

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

**Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем**

КУРСОВА РОБОТА

з дисципліни “Бази даних”

спеціальність 121 – Програмна інженерія

на тему: База даних організації спортивних змагань

**Студентка
групи КП-02**

Красношапка А.А.

(підпис)

**Викладач
к.т.н, доцент кафедри СПіСКС**

Радченко К.О.

(підпис)

Захищено з оцінкою _____

Київ – 2021

Анотація

Під час виконання курсової роботи був створений додаток для аналізу роботи університету та успішності студентів. Для зручного зберігання великої кількості інформації про університет була створена база даних. Також в курсовій роботі було використано декілька шляхів генерування даних, щоб продемонструвати роботу додатка.

В анотації у реферативному стилі наводиться інформація про зміст та результати, що отримані в курсовій роботі. В анотації у реферативному стилі наводиться інформація про зміст та результати, що отримані в курсовій роботі.

Зміст

Вступ.

1. Аналіз інструментарію для виконання курсової роботи.	1
2. Структура бази даних.	2
3. Опис програмного забезпечення.	3
3.1. Загальна структура програмного забезпечення.	3
3.2. Опис модулів програмного забезпечення.	3
3.3. Опис основних алгоритмів роботи.	4
4. Аналіз функціонування засобів реплікації.	5
5. Аналіз функціонування засобів резервування/відновлення.	6
6. Аналіз результатів підвищення швидкодії запитів.	7
7. Опис результатів аналізу предметної галузі.	8

Висновки.

Література.

Додатки.

Вступ

Призначенням роботи є набуття практичних навичок розробки сучасного програмного забезпечення, що взаємодіє з реляційними базами даних, а також здобуття навичок оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації.

Об'єктом розробки у курсовій роботі є інформаційно-аналітична система, яка дозволяє отримувати інформацію з організації спортивних загань або генерувати та здійснювати її обробку та аналіз.

Система складається з наступних компонентів: засобів генерації даних, засобів фільтрації та валідації даних, бази даних системи, засобів реплікації, засобів аналізу даних та засобів резервування та відновлення даних.

Аналіз інструментарію для виконання курсової роботи

Для роботи була обрана СУБД PostgreSQL. Вона має значні переваги над іншими SQL базами даних з відкритим вихідним кодом, тому що PostgreSQL не просто реляційна, а об'єктно-реляційна СУБД. Фундаментальна характеристика об'єктно-реляційної бази даних – це підтримка об'єктів і їх поведінки, включаючи типи даних, функції, операції, домени і індекси. Це робить PostgreSQL неймовірно гнучким і надійним. Серед іншого, він вміє створювати, зберігати та видавати складні структури даних. Існує великий список типів даних, які підтримує PostgreSQL.

Для розробки програмного додатку була обрана мова Python, оскільки це мова програмування загального призначення, націлена в першу чергу на підвищення швидкості написання коду. Python багато в чому лаконічний і зрозумілий. За рахунок простоти коду, подальший супровід програм, написаних на Python, стає легше і приємніше в порівнянні з іншими мовами програмування. Також, оскільки нами була обрана СУБД PostgreSQL, нам необхідна була мова, яка б забезпечувала зручну роботу з нею. Для роботи з PostgreSQL Python пропонує зручні бібліотеки такі як psycopg2 та SQLAlchemy, що значно пришвидшує розробку. Для написання програмного додатку мовою Python було обрано середовище розробки Pycharm, оскільки воно забезпечує зручну організацію проектів, рефакторинг та автокомпліт. Він слідкує за дотриманням всіх правил написання коду мовою програмування Python. А також автоматично налаштовує віртуальне середовище та забезпечує можливість зручного встановлення пакетів в проект.

Структура бази даних

Схема бази даних - Див. додаток 1.

Перелік сутностей:

1. Сутність “participants”, яка містить інформацію про ім’я, дату народження, стать, зріст, вагу та вчителя учасника спортивного змагання.
2. Сутність “teachers”, яка містить інформацію про ім’я та дату народження вчителя, а також дату початку його занять з дисципліни.
2. Сутність “competitions”, яка містить інформацію про назву змагання, вид спорту, його організатора, бюджет, країну та рік його організації.

Опис програмного забезпечення

1. Загальна структура програмного забезпечення

Програмне забезпечення складається з двох частин:

- 1) Project – програма, що дозволяє користувачу взаємодіяти з базою даних за допомогою консольного інтерфейсу. Розроблена за допомогою шаблону MVC.
- 2) Index – утиліта, яка створює індекс та аналізує збільшення швидкодії оброблення запиту за його наявності.

2. Опис модулів програмного забезпечення

Project складається з наступних модулів:

- 1) main.py – передає отримані запити від view.py до database.py, отримує відповідь, яку обробляє та надсилає результат до view.py.
- 2) database.py – реалізує взаємодію з базою даних та передає отримані результати в main.py.
- 3) view.py – реалізує взаємодію з користувачем, тобто отримує дані, передає дані до main.py та виводить дані з нього.
- 4) models.py – зберігає інформацію про ORM моделі бази даних.
- 5) text_reading.py – реалізує взаємодію з файлом для генерації псевдовипадкових даних.

Index складається з єдиного модуля index_b_tree, який містить в собі одразу всі засоби для створення індексу та аналізу збільшення швидкодії.

3. Опис основних алгоритмів роботи

Програмний додаток реалізує такі основні алгоритми роботи:

- 1) Операції CRUD над усіма таблицями бази даних. (Див. додаток 2)
- 2) Фільтрація запитів за визначеними користувачем критеріями. (Див. Додаток 3)
- 3) Генерація всіх таблиць в базі даних.(Див. додаток 4)
- 4) Статистичний аналіз даних за допомогою pandas з наступним виведенням його у вигляді графіків за допомогою matplotlib. (Див. Додаток 5)

Аналіз функціонування засобів реплікації

В якості методу реплікації бази даних була вибрана логічна реплікація. Її було реалізовано на моделі публікації/підписки, де головна база даних створює публікацію при зміні даних, а база даних-підписник отримує інформацію о змінах через підписку та повторює зміни для даних, що в ній зберігаються. Такий метод реплікації дозволяє у випадку неможливості підключення до головної бази даних зчитувати інформацію з бази даних-підписника, але не дозволяє редагувати дані, оскільки зв'язок між базами даних односторонній.

Тестування реплікації проводилось через унеможливлення під'єднання до головної бази даних. У цьому випадку задачу пошуку даних на себе взяла база даних-підписник. (Див. Додаток 6)

Аналіз функціонування засобів резервування/відновлення

Резервування було реалізоване за допомогою додатка Effector Saver. Він створює копію бази даних у файлі з розширенням .backup, архівує його для зменшення використаної пам'яті та зберігає створений архів на Google Drive (Див. Додаток 7). Резервування відбувається кожного дня у 1:00 годину ночі.

За допомогою цього ж додатка Effector Saver було проведення відновлення даних у спеціально створену для цього іншу базу даних. Після відновлення у пустій базі даних всі таблиці та дані з початкової були відновлені без жодних порушень (Див. Додаток 8).

Аналіз результатів підвищення швидкодії запитів

Для підвищення швидкодії був створений індекс Btree в таблиці participants для поля height. Індекс btree, він же В-дерево, придатний для даних, які можна відсортувати. Іншими словами, для типу даних повинні бути визначені оператори «більше», «більше або одно».

Для перевірки збільшення швидкодії був виконан 100 разів запит “explain analyze select height from participants where height>80” з індексом та без індекса. На графіку видно, що індекси допомагають значно пришвидшити роботу запитів. (Див. додаток 9)

Опис результатів аналізу предметної галузі

Приклади виконання операцій CRUD над базою даних (Див. Додаток 10):

Приклад фільтрації даних з веденням значень фільтрації користувачем (Див. додаток 11)

Приклад генерації (Див. додаток 12). Генерування було здійснене за допомогою зчитування даних з .csv файлу, завантаженого з сайту [kaggle.com](https://www.kaggle.com) з датасетами. Також для генерації імен був використаний пакет `Faker`.

Приклад статичного аналізу даних з наступним виведенням результату у вигляді графіку (Див. додаток 13). Аналіз даних був зроблений за допомогою `pandas` з наступним виведенням його у вигляді графіків за допомогою `matplotlib`.

Висновки

Під час виконання курсової роботи був розроблений програмний додаток, який забезпечує збір, аналіз, маніпуляцію, валідацію та фільтрацію даних, необхідних для організації спортивних змагань. В програмному додатку наявні всі можливі операції CRUD над усіма таблицями. Крім того наявна можливість фільтрації даних за допомогою попередньо введених користувачем значень фільтрації.

Також в програмі було реалізована генерація даних для всіх таблиць. Генерація відбувається за допомогою пакету Faker та датасетів з kaggle.com.

Для підтримки безвідмовної роботи програми та розподілення навантаження була створена реплікація, яка розміщена на цьому ж пристрої. При відмові головної бази даних, підписана на неї база даних бере виконання задач пошуку на себе.

Крім цього додатково регулярно виконується бекап нашої бази даних додатком Effector Saver, щоб унеможливити втрату даних.

Статистичний аналіз різних даних був проведений за допомогою бібліотеки pandas та matplotlib.

Також було створено декілька індексів та було розглянуто їх швидкодія на 100 запитах мовою SQL, щоб переконатись, що створені індекси і справді пришвидшують роботу програми.

Мета курсової роботи була досягнута та необхідні навички були набуті.

Література

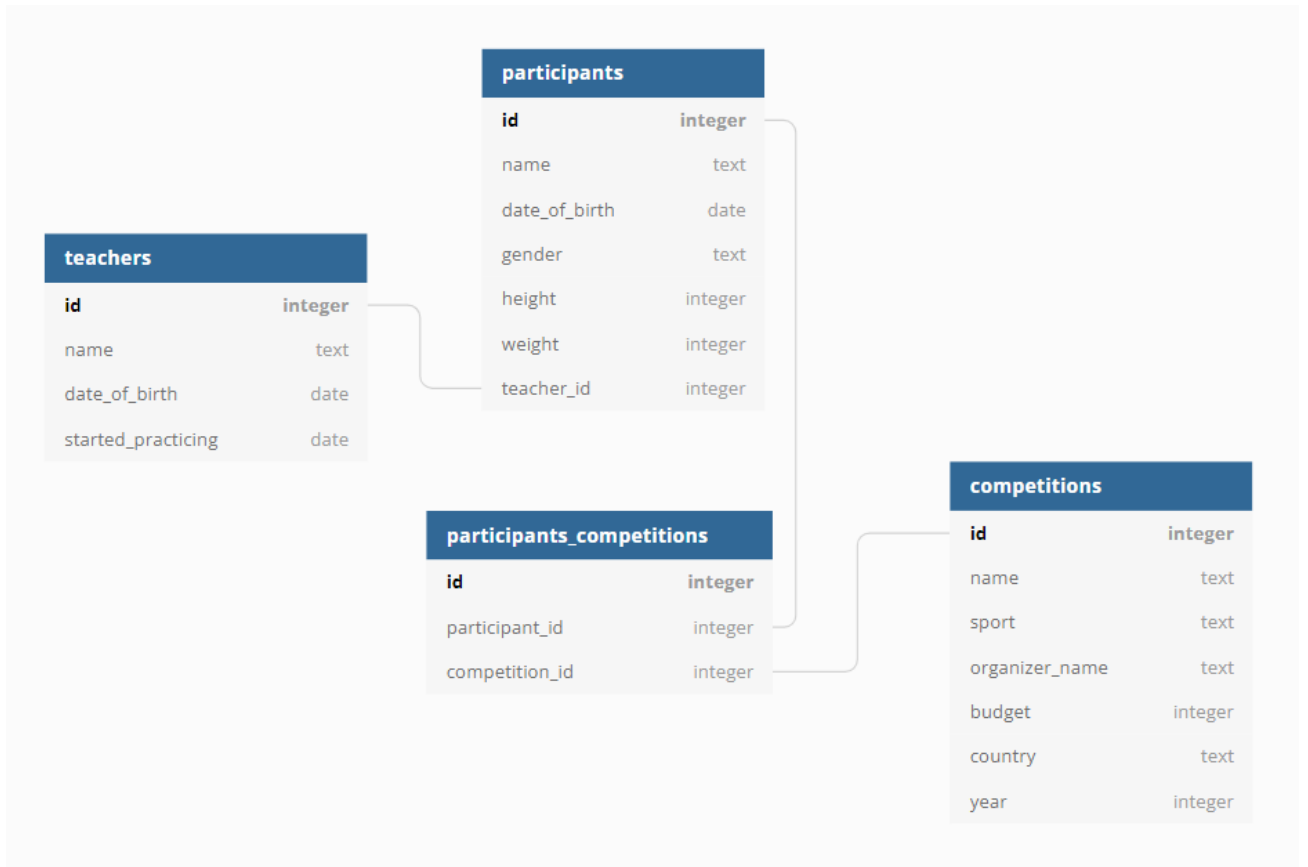
1. Глава 31. Логическая репликация. URL:
<https://postgrespro.ru/docs/postgresql/10/logical-replication> (дата звернення 20.12.2021)
2. Репликация в PostgreSQL без проблем. URL:
<https://habr.com/ru/sandbox/150526/> (дата звернення 19.12.2021)
3. Publication. URL:
<https://www.postgresql.org/docs/current/logical-replication-publication.html> (дата звернення 20.12.2021)
4. Subscription. URL:
<https://www.postgresql.org/docs/current/logical-replication-subscription.html> (дата звернення 20.12.2021)
5. Резервне копіювання бази даних PostgreSQL. URL:
<https://efsaver.com.ua/docs/rezervne-kopiyuvannya-bazi-danix-postgresql.html> (дата звернення 20.12.2021)
6. Як відновити дані PostgreSQL з резервної копії. URL:
<https://efsaver.com.ua/docs/restore-postgresql-data-from-backup.html> (дата звернення 20.12.2021)
7. Индексы в PostgreSQL — 4. URL:
<https://habr.com/ru/company/postgrespro/blog/330544/> (дата звернення 18.12.2021)
8. Matplotlib. Tutorials. URL: <https://matplotlib.org/stable/tutorials/index.html> (дата звернення 09.12.2021)

9. Pandas documentation. Pandas.DataFrame. URL:
<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html> (дата звернення 09.12.2021)
10. SQLAlchemy ORM Tutorial for Python Developers. URL:
<https://auth0.com/blog/sqlalchemy-orm-tutorial-for-python-developers/> (дата звернення 29.11.2021)
11. Welcome to Faker's documentation! URL: <https://faker.readthedocs.io/en/master/>
(дата звернення 20.12.2021)
12. Olympic Sports and Medals, 1896-2014. URL:
<https://www.kaggle.com/the-guardian/olympic-games?select=summer.csv> (дата звернення 21.12.2021)

Додатки

Додаток 1

Схема бази даних:



Додаток 2:

Операції CRUD над усіма таблицями бази даних:

1. Find a participant by id
2. Find a teacher by id
3. Find a competition by id


```
7. Find all participants
8. Find all teachers
9. Find all competitions
```

```
10. Add a participant
11. Add a teacher
12. Add a competition
13. Update a participant
14. Update a teacher
15. Update a competition
16. Delete a participant
17. Delete a teacher
18. Delete a competition
```

Додаток 3:

Фільтрація запитів за визначеними користувачем критеріями:

```
19. Find participant of a particular gender and where name like some text and teacher's name is like some text.
20. Find competition on a particular sport with participants whose height is between some range.
```

Додаток 4:

Генерація всіх таблиць в базі даних:

```
21. Generate participants
22. Generate teachers
23. Generate competitions
24. Generate participants and competitions connections
```

Додаток 5:

Статистичний аналіз даних:

```
25. Count competitions of each year for a particular participant
26. Count participants of each gender for a particular competition
```

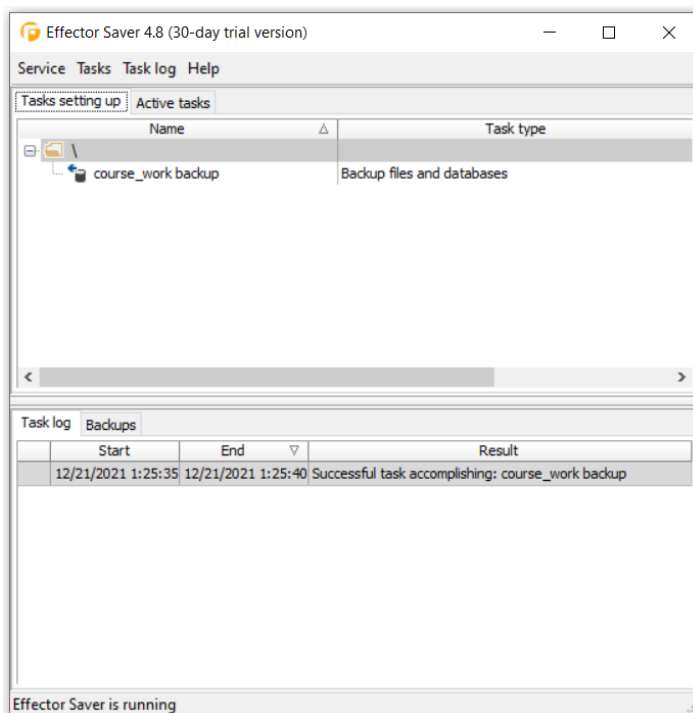
Додаток 6:

Пошук здійснений на базі даних-підписнику:

```
Enter your choice:2
Enter id:60
Go to slave server
<Teacher(id=60, name='Lisa Francis', date_of_birth='1982-10-30', started_practicing='1993-11-03')>
```


Додаток 7:

Резервування за допомогою Effector Saver:




Збережений архів на Google Drive:

My Drive > db_course_work ▾


Name ↑	Owner	Last modified	File size
 cw_backup_2021-12-21.zip	me	1:25 AM	3 KB


Додаток 8:


Відновлена база даних:


>  Procedures


> 1..3 Sequences


>  Tables (4)


>  competitions


>  participants

>  participants_competitions

>  teachers

>  Trigger Functions

>  Types

>  Views

Subscriptions

Data Output

Explain

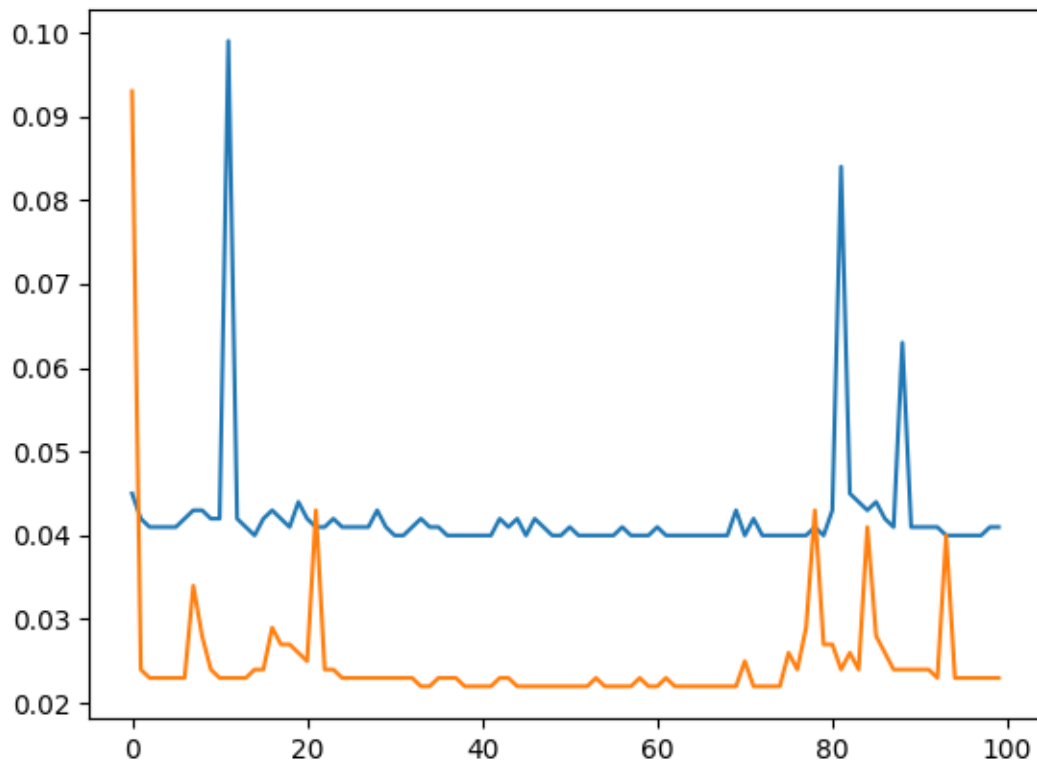
Messages

Notifications

	<div>id</div> <div>[PK] integer</div>	<div>participant_id</div> <div>integer</div>	<div>competition_id</div> <div>integer</div>
1	1	301	44
2	2	89	149
3	3	308	134
4	4	262	295
5	5	145	131
6	6	385	287
7	7	145	110

Додаток 9:

Підвищення швидкодії запитів за допомогою індексування:



Додаток 10:

Приклади виконання операцій CRUD:

Add participant:

```
Enter your choice: 10
Enter full name:Alice Watergate
Date of birth:
Enter year:1990
Enter month:6
Enter day: 21
Enter gender M or F:F
Enter height in inches:60
Enter weight in kg:55
Enter teacher_id:10
Add is successful
```

Find teacher:

```
Enter your choice:2
Enter id:10
<Teacher(id=10, name='Craig Bates', date_of_birth='1982-08-25', started_practicing='2000-07-01')>
```

Find all participants:

```
Enter your choice:1
<Participant(id=17, name='John Simpson', date_of_birth='1987-05-07', gender='M', height='63', weight='81', teacher_id='90')>
<Participant(id=18, name='Linda Rivas', date_of_birth='1996-11-07', gender='M', height='65', weight='140', teacher_id='43')>
<Participant(id=19, name='David Ford', date_of_birth='1981-08-09', gender='M', height='82', weight='66', teacher_id='75')>
<Participant(id=20, name='Marc Gill', date_of_birth='1966-08-03', gender='F', height='45', weight='81', teacher_id='38')>
<Participant(id=21, name='Anthony Salazar', date_of_birth='1997-05-08', gender='M', height='80', weight='59', teacher_id='78')>
<Participant(id=22, name='Susan Hernandez', date_of_birth='1979-12-27', gender='M', height='59', weight='102', teacher_id='294')>
<Participant(id=23, name='Dawn Valentine', date_of_birth='1986-07-17', gender='F', height='65', weight='124', teacher_id='51')>
<Participant(id=24, name='Jeremy Woods', date_of_birth='1987-02-12', gender='F', height='80', weight='72', teacher_id='199')>
<Participant(id=25, name='Jeffery Keith', date_of_birth='1967-02-11', gender='F', height='50', weight='48', teacher_id='250')>
<Participant(id=26, name='Stanley Tapia', date_of_birth='1966-12-18', gender='F', height='73', weight='146', teacher_id='287')>
<Participant(id=27, name='Mrs. Robin Martin', date_of_birth='1975-05-26', gender='F', height='70', weight='69', teacher_id='104')>
<Participant(id=28, name='Andrew Edwards', date_of_birth='1985-03-11', gender='M', height='46', weight='73', teacher_id='219')>
<Participant(id=29, name='Gregory Vargas', date_of_birth='1999-07-28', gender='F', height='61', weight='118', teacher_id='57')>
<Participant(id=30, name='Katherine Hensley', date_of_birth='1975-09-23', gender='F', height='78', weight='131', teacher_id='211')>
<Participant(id=31, name='James Perez', date_of_birth='1986-12-18', gender='F', height='75', weight='105', teacher_id='167')>
<Participant(id=32, name='Brianna Armstrong', date_of_birth='1978-10-16', gender='F', height='62', weight='67', teacher_id='261')>
<Participant(id=33, name='Felicia Thomas', date_of_birth='1982-12-18', gender='F', height='59', weight='70', teacher_id='221')>
<Participant(id=34, name='Michael Ramos', date_of_birth='2001-01-02', gender='F', height='62', weight='89', teacher_id='104')>
<Participant(id=35, name='Harold Smith', date_of_birth='1960-03-21', gender='M', height='54', weight='53', teacher_id='64')>
<Participant(id=36, name='Emma Quinn', date_of_birth='2002-01-23', gender='M', height='50', weight='109', teacher_id='39')>
<Participant(id=37, name='Jennifer Mcguire', date_of_birth='1988-09-29', gender='M', height='52', weight='119', teacher_id='62')>
<Participant(id=38, name='Isabella Hawkins', date_of_birth='1973-01-06', gender='F', height='67', weight='107', teacher_id='170')>
<Participant(id=39, name='April Conway', date_of_birth='1965-03-09', gender='F', height='61', weight='135', teacher_id='247')>
<Participant(id=40, name='Alexandra Jones', date_of_birth='1971-05-09', gender='M', height='59', weight='62', teacher_id='227')>
<Participant(id=41, name='Barbara Carter', date_of_birth='1996-01-01', gender='M', height='69', weight='84', teacher_id='65')>
```

Update competition:

```
Enter your choice:15
Enter id:40
Enter name of the competition or nothing:Ball throwring
Enter type of sport or nothing:
Enter organizer's name or nothing:
Enter budget or 0:0
Enter country or nothing:Canada
Enter year:2018
Update competition is successful
```

Delete competition:

```
Enter your choice:18
Enter id:56
Delete is successful
```

Додаток 11:

Приклад фільтрації даних з веденням значень фільтрації користувачем:

```
Enter your choice:20
Enter sport:Athletics
Enter min height in inches:40
Enter max height in inches:80
(40, 'Ball throwring', 'Mrs. Robin Martin', 70)
(250, '100M', 'Brianna Armstrong', 62)
(294, 'Hammer Throw', 'April Morales', 72)
(62, 'Shot Put', 'Ms. Hailey Garcia', 59)
(277, '5000M Team', 'Chelsey Richardson', 64)
(38, '400M', 'Glenn Collins', 78)
(315, 'Shot Put', 'Mason Le', 61)
(304, 'Long Jump', 'Nicholas Jackson', 51)
(261, '200M Hurdles', 'Daniel Le', 58)
(248, '100M', 'John Jordan', 53)
(298, 'High Jump', 'John Jordan', 53)
(303, 'Long Jump', 'Meghan Gonzalez', 68)
(33, '110M Hurdles', 'Susan Guzman', 68)
(51, 'Long Jump', 'Brett Woods', 67)
(291, 'Discus Throw', 'Jordan Sherman', 74)
(312, 'Pole Vault', 'Melissa Miller', 77)
(289, '800M', 'Jennifer Morgan', 53)
(278, '5000M Team', 'Jennifer Morgan', 53)
(255, '1500M', 'Sean Osborne', 53)
(43, '800M', 'Ronald Rivera', 68)
```

Додаток 12:

Генерація даних:

```
Enter your choice:21
Enter number for generate:30
Generated
```

Приклад генерації:

	id [PK] integer	name text	date_of_birth date	gender text	height integer	weight integer	teacher_id integer
423	439	Brian Johnson	1965-03-02	M	71	97	10
424	440	Gina Cook	1986-10-20	M	57	47	216
425	441	Noah Gentry	1974-11-25	M	82	123	25
426	442	Angela Hansen	1986-06-14	M	80	47	300
427	443	Victoria Warren	1995-11-12	M	61	131	45
428	444	Claudia Green	1972-12-10	M	50	121	287
429	445	Erik Richmond PhD	1988-11-13	F	53	91	193
430	446	James Burke	1972-08-04	M	66	126	163
431	447	Crystal Williams	1996-09-28	F	60	139	72

Додаток 13:

Приклад статичного аналізу даних з наступним виведенням результату у вигляді графіку:

```
Enter your choice:25
Enter id:127
```

Графік:

