

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»

Тема: «Проектування бази даних та ознайомлення з
базовими операціями СУБД PostgreSQL»

Виконав: студент III курсу

ФПМ групи KB-82

Любич І. Д.

Перевірив: Павловський В. І.

Ознайомлення з базовими операціями СУБД PostgreSQL

Метою роботи є здобуття практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

Завдання роботи полягає у наступному:

1. Провести аналіз та опис предметного середовища;
2. Розробити концептуальну модель - модель «сутність-зв'язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі»;
3. Розробити логічну модель (схему даних) БД;
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4;
5. Створити в СУБД PostgreSQL фізичну модель БД, використовуючи конструктори таблиць та стовпчиків;
6. Сформулювати обмеження цілісності, що забезпечують:
 - а. унікальність та обов'язковість вводу первинних ключів для всіх таблиць;
 - б. перевірка на відповідність зовнішніх ключів таблиць;
 - в. обмеження на значення даних відповідних атрибутів і вивід відповідних повідомлень при їх порушенні;
 - г. обов'язковість вводу даних відповідних атрибутів;
 - д. сформулювати маску вводу для відповідних атрибутів;
7. Проаналізувати фізичну модель створеної БД;
8. Заповнити створену БД даними (порядку 5-10 записів в кожній таблиці).
9. Вивести вміст таблиць створеної БД.

Вимоги до ER-моделі

1. Сутності моделі предметної галузі мають містити зв'язки типу 1: N або N: M;
2. Кількість сутностей у моделі – 3-4. Кількість атрибутів у кожній сутності: від двох до п'яти;
3. Сутності мають включати атрибути для коректної реалізації особливостей пошуку, наведених у варіанті;
4. Для побудови ER-діаграм використовувати одну із нотацій: Чена, “Пташиної лапки (Crow's foot)”, UML.

Зміст звіту

1. Опис предметної області;
2. Модель сутність-зв'язок;
3. Схема перетвореної моделі в схеми БД;
4. Опис структури БД;
5. Нормалізація БД;
6. Логічна модель (SqlDMB);
7. Текст програми (PostgreSql);

Опис предметної області БД «Командні спортивні змагання»

Сутності:

1. Команда – група учасників змагань. Має такі атрибути: назва команди, спонсор.
2. Змагання – змагання, яке проводиться на стадіоні. Має такі атрибути: дата та час початку, тривалість, арбітр.
3. Результати вправ – інформація про кількість очок команд з вправ. Має такі атрибути: назва вправи, кількість очок.
4. Стадіон – місце, де відбуваються змагання. Має такі атрибути: місцезнаходження, кількість місць.
5. Тренер – той, хто тренує команду. Має такі атрибути: дата народження, прізвище та ім'я.

Зв'язки:

1. Декілька команд може записатись на одне змагання, у той же час одна команда має право записатись на декілька змагань — M:N. Кожна команда, яка бере участь у змаганні має зайняте місце та суму набраних очок.
2. Одне змагання може мати декілька результатів вправ — 1:N.
3. Одна команда може мати декілька результатів вправ — 1:N.
4. Кожна команда має одного тренера, але не кожен тренер має команду — 1: 0..1.
5. На одному стадіоні може проходити декілька ігор — 1: 0..N.

Приклад:

Змагання з силового триборства на стадіоні S. Беруть участь три команди: А, Б, В, які мають тренерів _А, _Б, _В. Команди виконують вправи: жим лежачи, присід та тягу. 9 результатів вправ записано до таблиці «Результати вправ». За даними цієї таблиці (серед яких id команд та змагань) розраховуються атрибути зв'язку Команда_Змагання.

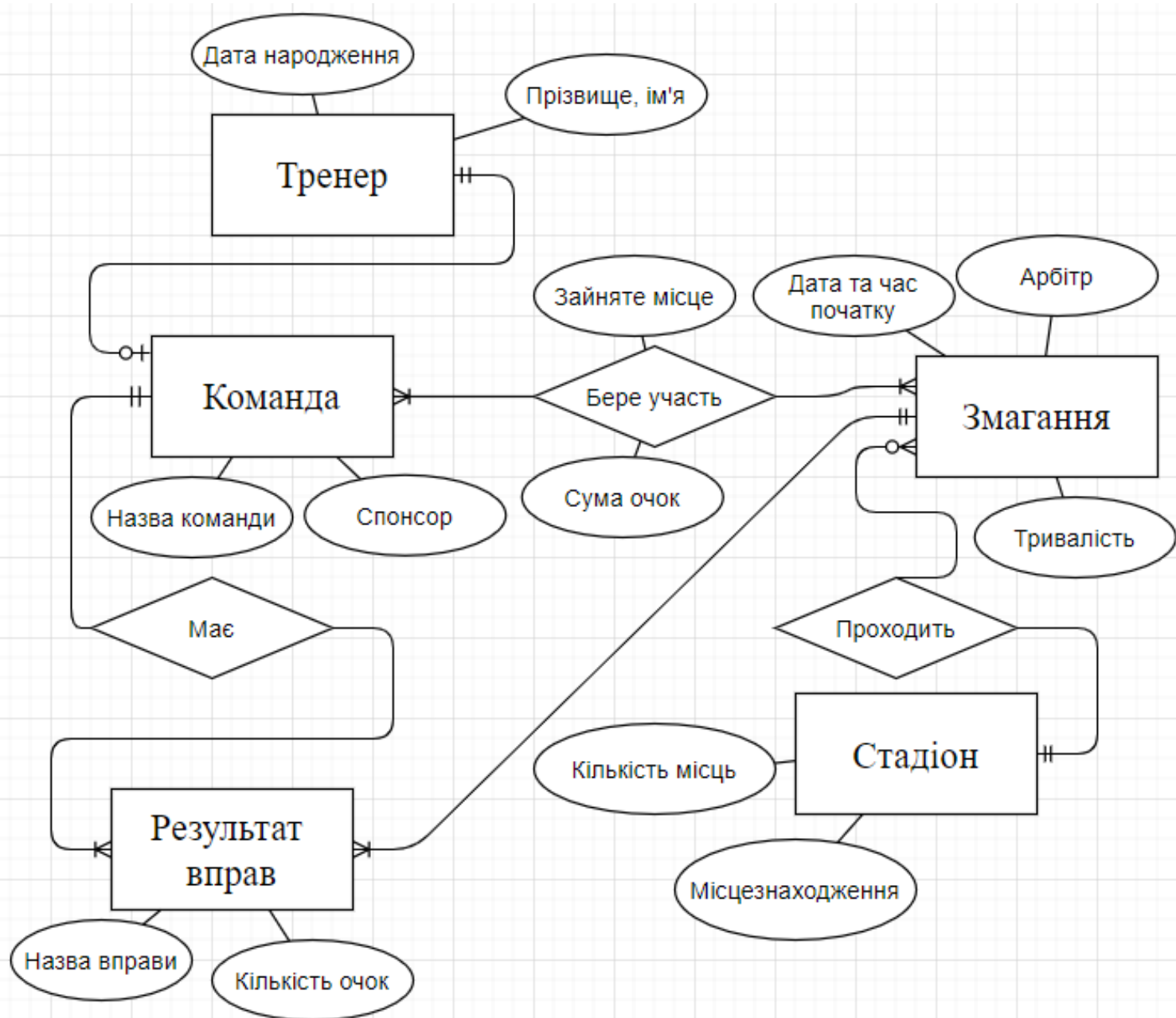


Рисунок 1 - Графічний файл розробленої моделі «сутність-зв'язок»

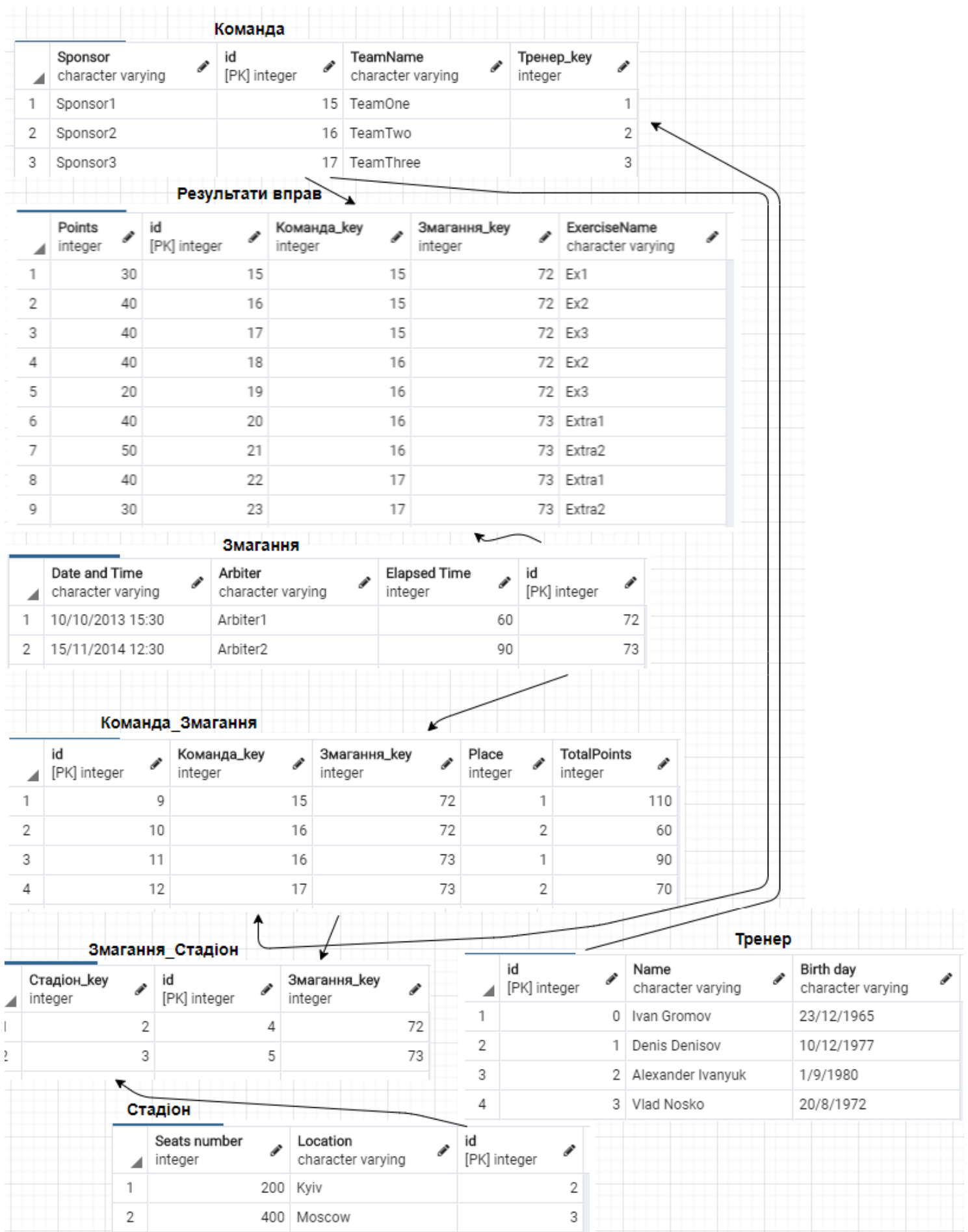


Рисунок 2 – Схема БД PostgreSQL

Сутності було перетворено у відповідні таблиці БД.

Оскільки зв'язок між командою і результатами вправ 1 : N, у таблиці «Результати вправ» було створено атрибут з первинним ключем команди. За таким же принципом до результатів вправ додано атрибут з первинним ключем змагання.

Зв'язок між сутностями «Команда» і «Змагання» та «Стадіон» і «Змагання» зумовили появу двох таблиць, які зберігають первинні ключі відповідних сутностей, а таблиця з ключами команд і змагань також зберігає інформацію про результати команд у змаганнях.

Зв'язок між тренером і командою 1:0..1 зумовив додавання атрибута (Unique) з первинним ключем тренера до таблиці «Команда».

Опис структури БД

Таблиця 1 – Опис таблиць БД

Назва таблиці, опис	Атрибути, опис	Тип атрибута
Змагання – інформація про змагання	(PK) id – унікальний ідентифікатор	integer
	Date and Time – дата та час змагання	character varying
	Arbiter – арбітр змагання	character varying
	Elapsed Time – час, який займає змагання	integer
Змагання_Стадіон – зв'язок між «Змагання» та «Стадіон»	(PK) id – унікальний ідентифікатор зв'язку	integer
	(FK) Стадіон_key – зовнішній ключ для доступу до стадіону	integer
	(FK) Змагання_key - зовнішній ключ для доступу до змагання	integer
Команда_Змагання – зв'язок між «Команда» та «Змагання». Зберігає інформацію про результати змагань команд	(PK) id – унікальний ідентифікатор зв'язку	integer
	(FK) Команда_key - зовнішній ключ для доступу до команди	integer
	(FK) Змагання_key - зовнішній ключ для доступу до змагання	integer

	Place - рейтингове місце	integer
	TotalPoints – загальна кількість очок	integer
Стадіон – місце проведення змагань	(PK) id – унікальний ідентифікатор	integer
	Seats number – кількість місць на стадіоні	integer
	Location – місце-знаходження ст-ну	character varying
Команда – учасник змагань	(PK) id – унікальний ідентифікатор	integer
	Sponsor – спонсор команди	character varying
	TeamName – назва команди	character varying
	(FK) Тренер_key – зовнішній ключ для доступу до тренера	integer
Результат вправ – інформація результати виконаних командою вправ	(PK) id – унікальний ідентифікатор	integer
	Points – кількість очок	integer
	(FK) Команда_key - зовнішній ключ для доступу до команди	integer
	(FK) Змагання_key – зовнішній ключ для доступу до змагання	integer
	ExerciseName	character varying
Тренер – той, хто тренує команду	(PK) id – унікальний ідентифікатор	integer
	Name – ім'я та прізвище	character varying
	Birth day – дата народження	character varying

Нормалізація БД

Перша нормальна форма. Таблиці відповідають 1НФ, оскільки кожній комірці таблиці відповідає одне значення і записи у таблицях не повторюються.

Друга нормальна форма. Таблиці відповідають 2НФ, оскільки п.1 і кожен атрибут в таблицях залежать від цілого ключа, а не від його частини.

Третя нормальна форма. Таблиці відповідають 3НФ, оскільки п.2 і кожен непервинний атрибут таблиць є транзитивно незалежним від первинних атрибутів.

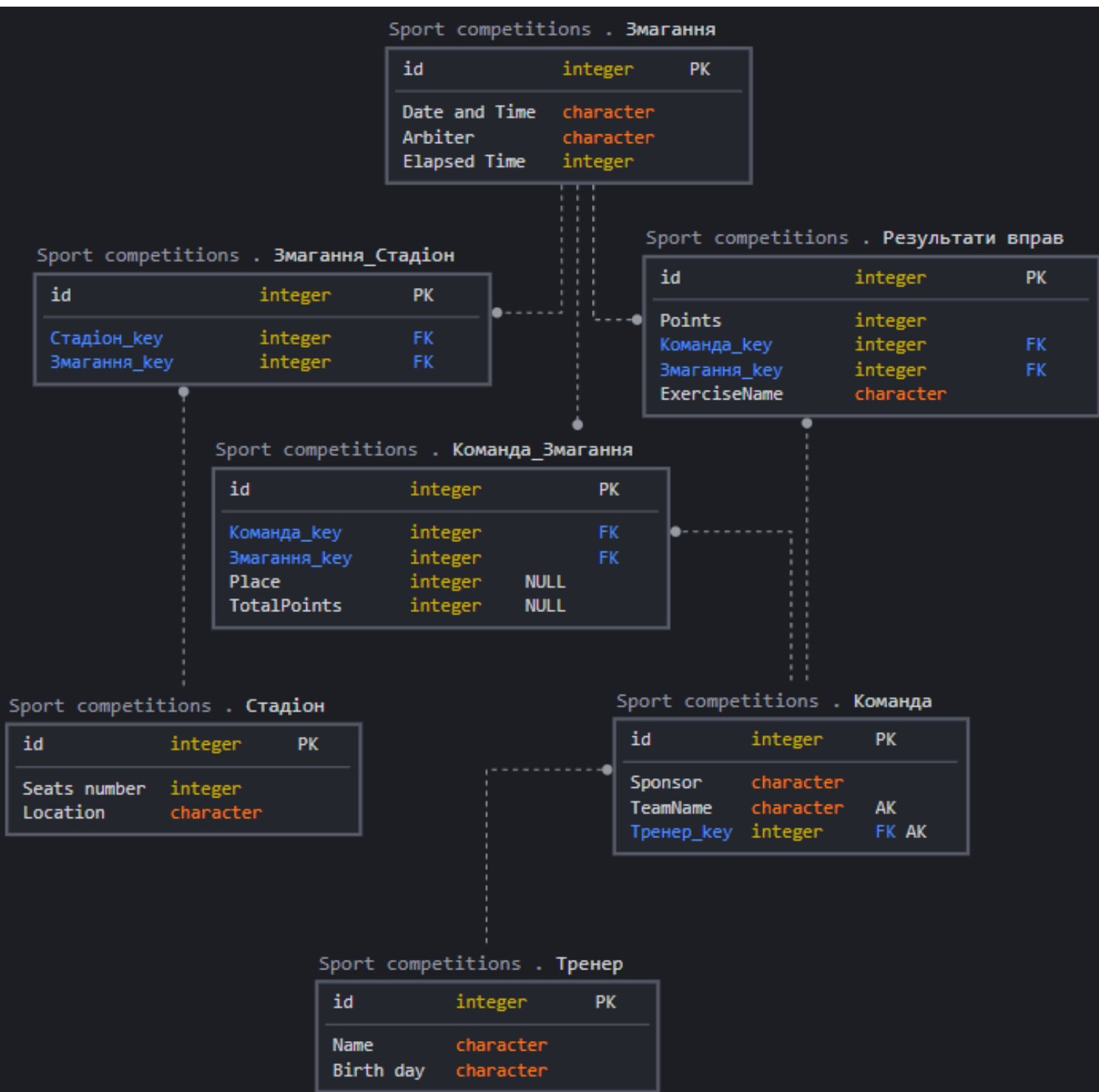
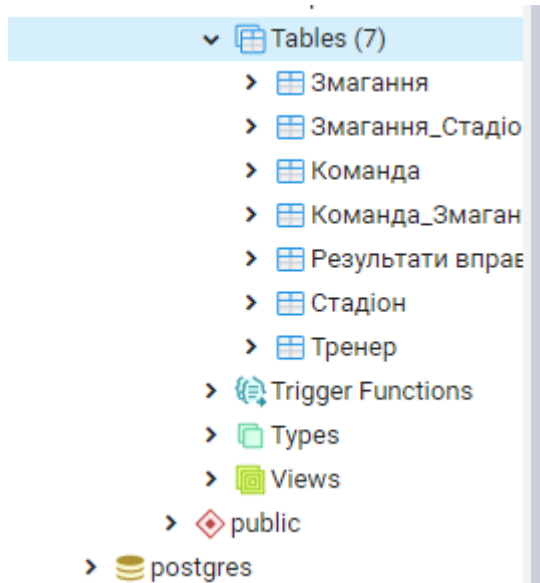


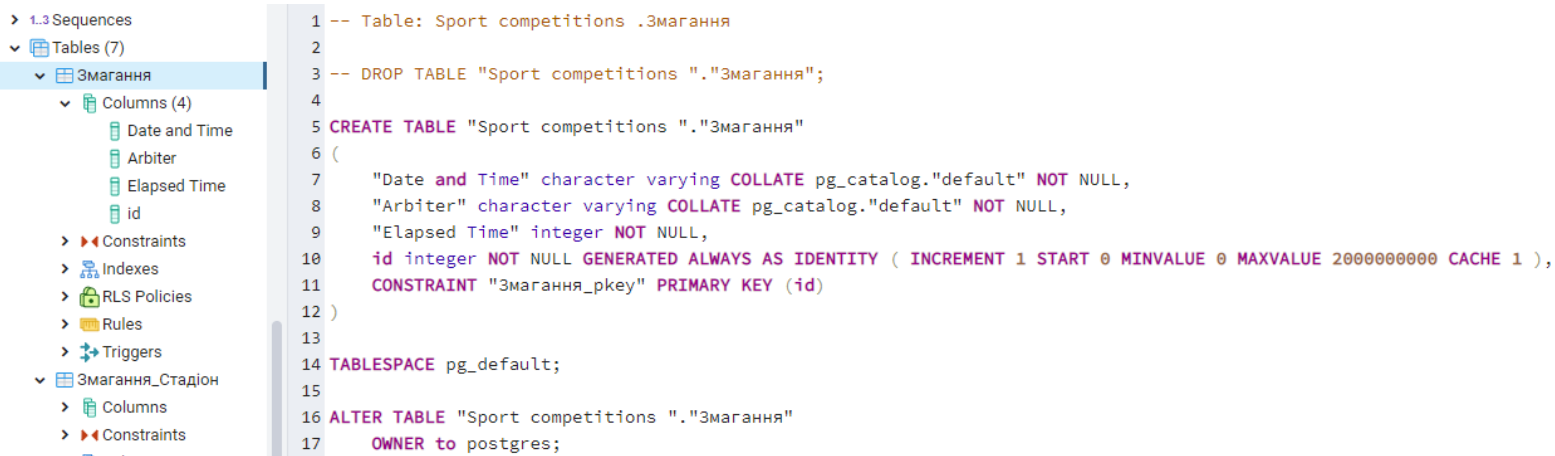
Рисунок 3 – Логічна модель БД PostgreSQL (використовуючи SqlDBM)

Структура програми БД (pgAdmin 4)

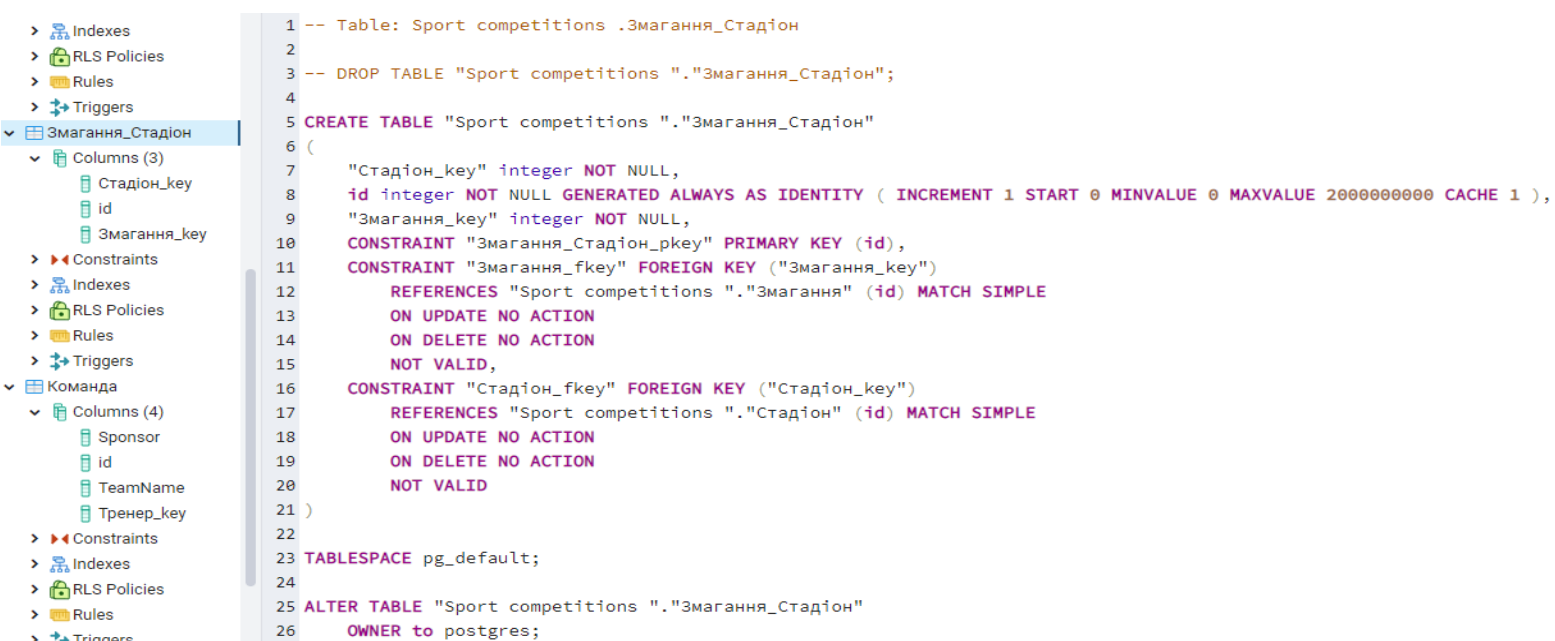


Текст програми БД (pgAdmin 4)

Змагання



Змагання_Стадіон



Команда

> Indexes	1 -- Table: Sport competitions .Команда
> RLS Policies	2
> Rules	3 -- DROP TABLE "Sport competitions"."Команда";
> Triggers	4
▼ Команда	5 CREATE TABLE "Sport competitions"."Команда"
▼ Columns (4)	6 (
Sponsor	7 "Sponsor" character varying COLLATE pg_catalog."default" NOT NULL,
id	8 id integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1),
TeamName	9 "TeamName" character varying COLLATE pg_catalog."default" NOT NULL,
Тренер_key	10 "Тренер_key" integer NOT NULL,
> Constraints	11 CONSTRAINT "Команда_pkey" PRIMARY KEY (id),
> Indexes	12 CONSTRAINT "CoachUnique" UNIQUE ("Тренер_key"),
> RLS Policies	13 CONSTRAINT "NameUnique" UNIQUE ("TeamName"),
> Rules	14 CONSTRAINT "Тренер_fkey" FOREIGN KEY ("Тренер_key")
> Triggers	15 REFERENCES "Sport competitions"."Тренер" (id) MATCH SIMPLE
▼ Команда_Змагання	16 ON UPDATE NO ACTION
▼ Columns (5)	17 ON DELETE NO ACTION
id	18 NOT VALID
Команда_key	19)
Змагання_key	20
Place	21 TABLESPACE pg_default;
TotalPoints	22
> Constraints	23 ALTER TABLE "Sport competitions"."Команда"
	24 OWNER to postgres;

Команда_Змагання

> Constraints	1 -- Table: Sport competitions .Команда_Змагання
> Indexes	2
> RLS Policies	3 -- DROP TABLE "Sport competitions"."Команда_Змагання";
> Rules	4
> Triggers	5 CREATE TABLE "Sport competitions"."Команда_Змагання"
Команда_Змагання	6 (
▼ Columns (5)	7 id integer NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT 1 START 0 MINVALUE 0 MAXVALUE 2000000000 CACHE 1),
id	8 "Команда_key" integer NOT NULL,
Команда_key	9 "Змагання_key" integer NOT NULL,
Змагання_key	10 "Place" integer DEFAULT 0,
Place	11 "TotalPoints" integer DEFAULT 0,
TotalPoints	12 CONSTRAINT "Команда_Змагання_pkey" PRIMARY KEY (id),
> Constraints	13 CONSTRAINT "Змагання_fkey" FOREIGN KEY ("Змагання_key")
> Indexes	14 REFERENCES "Sport competitions"."Змагання" (id) MATCH SIMPLE
> RLS Policies	15 ON UPDATE NO ACTION
> Rules	16 ON DELETE NO ACTION
> Triggers	17 NOT VALID,
▼ Результати вправ	18 CONSTRAINT "Команда_fkey" FOREIGN KEY ("Команда_key")
▼ Columns	19 REFERENCES "Sport competitions"."Команда" (id) MATCH SIMPLE
> Constraints	20 ON UPDATE NO ACTION
> Indexes	21 ON DELETE NO ACTION
> RLS Policies	22 NOT VALID
> Rules	23)
> Triggers	24
▼ Стадіон	25 TABLESPACE pg_default;
▼ Тренер	26
Trigger Functions	27 ALTER TABLE "Sport competitions"."Команда_Змагання"
	28 OWNER to postgres;

Результати вправ

- Place
- TotalPoints
- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Результати вправ
 - Columns (5)
 - Points
 - id
 - Команда_key
 - Змагання_key
 - ExerciseName
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- Стадіон
- Тренер
- Trigger Functions
- Types
- Views
- public
- es
- up Roles (9)

```
1 -- Table: Sport competitions .Результати вправ
2
3 -- DROP TABLE "Sport competitions"."Результати вправ";
4
5 CREATE TABLE "Sport competitions"."Результати вправ"
6 (
7     "Points" integer NOT NULL DEFAULT 0,
8     id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 0 MINVALUE 0 MAXVALUE 2000000000 CACHE 1 ),
9     "Команда_key" integer NOT NULL,
10    "Змагання_key" integer NOT NULL,
11    "ExerciseName" character varying COLLATE pg_catalog."default" NOT NULL,
12    CONSTRAINT "РезультатиВправ_pkey" PRIMARY KEY (id),
13    CONSTRAINT "Змагання_fkey" FOREIGN KEY ("Змагання_key")
14        REFERENCES "Sport competitions"."Змагання" (id) MATCH SIMPLE
15        ON UPDATE NO ACTION
16        ON DELETE NO ACTION
17        NOT VALID,
18    CONSTRAINT "Команда_fkey" FOREIGN KEY ("Команда_key")
19        REFERENCES "Sport competitions"."Команда" (id) MATCH SIMPLE
20        ON UPDATE NO ACTION
21        ON DELETE NO ACTION
22        NOT VALID
23 )
24
25 TABLESPACE pg_default;
26
27 ALTER TABLE "Sport competitions"."Результати вправ"
28     OWNER to postgres;
```

Стадіон

- Constraints
- Indexes
- RLS Policies
- Rules
- Triggers
- Стадіон
 - Columns (3)
 - Seats number
 - Location
 - id
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers

```
1 -- Table: Sport competitions .Стадіон
2
3 -- DROP TABLE "Sport competitions"."Стадіон";
4
5 CREATE TABLE "Sport competitions"."Стадіон"
6 (
7     "Seats number" integer NOT NULL,
8     "Location" character varying COLLATE pg_catalog."default" NOT NULL,
9     id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 0 MINVALUE 0 MAXVALUE 2000000000 CACHE 1 ),
10    CONSTRAINT "Стадіон_pkey" PRIMARY KEY (id)
11 )
12
13 TABLESPACE pg_default;
14
15 ALTER TABLE "Sport competitions"."Стадіон"
16     OWNER to postgres;
```

Тренер

- Triggers
- Стадіон
- Тренер
 - Columns (3)
 - id
 - Name
 - Birth day
 - Constraints

```
1 -- Table: Sport competitions .Тренер
2
3 -- DROP TABLE "Sport competitions"."Тренер";
4
5 CREATE TABLE "Sport competitions"."Тренер"
6 (
7     id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 0 MINVALUE 0 MAXVALUE 2000000000 CACHE 1 ),
8     "Name" character varying COLLATE pg_catalog."default" NOT NULL,
9     "Birth day" character varying COLLATE pg_catalog."default" NOT NULL,
10    CONSTRAINT "Тренер_pkey" PRIMARY KEY (id)
11 )
12
13 TABLESPACE pg_default;
14
15 ALTER TABLE "Sport competitions"."Тренер"
16     OWNER to postgres;
```