

### Практическое задание № 15

**Наименование:** составление программ с SQL базами данных в IDE PyCharm Community

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с SQL базами данных в IDE PyCharm Community.

**Задача.** Приложение НОТАРИАЛЬНАЯ КОНТОРА для некоторой организации. БД должна содержать таблицу Нотариальные услуги со следующей структурой записи: ФИО клиента, услуга, сумма сделки, комиссионные (доход конторы).

```
# Вариант 20
# Приложение НОТАРИАЛЬНАЯ КОНТОРА для некоторой организации.
# БД должна содержать таблицу Нотариальные услуги со следующей структурой
записи: ФИО клиента, услуга, сумма сделки, комиссионные (доход конторы).

import sqlite3
from clients_data import clients_info

def create_connection():
    conn = None
    try:
        conn = sqlite3.connect('notary_office.db')
    except sqlite3.Error as e:
        print(e)
    return conn

def close_connection(conn):
    if conn:
        conn.close()

def create_table(conn):
    try:
        cursor = conn.cursor()
        cursor.execute('''CREATE TABLE IF NOT EXISTS deals (
                        clients_id INTEGER PRIMARY KEY AUTOINCREMENT,
                        clients_fio TEXT NOT NULL,
                        service TEXT NOT NULL,
                        cost INTEGER NOT NULL,
                        commission INTEGER NOT NULL);''')
    except sqlite3.Error as e:
        print(e)

def display_table(conn):
    cur = conn.cursor()
    cur.execute('''SELECT * FROM deals''')

    rows = cur.fetchall()

    for row in rows:
        print(row)

def add_deal(conn, deal):
    sql = '''INSERT INTO deals(clients_fio, service, cost, commission)
VALUES(?,?,?,?)'''
    cur = conn.cursor()
    cur.executemany(sql, deal)
```

```

        conn.commit()
        return cur.lastrowid

def search_deal(conn, query):
    cur = conn.cursor()
    cur.execute("SELECT * FROM deals WHERE clients_fio=? or service=? or
cost=? or commission=?", (query,)*4)
    rows = cur.fetchall()
    for row in rows:
        print(row)
        print()

def delete_deal(conn, id):
    # sql = '''DELETE FROM deals WHERE clients_id=? or clients_fio=? or
service=? or cost=? or commission=?'''
    # cur = conn.cursor()
    # cur.execute(sql, (id,))
    cur = conn.cursor()
    cur.execute("DELETE FROM deals WHERE clients_id=? or clients_fio=? or
service=? or cost=? or commission=?", (id,) * 5)
    conn.commit()

def update_deal(conn, deal):
    sql = 'UPDATE deals SET clients_fio=?, service=?, cost=?, commission=?
WHERE clients_id=?'
    cur = conn.cursor()
    cur.execute(sql, deal)
    conn.commit()

if __name__ == '__main__':
    connection = create_connection()
    if connection is not None:
        create_table(connection)

        add_deal(connection, clients_info)

        search_deal(connection, 'Козлов Владислав Павлович')
        search_deal(connection, 5000)
        search_deal(connection, 'Оформление доверенности')

        delete_deal(connection, 1)
        delete_deal(connection, 50)
        delete_deal(connection, 'Козлов Владислав Павлович')

        update_deal(connection, ('Васильева Екатерина Петровна', 'Оформление
брачного контракта', 5000, 100, 2))
        update_deal(connection, ('Жегалкин Петр Петрович', 'Оформление
страхования', 6000, 150, 4))
        update_deal(connection, ('Майская Драздраперма Дамировна',
'Оформление договора дарения', 16000, 1500, 5))

        display_table(connection)

        close_connection(connection)
    else:
        print("Error! Cannot create the database connection.")

```

### Протокол работы программы:

(7, 'Козлов Владислав Павлович', 'Оформление сделок дарения', 2900, 180)

(4, 'Васильев Игорь Валентинович', 'Оформление брачного контракта', 5000, 100)

(3, 'Федоров Евдокий Даниилович', 'Оформление доверенности', 1900, 100)

(2, 'Васильева Екатерина Петровна', 'Оформление брачного контракта', 5000, 100)

(3, 'Федоров Евдокий Даниилович', 'Оформление доверенности', 1900, 100)

(4, 'Жегалкин Петр Петрович', 'Оформление страхования', 6000, 150)

(5, 'Май Драздраперма Дамировна', 'Оформление договора дарения', 16000, 1500)

(8, 'Кузнецова Мария Владимировна', 'Оформление сделок дарения', 2900, 180)

### File clients\_data.py:

```
clients_info = [  
    1 ('Иванова Анна Сергеевна', 'Заверение подписи', 2200, 70),  
    2 ('Соколова Вера Николаевна', 'Составление нотариального завещания', 1800, 80),  
    3 ('Федоров Евдокий Даниилович', 'Оформление доверенности', 1900, 100),  
    4 ('Васильев Игорь Валентинович', 'Оформление брачного контракта', 5000, 100),  
    5 ('Смирнов Сергей Игоревич', 'Подтверждение права собственности', 1800, 100),  
    6 ('Михайлов Павел Викторович', 'Оформление сделок кредитования', 3700, 50),  
    7 ('Козлов Владислав Павлович', 'Оформление сделок дарения', 2900, 180),  
    8 ('Кузнецова Мария Владимировна', 'Оформление сделок дарения', 2900, 180),  
    9 ('Новикова Евгения Алексеевна', 'Оформление сделок кредитования', 3700, 50)  
]
```

**Вывод:** в процессе выполнения практического занятия, я выработала навыки составления программ с SQL базами данных в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация, программного кода. Готовые программные коды выложены на GitHub.