Практическое задание No 15 Студент группы ИС-27 Гончаров М. Е

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:

```python
#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 Tourist (
            id_client INTEGER PRIMARY KEY,
            client TEXT NOT NULL,
            phone TEXT NOT NULL,
            country_name TEXT NOT NULL,
            region TEXT NOT NULL,
            trip_duration INTEGER NOT NULL,
            cost_vacation REAL NOT NULL
        );
        """
        c = conn.cursor()
        c.execute(create_table_sql)
        print("Таблица создана.")
    except Error as e:
        print(e)

def add_tourist(conn, tourist):
    sql = ''' INSERT INTO Tourist(client, phone, country_name, region, trip_duration, c
              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()
```

```python
    return rows

def delete_tourist(conn, id_client):
    sql = 'DELETE FROM Tourist WHERE id_client=?'
    cur = conn.cursor()
    cur.execute(sql, (id_client,))
    conn.commit()

def update_tourist(conn, tourist):
    sql = ''' UPDATE Tourist
              SET client = ? ,
                  phone = ? ,
                  country_name = ? ,
                  region = ? ,
                  trip_duration = ? ,
                  cost_vacation = ?
              WHERE id_client = ?'''
    cur = conn.cursor()
    cur.execute(sql, tourist)
    conn.commit()

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

        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.isdigit() and 1 <= int(choice) <= 5:
                    choice = int(choice)
                    if choice == 1:
                        try:
                            client = input("Введите фамилию клиента: ")
                            phone = input("Введите телефон клиента: ")
```

```
            phone = input("Введите телефон клиента: ")
            country_name = input("Введите название страны: ")
            region = input("Введите регион: ")
            trip_duration = int(input("Введите продолжительность поездки: "))
            cost_vacation = float(input("Введите стоимость путевки: "))

            tourist = (client, phone, country_name, region, trip_duration, cost_vacation)
            add_tourist(conn, tourist)
            print("Клиент добавлен успешно.")
        except ValueError:
            print("Ошибка: Некорректный формат данных. Пожалуйста, попробуйте снова.")

    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.isdigit() and 1 <= int(sub_choice) <= 3:
            sub_choice = int(sub_choice)
            if sub_choice == 1:
                client = input("Введите фамилию клиента: ")
                query = "SELECT * FROM Tourist WHERE client LIKE ?"
                params = ('%' + client + '%',)
                results = search_tourist(conn, query, params)
            elif sub_choice == 2:
                phone = input("Введите телефон клиента: ")
                query = "SELECT * FROM Tourist WHERE phone LIKE ?"
                params = ('%' + phone + '%',)
                results = search_tourist(conn, query, params)
            elif sub_choice == 3:
                country_name = input("Введите название страны: ")
                query = "SELECT * FROM Tourist WHERE country_name LIKE ?"
                params = ('%' + country_name + '%',)
                results = search_tourist(conn, query, params)
        else:
            print("Неверный выбор.")
            continue

        for row in results:
            print(row)

    elif choice == 3:
        try:
            id_client = int(input("Введите код клиента для удаления: "))
            delete_tourist(conn, id_client)
```

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