

Тема: составление программ с использованием SQL-запросов в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия , алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием SQL-запросов в IDE PyCharm Community.

Постановка задачи.

1. Приложение для туристического агентства ТУР. Таблица Турист должна содержать следующую информацию о клиентах турфирмы: Код клиента, Клиент (Фамилия), Телефон, Название страны, Регион, Продолжительность поездки, Стоимость путёвки

Тип алгоритма: ветвящийся, линейная.

Текст программы 1:

```
PZ_15_1.py > ...
#Приложение для туристического агентства ТУР. Таблица Турист должна
#содержать следующую информацию о клиентах турфирмы: Код клиента,
#Клиент (Фамилия), Телефон, Название страны, Регион, Продолжитель-
#ность поездки, Стоимость путевки

import sqlite3
from sqlite3 import Error

def create_connection(db_file):
    conn = None
    try:
        conn = sqlite3.connect(db_file)
        print(f"Соединение установлено: SQLite версия {sqlite3.version}")
    except Error as e:
        print(e)
    return conn

def create_table(conn):
    try:
        create_table_sql = """
        CREATE TABLE IF NOT EXISTS Турист (
            Код_клиента INTEGER PRIMARY KEY,
            Клиент TEXT NOT NULL,
            Телефон TEXT NOT NULL,
            Название_страны TEXT NOT NULL,
            Регион TEXT NOT NULL,
            Продолжительность_поездки INTEGER NOT NULL,
            Стоимость_путевки REAL NOT NULL
        );
        """
    except:
        print('Ошибка!\nВводите в продолжительности поездки и Стоимости путёвки чи')
    try:
        c = conn.cursor()
        c.execute(create_table_sql)
        print("Таблица создана.")
    except Error as e:
        print(e)

database = "tourism_agency.db"

conn = create_connection(database)
if conn is not None:
    create_table(conn)
```

```
PZ_15_1.py > ...
    create_table(conn)
else:
    print("Ошибка! Невозможно создать соединение с базой данных.")

def add_tourist(conn, tourist):
    sql = ''' INSERT INTO Турист(Клиент, Телефон, Название_страны, Регион, Продолжительность_поездки, Стоимость_путевки)
              VALUES(?,?,?,?,?,?) '''
    cur = conn.cursor()
    cur.execute(sql, tourist)
    conn.commit()
    return cur.lastrowid

def search_tourist(conn, query, params):
    cur = conn.cursor()
    cur.execute(query, params)
    rows = cur.fetchall()
    return rows

def delete_tourist(conn, id):
    sql = 'DELETE FROM Турист WHERE Код_клиента=?'
    cur = conn.cursor()
    cur.execute(sql, (id,))
    conn.commit()

def update_tourist(conn, tourist):
    sql = ''' UPDATE Турист
              SET Клиент = ? ,
                Телефон = ? ,
                Название_страны = ? ,
                Регион = ? ,
                Продолжительность_поездки = ? ,
                Стоимость_путевки = ?
              WHERE Код_клиента = ? '''
    cur = conn.cursor()
    cur.execute(sql, tourist)
    conn.commit()

def main():
    try:
        database = "tourism_agency.db"
```

PZ_15_1.py > main

```
def main():
    conn = create_connection(database)
    if conn is not None:
        create_table(conn)

    while True:
        print("Выберите действие: ")
        print("1. Добавить клиента")
        print("2. Найти клиента")
        print("3. Удалить клиента")
        print("4. Обновить информацию о клиенте")
        print("5. Выйти")

        choice = input("Введите номер действия: ")

        if choice == '1':
            Клиент = input("Введите фамилию клиента: ")
            Телефон = input("Введите телефон клиента: ")
            Название_страны = input("Введите название страны: ")
            Регион = input("Введите регион: ")
            Продолжительность_поездки = int(input("Введите продолжительность по
Стоимость_путевки = float(input("Введите стоимость путевки: "))

            tourist = (Клиент, Телефон, Название_страны, Регион, Продолжительно
add_tourist(conn, tourist)
            print("Клиент добавлен успешно.")

        elif choice == '2':
            print("1. Поиск по фамилии")
            print("2. Поиск по телефону")
            print("3. Поиск по стране")
            sub_choice = input("Введите номер действия: ")

            if sub_choice == '1':
                Клиент = input("Введите фамилию клиента: ")
                query = "SELECT * FROM Турист WHERE Клиент LIKE ?"
                params = ('%' + Клиент + '%',)
                results = search_tourist(conn, query, params)
            elif sub_choice == '2':
                Телефон = input("Введите телефон клиента: ")
                query = "SELECT * FROM Турист WHERE Телефон LIKE ?"
                params = ('%' + Телефон + '%',)
                results = search_tourist(conn, query, params)
```

```

PZ_15_1.py > main
def main():
    elif sub_choice == '3':
        Название_страны = input("Введите название страны: ")
        query = "SELECT * FROM Турист WHERE Название_страны LIKE ?"
        params = ('%' + Название_страны + '%',)
        results = search_tourist(conn, query, params)
    else:
        print("Неверный выбор.")
        continue

    for row in results:
        print(row)

elif choice == '3':
    id = int(input("Введите код клиента для удаления: "))
    delete_tourist(conn, id)
    print("Клиент удален успешно.")

elif choice == '4':
    id = int(input("Введите код клиента для обновления: "))
    Клиент = input("Введите фамилию клиента: ")
    Телефон = input("Введите телефон клиента: ")
    Название_страны = input("Введите название страны: ")
    Регион = input("Введите регион: ")
    Продолжительность_поездки = int(input("Введите продолжительность по
    Стоимость_путевки = float(input("Введите стоимость путевки: "))

    tourist = (Клиент, Телефон, Название_страны, Регион, Продолжительно
    update_tourist(conn, tourist)
    print("Информация о клиенте обновлена.")

elif choice == '5':
    break
else:
    print("Неверный выбор. Пожалуйста, попробуйте снова.")

conn.close()
except:
    print('Ошибка')

if __name__ == '__main__':
    main()

```

Вывод: закрепил и усвоил знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с использованием MySql запросов в IDE PyCharm Community.