Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №6  
По дисциплине: «Проектирование баз знаний»  
Тема: Создание приложения для работы с БД и организация пользовательского интерфейса: табличное представление и отчеты.

Выполнил:  
студент 3 курса  
группы ИИ-21(1)  
Романко Н.А.

Проверил:  
Савонюк В. А.

Брест 2023

Цель: получить навыки разработки приложений БД и организации пользовательского интерфейса: табличного представления информации и построения отчетов.

**Вариант 12**

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

Задание.

1. Изучить материал, приведенный в “К лабораторной работе 6.doc”.

2. На основании логической модели (в соответствии с вариантом Лабораторной работы №1) создать приложение и организовать пользовательский интерфейс, используя соответствующие элементы и методы: организовать табличное представление информации и построить отчеты не менее 4.

Ход выполнения лабораторной работы:

**Таблица table1:**

**Изображение выглядит как текст, снимок экрана, дисплей, линия

Автоматически созданное описание**

**Таблица table2:**

**Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание**

**Таблица table3: Изображение выглядит как снимок экрана, текст, линия, диаграмма

Автоматически созданное описание**

**Код программы:**

**Поиск данных в таблице:**

def select\_record():

    global local\_state, local\_state\_rows

    local\_state = "search"

    con = mysql.connect(host="localhost", user="root", password="1234567890", database="kindergarden")

    cursor = con.cursor()

    cursor.execute(f'SELECT \* FROM {chosen} WHERE {combo\_out2} = "{sel\_entry.get()}"')

    rows = cursor.fetchall()

    local\_state\_rows = rows

    frame = Frame(root)

    frame.place(x=10, y=300)

    tv = ttk.Treeview(frame, columns=(1, 2, 3), show="headings", height="5")

    tv.pack()

    match chosen:

        case "table1":

            tv.heading(1, text="№")

            tv.heading(2, text="adress")

            tv.heading(3, text="FIO")

            for i in rows:

                tv.insert('', 'end', values=i)

        case "table2":

            tv.heading(1, text="Category")

            tv.heading(2, text="Minimum аge")

            tv.heading(3, text="Maximum аge")

            for i in rows:

                tv.insert('', 'end', values=i)

        case "table3":

            tv.heading(1, text="№")

            tv.heading(2, text="Category")

            tv.heading(3, text="Teacher")

            for i in rows:

                tv.insert('', 'end', values=i)

    con.commit()

    con.close()

**Поиск данных в таблице:**

**Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание**

**Код сортировки таблицы:**

def sort\_record():

    global local\_state, local\_state\_rows, local\_state\_sort

    local\_state = "sort"

    if sort\_entry.get() == "убыв":

        sort = "desc"

        local\_state\_sort = "desc"

    else:

        sort = "asc"

        local\_state\_sort = "asc"

    con = mysql.connect(host="localhost", user="root", password="1234567890", database="kindergarden")

    cursor = con.cursor()

    cursor.execute(f'SELECT \* FROM {chosen} ORDER BY {combo\_out3} {sort}')

    rows = cursor.fetchall()

    local\_state\_rows = rows

    frame = Frame(root)

    frame.place(x=10, y=300)

    tv = ttk.Treeview(frame, columns=(1, 2, 3), show="headings", height="5")

    tv.pack()

    match chosen:

        case "table1":

            tv.heading(1, text="№")

            tv.heading(2, text="adress")

            tv.heading(3, text="FIO")

            for i in rows:

                tv.insert('', 'end', values=i)

        case "table2":

            tv.heading(1, text="Category")

            tv.heading(2, text="Minimum аge")

            tv.heading(3, text="Maximum аge")

            for i in rows:

                tv.insert('', 'end', values=i)

        case "table3":

            tv.heading(1, text="№")

            tv.heading(2, text="Category")

            tv.heading(3, text="Teacher")

            for i in rows:

                tv.insert('', 'end', values=i)

    con.commit()

    con.close()

**Сортировка данных в таблице:**

**Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание**

**Код создания отчета**

 def generate\_report():

    global local\_state, local\_state\_rows, local\_state\_sort

    report\_window = Toplevel(root)

    report\_window.geometry("400x300")

    report\_window.title("Отчет")

    report\_text = Text(report\_window)

    report\_text.pack(expand=True, fill=BOTH)

    def create\_report\_text():

        report\_text.delete(1.0, END)

        if local\_state == "search":

            report\_text.insert(END, f"Строка для поиска: {sel\_entry.get()}")

        elif local\_state == "sort":

            if local\_state\_sort == "desc":

                report\_text.insert(END, f"Сортировка по убыванию столбца {combo\_out3}")

            else:

                report\_text.insert(END, f"Сортировка по возрастанию столбца {combo\_out3}")

        elif local\_state == "no":

            report\_text.insert(END, f"Не выбрано")

        report\_text.insert(END, "\n\n")

        con = mysql.connect(host="localhost", user="root", password="1234567890", database="kindergarden")

        cursor = con.cursor()

        cursor.execute(f'SELECT \* FROM {chosen}')

        if local\_state\_rows == "no":

            rows = cursor.fetchall()

        else:

            rows = local\_state\_rows

        report\_text.insert(END, f"Таблица: {chosen}:\n")

        headers = [description[0] for description in cursor.description]

        table\_data = [headers] + list(rows)

        report\_text.insert(END, tabulate.tabulate(table\_data, headers="firstrow", tablefmt="grid"))

        con.close()

    create\_report\_text()

**Примеры отчетов:**

**Отчет без выполнения действий с таблицей:**

**Изображение выглядит как текст, электроника, снимок экрана, число

Автоматически созданное описание**

**Отчет после выполнения поиска:**

**Изображение выглядит как текст, снимок экрана, дисплей, число

Автоматически созданное описание**

**Отчет после сортировки таблицы:**

**Изображение выглядит как текст, снимок экрана, число, дисплей

Автоматически созданное описание**

**Отчет при пустом выводе:**

**Изображение выглядит как электроника, текст, снимок экрана, дисплей

Автоматически созданное описание**

Вывод: получил навыки разработки приложений БД и организации пользовательского интерфейса: табличного представления данных и составлении отчета.