НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики Кафедра прикладної математики

Звіт

з лабораторної роботи №2

з дисципліни «Вступ до баз даних та інформаційних систем» на тему:

*Даталогічне проектування власної БД. Виконання запитів на мові SQL. Візуалізація запитів.*

Виконала: Керівник:

студентка групи КМ-92 *ст. викладач Бай Ю. П. Лутак М.В.*

Київ — 2021

*ЗМІСТ*

[*ЗАВДАННЯ 3*](#_Toc88142657)

[*Завдання 1 4*](#_Toc88142658)

[*Завдання 2 4*](#_Toc88142659)

[*Завдання 3 4*](#_Toc88142660)

[*Завдання 4 5*](#_Toc88142661)

[*Завдання 5 5*](#_Toc88142662)

[*Завдання 6 7*](#_Toc88142663)

[*Завдання 7 8*](#_Toc88142664)

[*Завдання 8 9*](#_Toc88142665)

[*Список літератури 11*](#_Toc88142666)

# ЗАВДАННЯ

1. Визначитися з тематикою власної БД, обрати датасет на kaggle або data.gov.ua. Перевірити, що обраної тематики ще немає у [Таблиці](https://docs.google.com/spreadsheets/d/1-VSnwiJ7IzvS08lIo8FOZ0fV2uOM6HFs3lL8cc6bFYY/edit?usp=sharing). Записати тематику та посилання на датасет у стовпчики *C, D*.
2. Визначити сутності, атрибути та зв’язки інфологічної моделі БД. *(Кількість сутностей та атрибутів повинна бути мінімально необхідною)*. Записати сутності та їх атрибути у стовпчик *E*.
3. Побудувати ERD реляційної бази даних, використовуючи Draw.io *(нотація для зв’язків між сутностями – “Crow’s Foot”).* Нормалізувати таблиці до 4NF включно. Оформити ERD бази даних у вигляді файлу *erd.drawio.* Назви таблиць БД після нормалізації записати у стовпчик *F*. *(Увага! Кількість таблиць повинна бути не менше 3-х і не більше 10)*.
4. Написати код для створення таблиць – *create.sql*.
5. Підготувати дані для заповнення таблиць, 5-10 рядків в кожну таблицю. Написати код для заповнення таблиць даними – populate.sql.
6. Сформулювати 3 запити (файл query.sql), що відповідають заданій структурі:
7. візуалізація – стовпчикова діаграма;
8. візуалізація – кругова діаграма;
9. візуалізація – графік залежності (в разі можна замінити іншою стовпчиковою діаграмою).

Короткий опис запитів записати у стовпчики G, H, I.

1. Написати код на Python, що реалізує підключення до PostgreSQL, виконує запити 6a, 6b, 6c та виводить результати їх виконання у консоль. Оформити даний код як файл *main.py*.
2. Реалізувати візуалізацію запитів 6a, 6b, 6c, побудувати dashboard та завантажити його на хостинг plotly. Зберегти файл як visualization.py. За бажанням можна виконати дане завдання за допомогою бібліотеки mathplotlib без використання plotly.
3. Створити проект *db\_lab2\_student01* на Github, в якому розташувати файли:

* *erd.drawio*
* *create.sql*
* *populate.sql*
* *query.sql*
* *main.py*
* *visualization.py*

та додати посилання на проект у стовпчик *J*. (***student01*** *=* ***Прізвище студента*** *латиницею*).

# Завдання 1

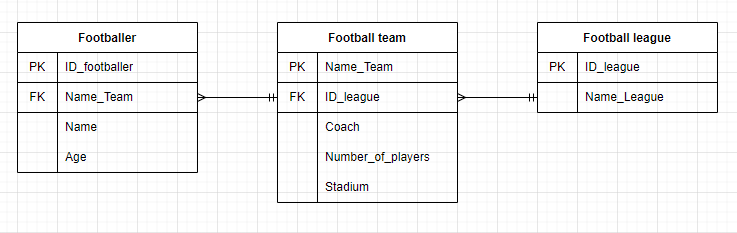
Для тематики власної БД було обрано датасет FIFA 22 Complete Player Dataset, що розташований за посиланням:

<https://www.kaggle.com/cashncarry/fifa-22-complete-player-dataset?select=teams_fifa22.csv>

# Завдання 2

Було обрано три сутності: Футболіст, Команда та Ліга. Сутність Футболіст має такі атриибути: ID, ім’я, вік та футбольну команду. Сутність Команда має наступні атрибути: назву, тренера, ID ліги, кількість гравців та стадіон, що закріплений за командою. Ліга має ID та назву.

# Завдання 3



Дана таблиця знаходиться в 1NF так, як жоден з її рядків не містить у будь-якому своєму полі більше одного значення і жодне з її ключових полів не порожнє. Таблиця перебуває в 2NF так, як ключ відношення в 1NF є простим (не композитним). Таблиця нормована до 3NF так, як дані в таблиці залежать лише від основного ключа, тобто немає транзитивних залежностей між неключовими полями. Таблиця також нормована і до 4NF, бо не містить нетривіальних багатозначних залежностей.

# Завдання 4

**CREATE** **TABLE** footballer(

id\_footballer **INTEGER** **NOT** **NULL**,

name\_team **CHAR**(50) **NOT** **NULL**,

name\_footballer **CHAR**(50) **NOT** **NULL**,

age **INTEGER**

);

**CREATE** **TABLE** football\_team(

name\_team **CHAR**(50) **NOT** **NULL**,

id\_league **INTEGER** **NOT** **NULL**,

coach **CHAR**(50) **NOT** **NULL**,

number\_of\_players **INTEGER** **NOT** **NULL**,

stadium **CHAR**(50)

);

**CREATE** **TABLE** football\_league(

id\_league **INTEGER** **NOT** **NULL**,

name\_league **CHAR**(50) **NOT** **NULL**

);

**ALTER** **TABLE** footballer **ADD** **CONSTRAINT** pk\_footballer **PRIMARY** **KEY** (id\_footballer);

**ALTER** **TABLE** football\_team **ADD** **CONSTRAINT** pk\_football\_team **PRIMARY** **KEY** (name\_team);

**ALTER** **TABLE** football\_league **ADD** **CONSTRAINT** pk\_football\_league **PRIMARY** **KEY** (id\_league);

**ALTER** **TABLE** footballer **ADD** **CONSTRAINT** fk\_footballer\_team **FOREIGN** **KEY** (name\_team) **REFERENCES** football\_team (name\_team);

**ALTER** **TABLE** football\_team **ADD** **CONSTRAINT** fk\_team\_league **FOREIGN** **KEY** (id\_league) **REFERENCES** football\_league (id\_league);

(Даний код збережений як файл *create.sql)*.

**Завдання 5**

**INSERT** **INTO** football\_league(id\_league, name\_league)

**VALUES**(10001, 'French Ligue 1');

**INSERT** **INTO** football\_league(id\_league, name\_league)

**VALUES**(10002, 'Spain Primera Division');

**INSERT** **INTO** football\_league(id\_league, name\_league)

**VALUES**(10003, 'Italian Serie A');

**INSERT** **INTO** football\_league(id\_league, name\_league)

**VALUES**(10004, 'English League Championship');

**INSERT** **INTO** football\_league(id\_league, name\_league)

**VALUES**(10005, 'Ukrainian Premier League');

**INSERT** **INTO** football\_team(name\_team, id\_league, coach, number\_of\_players, stadium)

**VALUES**('Paris Saint-Germain', 10001, 'Mauricio Roberto Pochettino Trossero', 28, 'Parc des Princes');

**INSERT** **INTO** football\_team(name\_team, id\_league, coach, number\_of\_players, stadium)

**VALUES**('Real Madrid', 10002, 'Carlo Ancelotti', 31, 'Estadio Santiago Bernabéu');

**INSERT** **INTO** football\_team(name\_team, id\_league, coach, number\_of\_players, stadium)

**VALUES**('Juventus', 10003, 'Massimiliano Allegri', 36, 'Allianz Stadium');

**INSERT** **INTO** football\_team(name\_team, id\_league, coach, number\_of\_players, stadium)

**VALUES**('Dynamo Kyiv', 10005, 'Mircea Lucescu', 29, 'NSC «Оlimpiyskiy»');

**INSERT** **INTO** football\_team(name\_team, id\_league, coach, number\_of\_players, stadium)

**VALUES**('Blackpool', 10004, 'Neil Critchley', 30, 'Bloomfield Road');

**INSERT** **INTO** footballer(id\_footballer, name\_team, name\_footballer, age)

**VALUES**(231747, 'Paris Saint-Germain', 'K. Mbappé', 22);

**INSERT** **INTO** footballer(id\_footballer, name\_team, name\_footballer, age)

**VALUES**(158023, 'Paris Saint-Germain', 'L. Messi', 34);

**INSERT** **INTO** footballer(id\_footballer, name\_team, name\_footballer, age)

**VALUES**(20801, 'Juventus', 'Cristiano Ronaldo', 36);

**INSERT** **INTO** footballer(id\_footballer, name\_team, name\_footballer, age)

**VALUES**(190871, 'Paris Saint-Germain', 'Neymar Jr', 29);

**INSERT** **INTO** footballer(id\_footballer, name\_team, name\_footballer, age)

**VALUES**(165153, 'Real Madrid', 'K. Benzema', 33);

**INSERT** **INTO** footballer(id\_footballer, name\_team, name\_footballer, age)

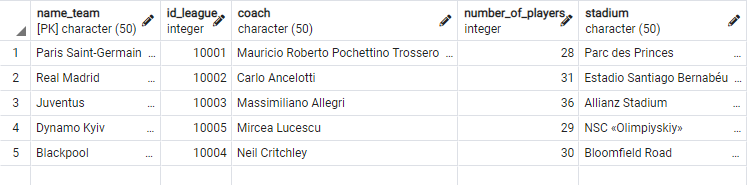
**VALUES**(15983, 'Dynamo Kyiv', 'G. Bushchan', 27);

(Даний код збережений як файл *populate.sql)*.

Таблиця **Football\_League**:



Таблиця **Football\_Teams**:



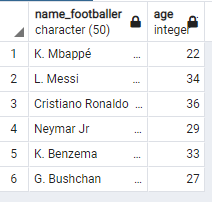
Таблиця **Footballer**:



**Завдання 6**

6a) Запит виводить ім'я гравця та його вік.

**SELECT** name\_footballer, age **FROM** footballer



6b) Запит показує скільки гравців з датасету грають за ту, чи іншу команду.

**SELECT** name\_team, **COUNT**(footballer.name\_team) **AS** pie\_taem **FROM** football\_team **LEFT** **JOIN** footballer **USING**(name\_team) **GROUP** **BY** name\_team



6c) Запит показує, яка кількість гравців перебуває в команді.

**SELECT** number\_of\_players, name\_team **FROM** football\_team



(Даний код збережений як файл *query.sql)*.

**Завдання 7**

Було написано код на Python, що реалізує підключення до PostgreSQL, виконує запити 6a, 6b, 6c та виводить результати їх виконання у консоль

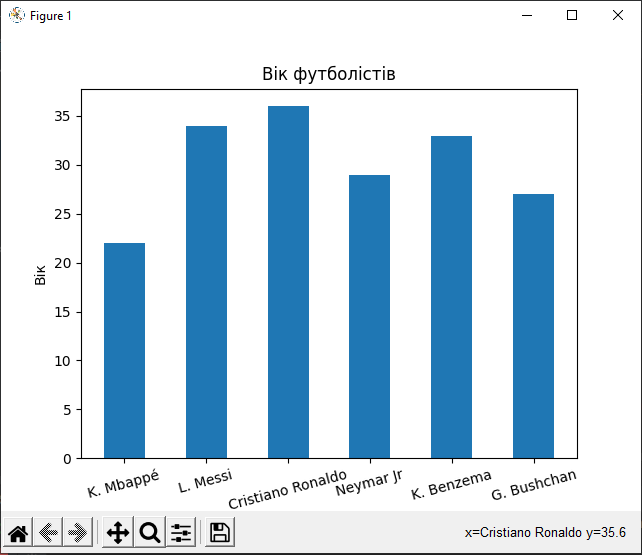
(Даний код збережений як файл *main.py)*.

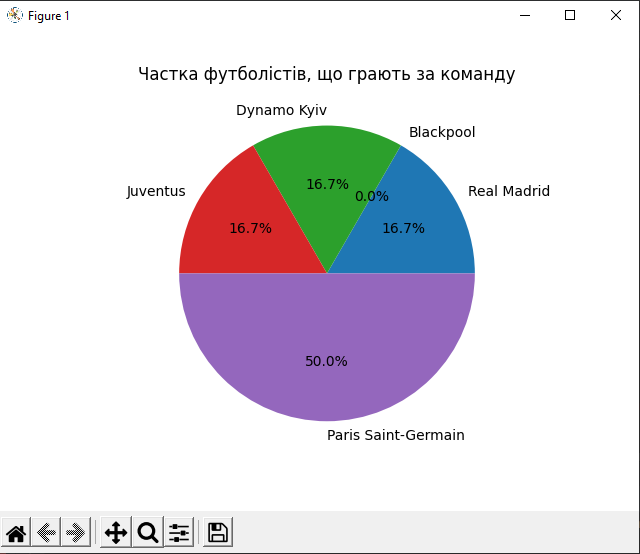


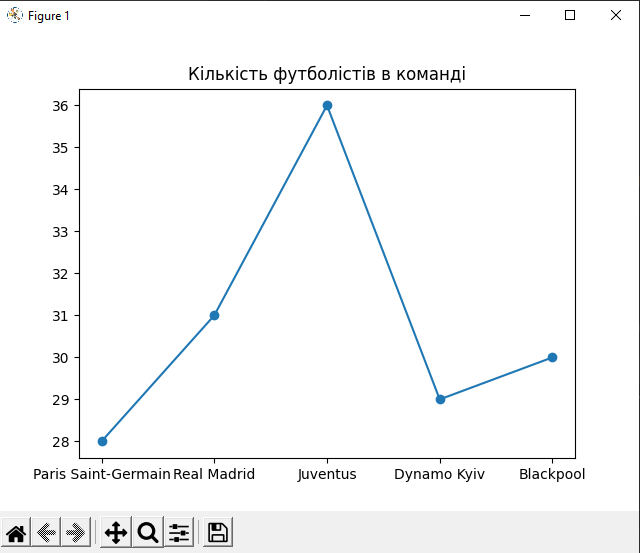
**Завдання 8**

Було реалізовано візуалізацію запитів 6a, 6b, 6c за допомогою mathplotlib на Python. Дана реалізація зображена на наступних рисунках.

(Даний код збережений як файл *visualization.sql)*.







# Список літератури

1. [Форта Б](https://drive.google.com/drive/folders/1QF5rmFwiHXGjSrpwleBnihH93d0VEnKk?usp=sharing). Освой самостоятельно SQL. 3-е изд.: Пер. с.англ. – М.: Изд. дом «Вильямс», 2006. – 100-201с.