1. Код программы:

dao.py:

*import* sqlite3

class SportsEventDao:

def **\_\_init\_\_**(*self*, *db\_file*):

*self*.conn = sqlite3.connect(*db\_file*)

*self*.cursor = *self*.conn.cursor()

*self*.create\_tables()

def **create\_tables**(*self*):

*# Удаление существующей таблицы sports\_events, если она есть*

*self*.cursor.execute('DROP TABLE IF EXISTS sports\_events')

*# Создание таблицы sports\_events с правильной структурой*

*self*.cursor.execute('''

CREATE TABLE sports\_events (

id INTEGER PRIMARY KEY AUTOINCREMENT,

date\_time TEXT,

sport\_type TEXT,

coach TEXT,

category TEXT,

participants\_count INTEGER

)

''')

*# Удаление существующей таблицы event\_locations, если она есть*

*self*.cursor.execute('DROP TABLE IF EXISTS event\_locations')

*# Создание таблицы event\_locations*

*self*.cursor.execute('''

CREATE TABLE event\_locations (

id INTEGER PRIMARY KEY AUTOINCREMENT,

event\_id INTEGER,

location TEXT,

FOREIGN KEY (event\_id) REFERENCES sports\_events(id)

)

''')

*self*.conn.commit()

def **insert\_event**(*self*, *event*):

*self*.cursor.execute('''

INSERT INTO sports\_events (date\_time, sport\_type, coach, category, participants\_count)

VALUES (?, ?, ?, ?, ?)

''', *event*)

*self*.conn.commit()

def **insert\_event\_location**(*self*, *event\_id*, *location*):

*self*.cursor.execute('''

INSERT INTO event\_locations (event\_id, location)

VALUES (?, ?)

''', (*event\_id*, *location*))

*self*.conn.commit()

def **get\_all\_events**(*self*):

*self*.cursor.execute('SELECT \* FROM sports\_events')

*return* *self*.cursor.fetchall()

def **get\_event\_locations**(*self*):

*try*:

*# Выполнение запроса для получения событий и их местоположений*

*self*.cursor.execute('''

SELECT se.id, se.sport\_type, el.location

FROM sports\_events se

LEFT JOIN event\_locations el ON se.id = el.event\_id

''')

results = *self*.cursor.fetchall()

*if* not results:

print("Нет данных для отображения.")

*return* results

*except* sqlite3.Error *as* e:

print(f"Ошибка при выполнении запроса: {e}")

*return* []

def **get\_team\_sports**(*self*):

*# Параметризованный запрос не требуется, так как нет пользовательского ввода*

*self*.cursor.execute('SELECT \* FROM sports\_events WHERE category = "Командные"')

*return* *self*.cursor.fetchall()

def **get\_events\_with\_more\_than\_10\_participants**(*self*):

*# Параметризованный запрос не требуется, так как нет пользовательского ввода*

*self*.cursor.execute('SELECT \* FROM sports\_events WHERE participants\_count > 10')

*return* *self*.cursor.fetchall()

def **close**(*self*):

*self*.conn.close()

main.py:

*from* dao *import* SportsEventDao

*import* sqlite3

*# Инициализация DAO*

dao = SportsEventDao('sports.db')

*# Вставка данных*

data = [

("2024-03-01 09:00", "Футбол", "Иванов И.И.", "Командные", 22),

("2024-03-01 10:00", "Баскетбол", "Петрова А.В.", "Командные", 10),

("2024-03-01 11:00", "Теннис", "Сидоров О.П.", "Индивидуальные", 2),

("2024-03-01 13:00", "Плавание", "Кузнецова Е.В.", "Индивидуальные", 8),

("2024-03-02 09:00", "Волейбол", "Морозов Д.С.", "Командные", 12),

("2024-03-02 10:00", "Бег", "Волкова Т.Н.", "Индивидуальные", 15),

("2024-03-02 12:00", "Хоккей", "Соколов А.М.", "Командные", 12),

("2024-03-03 09:00", "Шахматы", "Козлов П.Р.", "Индивидуальные", 2),

("2024-03-03 11:00", "Бадминтон", "Лебедева И.М.", "Индивидуальные", 4),

("2024-03-03 13:00", "Регби", "Павлова С.А.", "Командные", 30),

("2024-03-04 10:00", "Гимнастика", "Егорова В.Д.", "Индивидуальные", 5),

("2024-03-04 11:00", "Бокс", "Андреев К.П.", "Индивидуальные", 2),

("2024-03-04 13:00", "Фехтование", "Зайцева М.В.", "Индивидуальные", 2),

("2024-03-05 09:00", "Легкая атлетика", "Титова Ю.С.", "Индивидуальные", 20),

("2024-03-05 10:00", "Гандбол", "Семенов Г.А.", "Командные", 14),

("2024-03-05 12:00", "Каратэ", "Орлова Н.Н.", "Индивидуальные", 2),

("2024-03-06 09:00", "Йога", "Антонова Е.В.", "Индивидуальные", 10),

("2024-03-06 11:00", "Пауэрлифтинг", "Филиппов Д.С.", "Индивидуальные", 6),

("2024-03-06 13:00", "Скалолазание", "Максимова О.Ю.", "Индивидуальные", 8),

("2024-03-07 10:00", "Сноубординг", "Комаров А.И.", "Индивидуальные", 5)

]

*for* event *in* data:

*try*:

dao.insert\_event(event)

*except* sqlite3.IntegrityError:

print(f"Запись {event} уже существует и не будет добавлена.")

*# Вставка данных о местоположении событий*

locations = [

(1, "Стадион Лужники"),

(2, "Баскетбольный зал Олимпийский"),

(3, "Теннисный корт Центральный"),

*# Добавьте остальные местоположения*

]

*for* event\_id, location *in* locations:

*try*:

dao.insert\_event\_location(event\_id, location)

*except* sqlite3.IntegrityError:

print(f"Местоположение для события {event\_id} уже существует и не будет добавлено.")

*# Запрос 1: Вывод всех данных*

print("Все данные:")

*for* row *in* dao.get\_all\_events():

print(row)

*# Запрос 2: Вывод всех командных видов спорта*

print("\nКомандные виды спорта:")

*for* row *in* dao.get\_team\_sports():

print(row)

*# Запрос 3: Вывод всех событий с количеством участников больше 10*

print("\nСобытия с количеством участников больше 10:")

*for* row *in* dao.get\_events\_with\_more\_than\_10\_participants():

print(row)

*# Запрос 4: Вывод всех событий с их местоположениями*

print("\nСобытия и их местоположения:")

*for* row *in* dao.get\_event\_locations():

print(row)

*# Закрытие соединения*

dao.close()

Результат работы кода:



