

Практическое задание № 15

Наименование: составление программ для работы с базами данных в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.

Задача:

Приложение ГРУЗОВЫЕ ПЕРЕВОЗКИ для некоторой организации. БД должна содержать таблицу Перевозки со следующей структурой записи: маршрут, фамилия водителя, даты отправки и прибытия, масса груза.

```
import sqlite3

def create_table():
    conn = sqlite3.connect('transport.db')
    try:
        cursor = conn.cursor()

        cursor.execute('DROP TABLE IF EXISTS Transport')

        cursor.execute('CREATE TABLE IF NOT EXISTS Transport
                        (route TEXT,
                         driver_lastname TEXT,
                         departure_date TEXT,
                         arrival_date TEXT,
                         cargo_weight REAL)')

        conn.commit()
    except sqlite3.Error as e:
        print("Error while working with the database:", e)
    finally:
        conn.close()

def insert_data(info):
    conn = sqlite3.connect('transport.db')
    try:
        data_view = "\n".join(str(i) for i in info)

        cursor = conn.cursor()
        cursor.executemany('INSERT INTO Transport VALUES (?,
?, ?, ?, ?)', info)
        conn.commit()
        print(f"Данные: \n'{data_view}'\n\tуспешно добавлены в
базу данных.\n")
    except sqlite3.Error as e:
        print("Ошибка при добавлении данных:", e)
```

```
finally:
    conn.close()

def display_all():
    conn = sqlite3.connect('transport.db')
    try:
        cursor = conn.cursor()
        cursor.execute(''SELECT * FROM Transport'')
        rows = cursor.fetchall()
        for row in rows:
            print(row)
    except sqlite3.Error as e:
        print("Ошибка при выводе данных:", e)
    finally:
        conn.close()

def search_by_driver(driver):
    conn = sqlite3.connect('transport.db')
    try:
        cursor = conn.cursor()
        cursor.execute(''SELECT * FROM Transport WHERE
driver_lastname=?'', (driver,))
        rows = cursor.fetchall()
        for row in rows:
            print(row)
    except sqlite3.Error as e:
        print("Ошибка при поиске данных:", e)
    finally:
        conn.close()

def search_by_weight(weight, op=="="):
    conn = sqlite3.connect('transport.db')
    try:
        cursor = conn.cursor()
        cursor.execute(f''SELECT * FROM Transport WHERE
cargo_weight{op}?'', (weight,))
        rows = cursor.fetchall()
        for row in rows:
            print(row)
    except sqlite3.Error as e:
        print("Ошибка при поиске данных:", e)
    finally:
        conn.close()

def delete_by_route(route):
    conn = sqlite3.connect('transport.db')
    try:
        cursor = conn.cursor()
        cursor.execute(''DELETE FROM Transport WHERE
```

```

route=?'', (route,))
    conn.commit()
    print(f"Запись по маршруту '{route}' успешно удалена.")
except sqlite3.Error as e:
    print("Ошибка при удалении записи:", e)
finally:
    conn.close()

def update_weight_for_route(route, new_weight):
    conn = sqlite3.connect('transport.db')
    try:
        cursor = conn.cursor()

        cursor.execute('''SELECT cargo_weight FROM Transport
WHERE route=?'', (route,))

        old_weight = int(cursor.fetchone()[0])

        cursor.execute('''UPDATE Transport SET cargo_weight=?
WHERE route=?'', (new_weight, route))

        conn.commit()
        print(f"Запись по маршруту '{route}' успешно обновлена.
Вес {old_weight} -> {new_weight}")
    except sqlite3.Error as e:
        print("Ошибка при обновлении записи:", e)
    finally:
        conn.close()

create_table()

info = [
    ('Москва-Санкт-Петербург', 'Иванов', '2024-04-01', '2024-04-
03', 5000),
    ('Москва-Казань', 'Петров', '2024-04-05', '2024-04-07',
7000),
    ('Санкт-Петербург-Казань', 'Тополев', '2024-04-10', '2024-
04-12', 6000),
    ('Казань-Сочи', 'Хлеб', '2024-04-15', '2024-04-17', 5500),
    ('Сочи-Владивосток', 'Дорн', '2024-04-20', '2024-04-25',
8000),
    ('Владивосток-Красноярск', 'Акрель', '2024-04-28', '2024-05-
01', 4500),
    ('Красноярск-Екатеринбург', 'Королёв', '2024-05-05', '2024-
05-08', 7000),
    ('Екатеринбург-Новосибирск', 'Симоненко', '2024-05-10',
'2024-05-12', 6500),
    ('Новосибирск-Омск', 'Ват', '2024-05-15', '2024-05-18',
5500),
    ('Омск-Тюмень', 'Ульяр', '2024-05-20', '2024-05-23', 7500)
]

```

```

]

insert_data(info)

print("Таблица целиком:")
display_all()

# ----- ПОИСК -----

print("\nПоиск по фамилии водителя 'Иванов':")
search_by_driver('Иванов')

print("\nПоиск по фамилии водителя 'Хлеб':")
search_by_driver('Хлеб')

print("\nПоиск по фамилии водителя 'Ват':")
search_by_driver('Ват')

print("\nПоиск груза по весу меньше чем 6500:")
search_by_weight(6500, "<")

print("\nПоиск груза по весу больше чем 6500:")
search_by_weight(6500, ">")

print("\nПоиск груза по весу равному 6500:")
search_by_weight(6500)

# ----- удаление -----

print("\nУдаление записи по маршруту 'Москва-Санкт-Петербург':")
delete_by_route('Москва-Санкт-Петербург')

print("\nУдаление записи по маршруту 'Новосибирск-Омск':")
delete_by_route('Новосибирск-Омск')

print("\nУдаление записи по маршруту 'Казань-Сочи':")
delete_by_route('Казань-Сочи')

print("\nТаблица после удаления:")
display_all()

# ----- обновление -----

print("\nОбновление данных для маршрута 'Москва-Казань':")
update_weight_for_route('Москва-Казань', 8000)

print("\nОбновление данных для маршрута 'Сочи-Владивосток':")
update_weight_for_route('Сочи-Владивосток', 9400)

print("\nОбновление данных для маршрута 'Омск-Тюмень':")
update_weight_for_route('Омск-Тюмень', 3000)

```

```
print("\nТаблица после обновления:")
display_all()
```

Протокол работы программы:

Данные:

```
('Москва-Санкт-Петербург', 'Иванов', '2024-04-01', '2024-04-03', 5000)
('Москва-Казань', 'Петров', '2024-04-05', '2024-04-07', 7000)
('Санкт-Петербург-Казань', 'Тополев', '2024-04-10', '2024-04-12', 6000)
('Казань-Сочи', 'Хлеб', '2024-04-15', '2024-04-17', 5500)
('Сочи-Владивосток', 'Дорн', '2024-04-20', '2024-04-25', 8000)
('Владивосток-Красноярск', 'Акрель', '2024-04-28', '2024-05-01', 4500)
('Красноярск-Екатеринбург', 'Королёв', '2024-05-05', '2024-05-08', 7000)
('Екатеринбург-Новосибирск', 'Симоненко', '2024-05-10', '2024-05-12', 6500)
('Новосибирск-Омск', 'Ват', '2024-05-15', '2024-05-18', 5500)
('Омск-Тюмень', 'Ульяр', '2024-05-20', '2024-05-23', 7500)'
```

успешно добавлены в базу данных.

Таблица целиком:

```
('Москва-Санкт-Петербург', 'Иванов', '2024-04-01', '2024-04-03', 5000.0)
('Москва-Казань', 'Петров', '2024-04-05', '2024-04-07', 7000.0)
('Санкт-Петербург-Казань', 'Тополев', '2024-04-10', '2024-04-12', 6000.0)
('Казань-Сочи', 'Хлеб', '2024-04-15', '2024-04-17', 5500.0)
('Сочи-Владивосток', 'Дорн', '2024-04-20', '2024-04-25', 8000.0)
('Владивосток-Красноярск', 'Акрель', '2024-04-28', '2024-05-01', 4500.0)
('Красноярск-Екатеринбург', 'Королёв', '2024-05-05', '2024-05-08', 7000.0)
('Екатеринбург-Новосибирск', 'Симоненко', '2024-05-10', '2024-05-12', 6500.0)
('Новосибирск-Омск', 'Ват', '2024-05-15', '2024-05-18', 5500.0)
('Омск-Тюмень', 'Ульяр', '2024-05-20', '2024-05-23', 7500.0)
```

Поиск по фамилии водителя 'Иванов':

```
('Москва-Санкт-Петербург', 'Иванов', '2024-04-01', '2024-04-03', 5000.0)
```

Поиск по фамилии водителя 'Хлеб':

```
('Казань-Сочи', 'Хлеб', '2024-04-15', '2024-04-17', 5500.0)
```

Поиск по фамилии водителя 'Ват':

```
('Новосибирск-Омск', 'Ват', '2024-05-15', '2024-05-18', 5500.0)
```

Поиск груза по весу меньше чем 6500:

('Москва-Санкт-Петербург', 'Иванов', '2024-04-01', '2024-04-03', 5000.0)
('Санкт-Петербург-Казань', 'Тополев', '2024-04-10', '2024-04-12', 6000.0)
('Казань-Сочи', 'Хлеб', '2024-04-15', '2024-04-17', 5500.0)
('Владивосток-Красноярск', 'Акрель', '2024-04-28', '2024-05-01', 4500.0)
('Новосибирск-Омск', 'Ват', '2024-05-15', '2024-05-18', 5500.0)

Поиск груза по весу больше чем 6500:

('Москва-Казань', 'Петров', '2024-04-05', '2024-04-07', 7000.0)
('Сочи-Владивосток', 'Дорн', '2024-04-20', '2024-04-25', 8000.0)
('Красноярск-Екатеринбург', 'Королёв', '2024-05-05', '2024-05-08', 7000.0)
('Омск-Тюмень', 'Ульяр', '2024-05-20', '2024-05-23', 7500.0)

Поиск груза по весу равному 6500:

('Екатеринбург-Новосибирск', 'Симоненко', '2024-05-10', '2024-05-12', 6500.0)

Удаление записи по маршруту 'Москва-Санкт-Петербург':

Запись по маршруту 'Москва-Санкт-Петербург' успешно удалена.

Удаление записи по маршруту 'Новосибирск-Омск':

Запись по маршруту 'Новосибирск-Омск' успешно удалена.

Удаление записи по маршруту 'Казань-Сочи':

Запись по маршруту 'Казань-Сочи' успешно удалена.

Таблица после удаления:

('Москва-Казань', 'Петров', '2024-04-05', '2024-04-07', 7000.0)
('Санкт-Петербург-Казань', 'Тополев', '2024-04-10', '2024-04-12', 6000.0)
('Сочи-Владивосток', 'Дорн', '2024-04-20', '2024-04-25', 8000.0)
('Владивосток-Красноярск', 'Акрель', '2024-04-28', '2024-05-01', 4500.0)
('Красноярск-Екатеринбург', 'Королёв', '2024-05-05', '2024-05-08', 7000.0)
('Екатеринбург-Новосибирск', 'Симоненко', '2024-05-10', '2024-05-12', 6500.0)
('Омск-Тюмень', 'Ульяр', '2024-05-20', '2024-05-23', 7500.0)

Обновление данных для маршрута 'Москва-Казань':

Запись по маршруту 'Москва-Казань' успешно обновлена. Вес 7000 -> 8000

Обновление данных для маршрута 'Сочи-Владивосток':

Запись по маршруту 'Сочи-Владивосток' успешно обновлена. Вес 8000 -> 9400

Обновление данных для маршрута 'Омск-Тюмень':

Запись по маршруту 'Омск-Тюмень' успешно обновлена. Вес 7500 -> 3000

Таблица после обновления:

```
('Москва-Казань', 'Петров', '2024-04-05', '2024-04-07', 8000.0)
('Санкт-Петербург-Казань', 'Тополев', '2024-04-10', '2024-04-12', 6000.0)
('Сочи-Владивосток', 'Дорн', '2024-04-20', '2024-04-25', 9400.0)
('Владивосток-Красноярск', 'Акрель', '2024-04-28', '2024-05-01', 4500.0)
('Красноярск-Екатеринбург', 'Королёв', '2024-05-05', '2024-05-08', 7000.0)
('Екатеринбург-Новосибирск', 'Симоненко', '2024-05-10', '2024-05-12', 6500.0)
('Омск-Тюмень', 'Ульяр', '2024-05-20', '2024-05-23', 3000.0)
```

Вывод: в процессе выполнения практического занятия выработал навыки составления программ, работы с БД в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация, программного кода. Готовые программные коды выложены на GitHub.