

Практическое задание №15

Тема: Составление программ для работы с базами данных в IDE PyCharm Community.

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

Постановка задачи №1:

В матрице найти минимальный и максимальные элементы. Тип алгоритма: линейный

Текст программы:

```
Поиск данных sql-запрос 1
with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute('SELECT * FROM expences WHERE place_trip ==
"Ростов-на-Дону"')
    result_1 = cur.fetchall()
    print(result_1)

with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute('SELECT * FROM expences WHERE last_name LIKE "Б% "')
    result_2 = cur.fetchall()
    print(result_2)

with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute('SELECT * FROM expences WHERE expence_type = "Питание"')
    result_3 = cur.fetchall()
    print(result_3)

sql - 2
with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute("UPDATE expences SET expence_type='Еда' WHERE
expence_type=='Питание'")
    cur.execute("SELECT * FROM expences")
    result_4 = cur.fetchall()
    print(result_4)

with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute("UPDATE expences SET place_trip='Ростов-на-Дону'
WHERE place_trip!='Ростов-на-Дону'")
    cur.execute("SELECT * FROM expences")
    result_5 = cur.fetchall()
    print(result_5)
```

```

with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute("UPDATE expences SET payment = payment+5000
WHERE order_number = 2")
    cur.execute("SELECT * FROM expences WHERE order_number = 2")
    result_6 = cur.fetchall()
    print(result_6)

with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute("DELETE FROM expences WHERE order_number == 1")
    cur.execute("SELECT * FROM expences")
    result_7 = cur.fetchall()
    print(result_7)

with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute("DELETE FROM expences WHERE order_number == 1")
    cur.execute("SELECT * FROM expences")
    result_7 = cur.fetchall()
    print(result_7)

with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute("DELETE FROM expences WHERE last_name LIKE 'Б%")
    cur.execute("SELECT * FROM expences")
    result_8 = cur.fetchall()
    print(result_8)

with sq.connect('expense_items.db') as con:
    cur = con.cursor()
    cur.execute("DELETE FROM expences WHERE expence_type = 'Транспорт'")
    cur.execute("SELECT * FROM expences")
    result_9 = cur.fetchall()
    print(result_9)

```

Протокол работы программы:

```

[(3, 'Артемов', 'Ростов-на-Дону', 7600.0, 4300.0, 'Питание', 3700.0)]
[(1, 'Белов', 'Москва', 6004.0, 3000.0, 'Транспорт', 3500.0), (4, 'Бушкин', 'Владивосток', 8920.0,
5600.0, 'Жилье', 5600.0), (5, 'Бостоевский', 'Мурманск', 21440.0, 16530.0, 'Питание', 4400.0)]
[(3, 'Артемов', 'Ростов-на-Дону', 7600.0, 4300.0, 'Питание', 3700.0), (5, 'Бостоевский',
'Мурманск', 21440.0, 16530.0, 'Питание', 4400.0), (7, 'Месси', 'Ижевск', 34020.0, 21453.0,
'Питание', 12000.0)]
[(1, 'Белов', 'Москва', 6004.0, 3000.0, 'Транспорт', 3500.0), (2, 'Иванов', 'Архангельск', 5400.0,
2800.0, 'Транспорт', 3200.0), (3, 'Артемов', 'Ростов-на-Дону', 7600.0, 4300.0, 'Еда', 3700.0), (4,
'Бушкин', 'Владивосток', 8920.0, 5600.0, 'Жилье', 5600.0), (5, 'Бостоевский', 'Мурманск',
21440.0, 16530.0, 'Еда', 4400.0), (6, 'Гризман', 'Орел', 21900.0, 17200.0, 'Транспорт', 6800.0), (7,
'Месси', 'Ижевск', 34020.0, 21453.0, 'Еда', 12000.0), (8, 'Роналдо', 'Пермь', 84880.0, 64220.0,
'Транспорт', 34020.0), (9, 'Кокорин', 'Санкт-Петербург', 50020.0, 27490.0, 'Транспорт', 30420.0),
(10, 'Федоров', 'Ростов', 50241.0, 2750.0, 'Жилье', 3540.0)]
[(1, 'Белов', 'Ростов-на-Дону', 6004.0, 3000.0, 'Транспорт', 3500.0), (2, 'Иванов',
'Ростов-на-Дону', 5400.0, 2800.0, 'Транспорт', 3200.0), (3, 'Артемов', 'Ростов-на-Дону', 7600.0,
4300.0, 'Еда', 3700.0), (4, 'Бушкин', 'Ростов-на-Дону', 8920.0, 5600.0, 'Жилье', 5600.0), (5,

```

'Бостоевский', 'Ростов-на-Дону', 21440.0, 16530.0, 'Еда', 4400.0), (6, 'Гризман', 'Ростов-на-Дону', 21900.0, 17200.0, 'Транспорт', 6800.0), (7, 'Месси', 'Ростов-на-Дону', 34020.0, 21453.0, 'Еда', 12000.0), (8, 'Роналдо', 'Ростов-на-Дону', 84880.0, 64220.0, 'Транспорт', 34020.0), (9, 'Кокорин', 'Ростов-на-Дону', 50020.0, 27490.0, 'Транспорт', 30420.0), (10, 'Федоров', 'Ростов-на-Дону', 50241.0, 2750.0, 'Жилье', 3540.0)]

[(2, 'Иванов', 'Ростов-на-Дону', 10400.0, 2800.0, 'Транспорт', 3200.0)]

[(2, 'Иванов', 'Ростов-на-Дону', 10400.0, 2800.0, 'Транспорт', 3200.0), (3, 'Артемов', 'Ростов-на-Дону', 7600.0, 4300.0, 'Еда', 3700.0), (4, 'Бушкин', 'Ростов-на-Дону', 8920.0, 5600.0, 'Жилье', 5600.0), (5, 'Бостоевский', 'Ростов-на-Дону', 21440.0, 16530.0, 'Еда', 4400.0), (6, 'Гризман', 'Ростов-на-Дону', 21900.0, 17200.0, 'Транспорт', 6800.0), (7, 'Месси', 'Ростов-на-Дону', 34020.0, 21453.0, 'Еда', 12000.0), (8, 'Роналдо', 'Ростов-на-Дону', 84880.0, 64220.0, 'Транспорт', 34020.0), (9, 'Кокорин', 'Ростов-на-Дону', 50020.0, 27490.0, 'Транспорт', 30420.0), (10, 'Федоров', 'Ростов-на-Дону', 50241.0, 2750.0, 'Жилье', 3540.0)]

[(2, 'Иванов', 'Ростов-на-Дону', 10400.0, 2800.0, 'Транспорт', 3200.0), (3, 'Артемов', 'Ростов-на-Дону', 7600.0, 4300.0, 'Еда', 3700.0), (4, 'Бушкин', 'Ростов-на-Дону', 8920.0, 5600.0, 'Жилье', 5600.0), (5, 'Бостоевский', 'Ростов-на-Дону', 21440.0, 16530.0, 'Еда', 4400.0), (6, 'Гризман', 'Ростов-на-Дону', 21900.0, 17200.0, 'Транспорт', 6800.0), (7, 'Месси', 'Ростов-на-Дону', 34020.0, 21453.0, 'Еда', 12000.0), (8, 'Роналдо', 'Ростов-на-Дону', 84880.0, 64220.0, 'Транспорт', 34020.0), (9, 'Кокорин', 'Ростов-на-Дону', 50020.0, 27490.0, 'Транспорт', 30420.0), (10, 'Федоров', 'Ростов-на-Дону', 50241.0, 2750.0, 'Жилье', 3540.0)]

[(2, 'Иванов', 'Ростов-на-Дону', 10400.0, 2800.0, 'Транспорт', 3200.0), (3, 'Артемов', 'Ростов-на-Дону', 7600.0, 4300.0, 'Еда', 3700.0), (6, 'Гризман', 'Ростов-на-Дону', 21900.0, 17200.0, 'Транспорт', 6800.0), (7, 'Месси', 'Ростов-на-Дону', 34020.0, 21453.0, 'Еда', 12000.0), (8, 'Роналдо', 'Ростов-на-Дону', 84880.0, 64220.0, 'Транспорт', 34020.0), (9, 'Кокорин', 'Ростов-на-Дону', 50020.0, 27490.0, 'Транспорт', 30420.0), (10, 'Федоров', 'Ростов-на-Дону', 50241.0, 2750.0, 'Жилье', 3540.0)]

[(3, 'Артемов', 'Ростов-на-Дону', 7600.0, 4300.0, 'Еда', 3700.0), (7, 'Месси', 'Ростов-на-Дону', 34020.0, 21453.0, 'Еда', 12000.0), (10, 'Федоров', 'Ростов-на-Дону', 50241.0, 2750.0, 'Жилье', 3540.0)]

Process finished with exit code

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления БД, IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub

