

Практическое занятие №15

Тема: Составление программы для работы с базами данных в Python

Цель: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в Python

Постановка задачи.

Реализовать программу для работы с однотабличной БД. Программа должна обеспечивать функционал по вводу данных в БД, их поиску, удалению и редактированию. При организации поиска, удаления и редактирования использовать WHERE.

Приложение БАНК для отслеживания накапливаемых на счетах клиентов банка сумм: Таблица Клиент должна содержать следующую информацию: Код клиента, Клиент (Ф.И.О.), Периодический платёж, Годовой %, Срок вклада, Пластиковая карта (логическое поле), Конечная сумма.

Текст программы:

"""

Реализовать программу для работы с однотабличной БД. Программа должна обеспечивать функционал по вводу данных в БД, их поиску, удалению и редактированию. При организации поиска, удаления и редактирования использовать WHERE.

Приложение БАНК для отслеживания накапливаемых на счетах клиентов банка сумм: Таблица Клиент должна содержать следующую информацию: Код клиента, Клиент (Ф.И.О.), Периодический платёж, Годовой %, Срок вклада, Пластиковая карта (логическое поле), Конечная сумма.

"""

```
import pymysql
```

```
def inputation(s: str, t: int, d=""):
```

```
    n = "
```

```
    while n == ":
```

```
        n = input(f'{s} {"(" if d != "" else ""}{d}{") " if d != "" else ""}>> ')
```

```
        if t >= 1 and n == ":
```

```
            print(s, 'не может быть пустым\n')
```

```
        if t == 2:
```

```
            try:
```

```
                n = float(n)
```

```
                if n < 0:
```

```
                    print(n, 'не может быть отрицательным')
```

```
                    continue
```

```

except ValueError:
    print(s, 'должно быть числом\n')
    n = ''
elif t == 3:
    if n == 't' or n == 'True' or n == 'true' or n == '1' or n == 'да':
        n = "true"
    elif n == 'f' or n == 'False' or n == 'false' or n == '0' or n == 'нет':
        n = "false"
    else:
        print(s, 'должно быть логическим\n')
        n = ''
else:
    break
return n

```

```

cmds = ['код', 'имя', 'фамилия', 'отчество', 'платёж', 'процент', 'срок', 'карта', 'сумма']
cmdsnm = ['code', 'name', 'sname', 'fname', 'pay', 'percent', 'period', 'card', 'final']

```

```

connection = pymysql.connect(
    host="localhost",
    port=3306,
    user="training",
    password="adfhiu1839",
    database="oaip",
)

```

```

with connection.cursor() as cursor:
    cursor.execute("CREATE TABLE IF NOT EXISTS clients ("
        "code INT PRIMARY KEY AUTO_INCREMENT,"
        "name VARCHAR(32) NOT NULL,"
        "sname VARCHAR(32) NOT NULL,"
        "fname VARCHAR(32),"
        "pay DECIMAL(12,2) NOT NULL,"
        "percent DECIMAL(5,2) NOT NULL,"
        "period INT NOT NULL,"
        "card BOOL NOT NULL,"
        "final DECIMAL (14,2));")
    connection.commit()

```

```

while True:
    print('\n' * 30)
    print('Выберите опцию вводом соответствующей цифры или названия')
    v = input('1. Ввод, 2. Поиск, 3. Удаление, 4. Редактирование\n>> ')
    if v == '1' or v == 'Ввод' or v == 'ввод' or v == '1. Ввод':
        print("\nвведите значения, которые будут занесены в базу данных")
        name = input('имя клиента', 1)
        sname = input('фамилия', 1)
        fname = input('отчество', 0, 'если нет просто нажмите enter')
        pay = input('периодический платёж', 2)

```

```

percent = input('годовой процент', 2)
period = input('срок вклада', 2)
card = input('имеется ли карта', 3)
final = input('конечная сумма', 2)
try:
    with connection.cursor() as cursor:
        insert_query = (f"INSERT INTO clients(name, sname, fname, pay, percent, period, card, final)
VALUES "
                        f"('{name}', '{sname}', '{fname}', {pay}, {percent}, {period}, {card}, {final});")
        cursor.execute(insert_query)
        connection.commit()
        print("успешный ввод, значения занесены в базу данных")
except Exception as ex:
    print("ошибка", ex)
    input("нажмите enter, чтобы продолжить")
elif v == '2' or v == 'Поиск' or v == 'поиск' or v == '2. Поиск':
    while True:
        vv = input("\nвведите * для выборки всей информации из таблицы\nили названия полей по
которым искать через "
                  "\nпробел\n(код, имя, фамилия, отчество, платёж, процент, срок, карта, сумма)\n>> ")
        if vv == "*":
            with connection.cursor() as cursor:
                cursor.execute("SELECT * FROM clients")
                rows = cursor.fetchall()
                for row in rows:
                    print("\nкод клиента:", row[0])
                    print("Ф.И.О.:", row[2], row[1], row[3])
                    print("периодический платёж:", row[4])
                    print("годовой процент:", row[5])
                    print("срок вклада:", row[6])
                    print("карта:", "есть" if row[7] == 1 else "нет")
                    print("конечная сумма:", row[8])
                input("\nнажмите enter, чтобы продолжить")
            break
        else:
            vv = vv.split()
            er = 0
            for i in vv:
                if i not in cmds:
                    er += 1
                    print(i, "нет среди полей, попробуйте снова\n")
            if er > 0:
                continue

            insert_query = "SELECT * FROM clients WHERE "
            c = 0
            for i in range(len(cmds)):
                if cmds[i] in vv:
                    if c > 0:
                        while True:

```

```

vvv = input(f"значение между полями - 1. и, 2. или, 3 ничего (возможно, для "
            f"сравнения полей)\n>> ")
if vvv == '1' or vvv == 'и' or vvv == '1.' or vvv == '1. и':
    opt = ' and '
    break
elif vvv == '2' or vvv == 'или' or vvv == '2.' or vvv == '2. или':
    opt = ' or '
    break
elif vvv == '3' or vvv == 'ничего' or vvv == '3.' or vvv == '3. ничего':
    opt = ' '
    break
else:
    print(vvv + ' не является допустимой опцией, попробуйте снова')
insert_query += opt
while True:
    vvv = input(f"Как искать по полю {cmds[i]}? Где значение - 1. равно, 2. больше, 3.
меньше, "
                f"4. не равно - указанному далее\n>> ")
    if vvv == '1' or vvv == 'равно' or vvv == '1.' or vvv == '1. равно':
        opt = '='
        break
    elif vvv == '2' or vvv == 'больше' or vvv == '2.' or vvv == '2. больше':
        opt = '>'
        break
    elif vvv == '3' or vvv == 'меньше' or vvv == '3.' or vvv == '3. меньше':
        opt = '<'
        break
    elif vvv == '4' or vvv == 'не равно' or vvv == '4.' or vvv == '4. не равно':
        opt = '!='
        break
    else:
        print(vvv + ' не является допустимой опцией, попробуйте снова')
insert_query += f"{cmdsnm[i]} {opt} {'"" if i == 1 or i == 2 or i == 3 else ""}"
insert_query += input(f"Поле {cmds[i]} должно быть {opt} "
                       f"{'чоро' if opt == '>' or opt == '<' else 'чему'}\n>> ")
insert_query += "" if i == 1 or i == 2 or i == 3 else "
c += 1
with connection.cursor() as cursor:
    try:
        cursor.execute(insert_query)
    except Exception as ex:
        print("ошибка", ex)
    rows = cursor.fetchall()
    for row in rows:
        print("\nкод клиента:", row[0])
        print("Ф.И.О.:", row[2], row[1], row[3])
        print("периодический платёж:", row[4])
        print("годовой процент:", row[5])
        print("срок вклада:", row[6])
        print("карта:", "есть" if row[7] == 1 else "нет")

```

```

        print("конечная сумма:", row[8])
        input("\nнажмите enter, чтобы продолжить")
        break
    elif v == '3' or v == 'удаление' or v == 'Удаление' or v == '3. удаление':
        while True:
            vv = input("\nвведите названия полей по которым удалять через "
                        "пробел\n(код, имя, фамилия, отчество, платёж, процент, срок, карта, сумма)\n>> ")
            vv = vv.split()
            er = 0
            for i in vv:
                if i not in cmds:
                    er += 1
                    print(i, "нет среди полей, попробуйте снова\n")
            if er > 0:
                continue

        insert_query = "DELETE FROM clients WHERE "
        c = 0
        for i in range(len(cmds)):
            if cmds[i] in vv:
                if c > 0:
                    while True:
                        vvv = input(f"значение между полями - 1. и, 2. или, 3. ничего (возможно, для "
                                    f"сравнения полей)\n>> ")
                        if vvv == '1' or vvv == 'и' or vvv == '1.' or vvv == '1. и':
                            opt = ' and '
                            break
                        elif vvv == '2' or vvv == 'или' or vvv == '2.' or vvv == '2. или':
                            opt = ' or '
                            break
                        elif vvv == '3' or vvv == 'ничего' or vvv == '3.' or vvv == '3. ничего':
                            opt = ' '
                            break
                        else:
                            print(vvv + ' не является допустимой опцией, попробуйте снова')
                    insert_query += opt
                while True:
                    vvv = input(f"Как искать по полю {cmds[i]}? Где значение - 1. равно, 2. больше, 3.
меньше, "
                                f"4. не равно - указанному далее\n>> ")
                    if vvv == '1' or vvv == 'равно' or vvv == '1.' or vvv == '1. равно':
                        opt = '= '
                        break
                    elif vvv == '2' or vvv == 'больше' or vvv == '2.' or vvv == '2. больше':
                        opt = '> '
                        break
                    elif vvv == '3' or vvv == 'меньше' or vvv == '3.' or vvv == '3. меньше':
                        opt = '< '
                        break
                    elif vvv == '4' or vvv == 'не равно' or vvv == '4.' or vvv == '4. не равно':

```

```

        opt = '!= '
        break
    else:
        print(vvv + ' не является допустимой опцией, попробуйте снова')
        insert_query += f"{cmdsnm[i]} {opt} {'\" if i == 1 or i == 2 or i == 3 else '\"}"
        insert_query += input(f'Поле {cmds[i]} должно быть {opt} '
                               f'{'<' if opt == '>' or opt == '<' else 'чему'}\n>> ')
        insert_query += "\" if i == 1 or i == 2 or i == 3 else \"
        c += 1

try:
    with connection.cursor() as cursor:
        cursor.execute(insert_query)
        connection.commit()
    input('успешно\nнажмите enter чтобы продолжить')
except Exception as ex:
    print("ошибка", ex)
    break

elif v == '4' or v == 'редактирование' or v == 'Редактирование' or v == '4. редактирование':
    while True:
        vv = input("\nкакое поле обновить "
                   "пробел\n(код, имя, фамилия, отчество, платёж, процент, срок, карта, сумма)\n>> ")
        if vv not in cmds:
            print(vv, "нет среди полей, попробуйте снова\n")
            continue

        vvv = ""
        for i in range(len(cmds)):
            if vv == cmds[i]:
                vvv = cmdsnm[i]
                break

        insert_query = f"UPDATE clients SET {vvv} = "
        vv = input(f"\nновое значение поля {vv}\n>> ")
        insert_query += f"{'{vv}' WHERE "
        obr = len(insert_query) - 7

        vv = input("\nвведите названия полей по которым изменять через "
                   "пробел\n(код, имя, фамилия, отчество, платёж, процент, срок, карта, сумма)\n>> ")
        vv = vv.split()
        er = 0
        for i in vv:
            if i not in cmds:
                er += 1
            print(i, "нет среди полей, попробуйте снова\n")
        if er > 0:
            continue

        c = 0
        for i in range(len(cmds)):
            if cmds[i] in vv:

```

```

if c > 0:
    while True:
        vvv = input(f"значение между полями - 1. и, 2. или, 3 ничего (возможно, для "
                    f"сравнения полей)\n>> ")
        if vvv == '1' or vvv == 'и' or vvv == '1.' or vvv == '1. и':
            opt = 'and'
            break
        elif vvv == '2' or vvv == 'или' or vvv == '2.' or vvv == '2. или':
            opt = 'or'
            break
        elif vvv == '3' or vvv == 'ничего' or vvv == '3.' or vvv == '3. ничего':
            opt = ''
            break
        else:
            print(vvv + ' не является допустимой опцией, попробуйте снова')
        insert_query += opt
    while True:
        vvv = input(f"Как искать по полю {cmds[i]}? Где значение - 1. равно, 2. больше, 3.
меньше, "
                    f"4. не равно - указанному далее\n>> ")
        if vvv == '1' or vvv == 'равно' or vvv == '1.' or vvv == '1. равно':
            opt = '='
            break
        elif vvv == '2' or vvv == 'больше' or vvv == '2.' or vvv == '2. больше':
            opt = '>'
            break
        elif vvv == '3' or vvv == 'меньше' or vvv == '3.' or vvv == '3. меньше':
            opt = '<'
            break
        elif vvv == '4' or vvv == 'не равно' or vvv == '4.' or vvv == '4. не равно':
            opt = '!='
            break
        else:
            print(vvv + ' не является допустимой опцией, попробуйте снова')
        insert_query += f"{cmdsnm[i]} {opt} {'" if i == 1 or i == 2 or i == 3 else '"}'
        insert_query += input(f"Поле {cmds[i]} должно быть {opt} "
                            f"{'чоро' if opt == '>' or opt == '<' else 'чему'}\n>> ")
        insert_query += '" if i == 1 or i == 2 or i == 3 else "'
        c += 1
    new_insert = "SELECT * FROM clients" + insert_query[obr:]
    try:
        with connection.cursor() as cursor:
            cursor.execute(insert_query)
    except Exception as ex:
        print("ошибка", ex)
    rows = cursor.fetchall()
    for row in rows:
        print("\nкод клиента:", row[0])

```

```

print("Ф.И.О.:", row[2], row[1], row[3])
print("периодический платёж:", row[4])
print("годовой процент:", row[5])
print("срок вклада:", row[6])
print("карта:", "есть" if row[7] == 1 else "нет")
print("конечная сумма:", row[8])

cursor.execute(insert_query)
connection.commit()
input("\nуспешно\nнажмите enter чтобы продолжить')
except Exception as ex:
    print("ошибка", ex)
break

```

Протокол работы:

Выберите опцию вводом соответствующей цифры или названия

1. Ввод, 2. Поиск, 3. Удаление, 4. Редактирование

>> 2

введите * для выборки всей информации из таблицы

или названия полей по которым искать через пробел

(код, имя, фамилия, отчество, платёж, процент, срок, карта, сумма)

>> *

код клиента: 6

Ф.И.О.: Огуркин Иван Денисович

периодический платёж: 10.00

годовой процент: 3.00

срок вклада: 10

карта: есть

конечная сумма: 13.00

код клиента: 7

Ф.И.О.: Ктошин Пётр Михайлович

периодический платёж: 1000.00

годовой процент: 13.00

срок вклада: 24

карта: есть

конечная сумма: 1260.00

нажмите enter, чтобы продолжить

Вывод: в процессе выполнения практических заданий выработал навыки составления программ для работы с базами данных в IDE Pycharm. Были использованы конструкции cursor fetch, fexecute; with as. Выполнены разработка кода, тестирование, отладка. Готовые коды программ выложены на GitHub