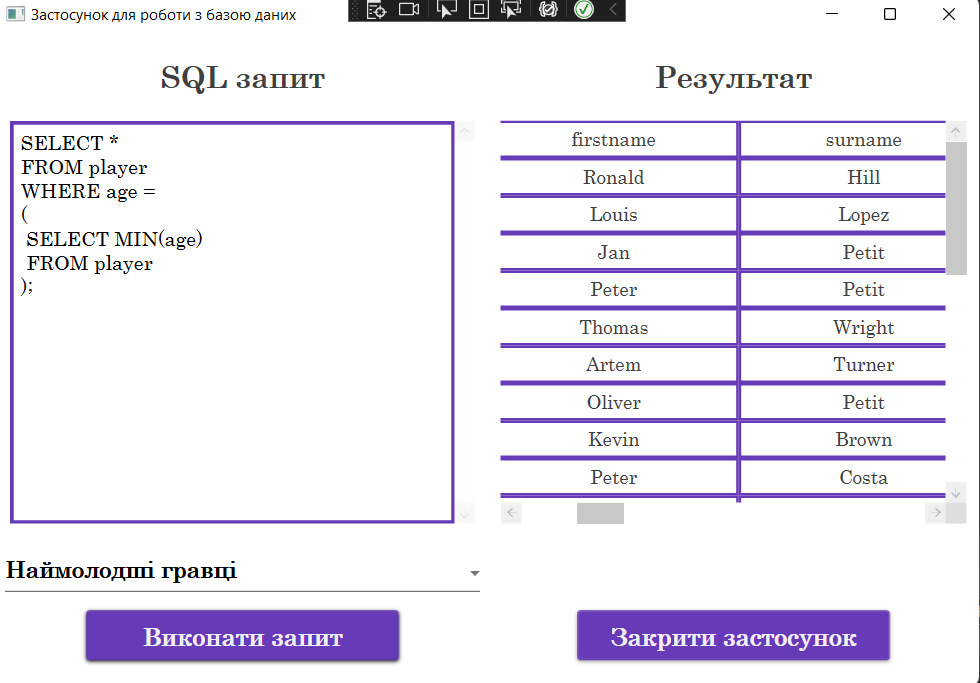
)

);

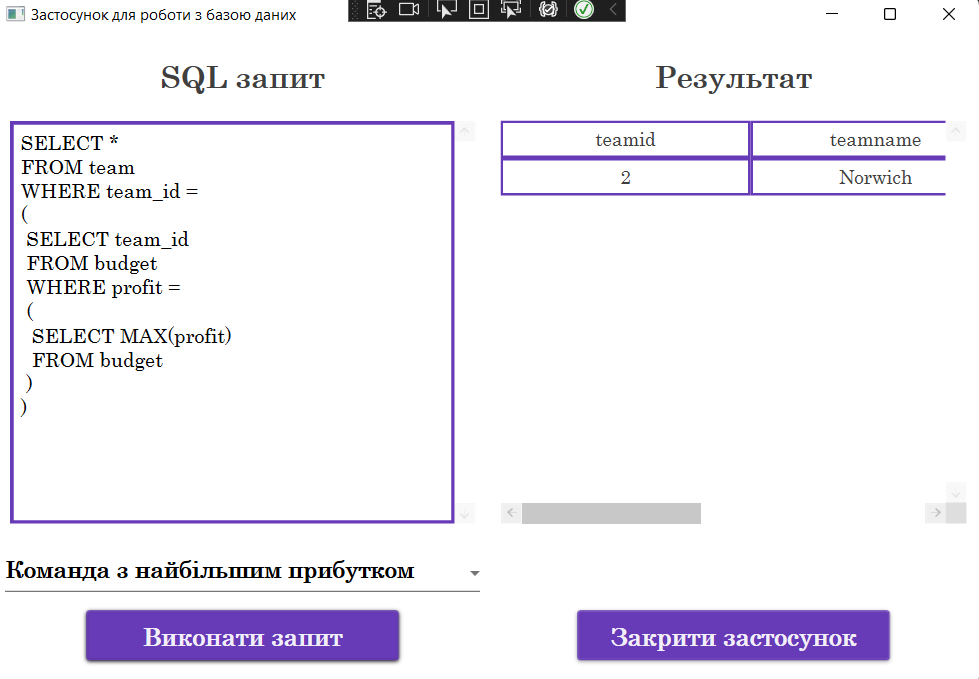
Слід зазначити, що хоча певні SQL-запити типу SELECT не потребують додаткового введення даних, серед інших є ті, у які необхідно вписати інформацію для уточнення (зазвичай, це назва команди). Ці пропуски помічені одинарними лапками та українським текстом всередині та починаються зі слова «Уведіть».

**Результат виконання певних команд запиту SELECT**

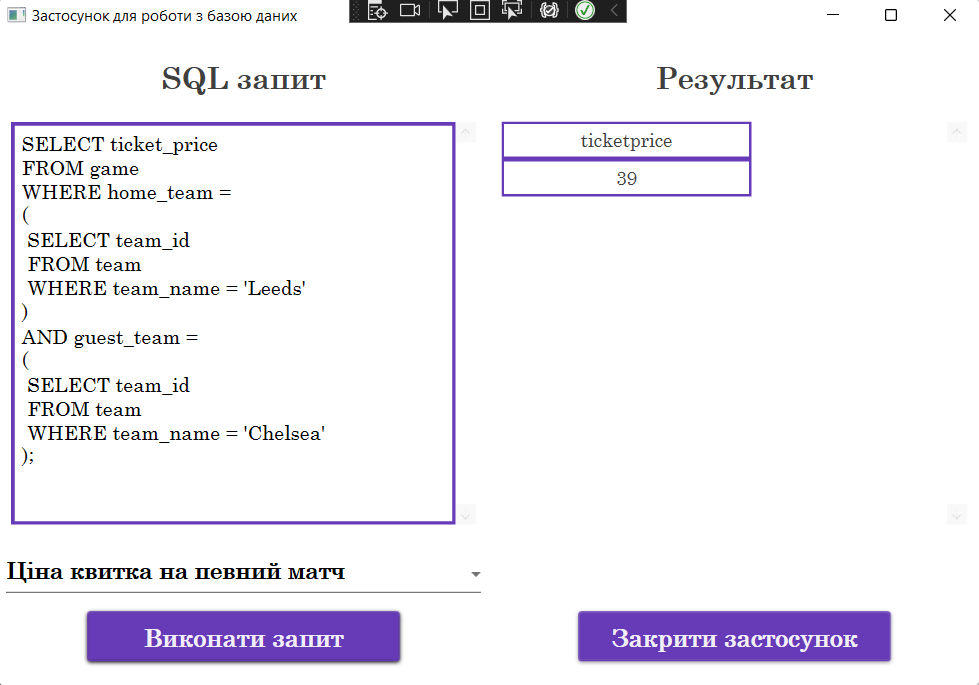
На рис. 5.3.2 – 5.3.5 показані результати виконання деяких типових запитів до бази даних національної першості країн з футболу

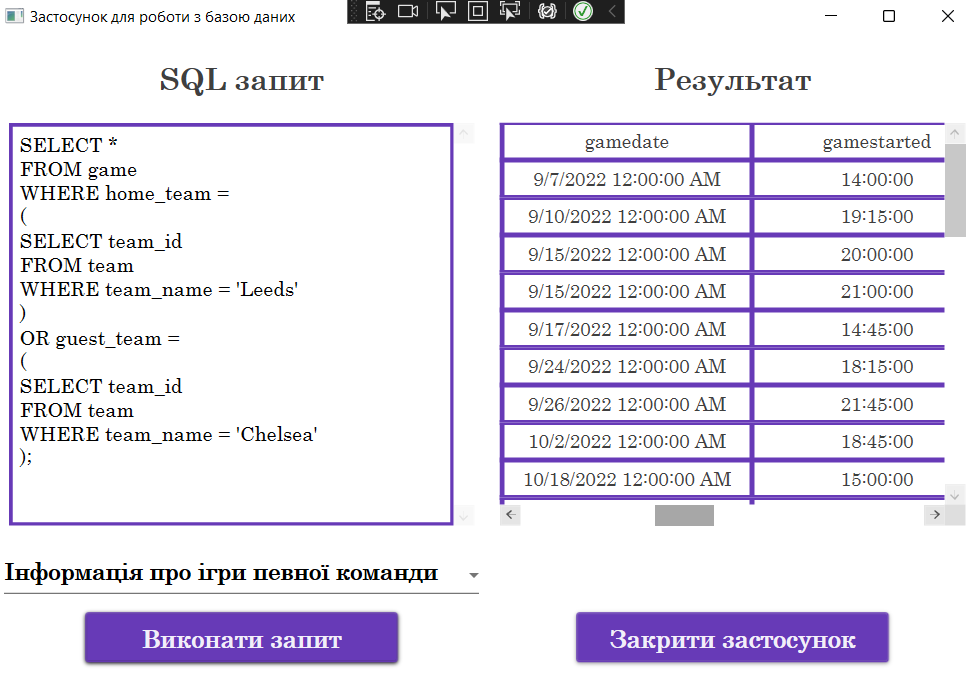


*Рис. 5.3.2 Результат виконання типової команди пошуку наймолодших гравців*



*Рис. 5.3.3 Результат виконання типового запиту пошуку команди з найбільшим загальним прибутком*

 *Рис. 5.3.4 Результат виконання типового запиту визначення ціни на квиток на певний матч (матч визначається через назву команди-господаря та команди-гостя)*



*Рис. 5.3.5 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 запитами.

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 c.

# 4. Sells, Chris Programming WPF: Building Windows UI with Windows Presentation Foundation. М.: O'Reilly Media, 2007. 864 с.