ЗАДАНИЕ: ДА СЕ СЪЗДАДЕ ПРИЛОЖЕНИЕ, КОЕТО СЪЗДАВА БАЗА ДАННИ ЗА СТУДЕНТИ, СЪДЪРЖАЩА ПОНЕ НЯКОЛКО ПОЛЕТА (ПО ВАШ ИЗБОР). ДА СЕ ПРЕДВИДИ ВЪЗМОЖНОСТ ЗА ДОБАВЯНЕ НА ЗАПИС, РЕДАКТИРАНЕ НА ЗАПИС, ИЗТРИВАНЕ НА ЗАПИС И НЯКАКЪВ ВИД СОРТИРАНЕ (ПО ИЗБОР)

Отдолу предоставяме кратко описание и код как най-често може да се действа.

Създаването на приложение за управление на бази данни в **Borland C++ Builder 6** е значително лесно, особено ако се разчита на мощната архитектура **VCL (Visual Component Library)**.
За да създадем приложение ще използваме компонентите **BDE (Borland Database Engine)**, тъй като те са вградени и най-лесни за първоначална настройка в тази версия.

Необходими стъпки и код за изграждане на приложението:

---

1. Подготовка на компонентите (Визуално проектиране)

Отворете нов проект (**File -> New -> Application**) и поставете следните компоненти върху формата от палитрите:

Палитра	Компонент	Име (Name)	Описание
Data Access	TTable	Table1	Връзка с файла на базата данни.
Data Access	TDataSource	DataSource1	Мост между данните и визуалните контроли.
Data Controls	TDBGrid	DBGrid1	Таблица за визуализация на записите.
Standard	TButton	BtnAdd	Бутон за добавяне.
Standard	TButton	BtnEdit	Бутон за редактиране.
Standard	TButton	BtnDelete	Бутон за изтриване.
Standard	TButton	BtnSave	Бутон за запис (Post).

Свързване (Properties):

1. **Table1**: Настройте DatabaseName на BCDEMOS, а TableName на customer.db (примерен файл, идващ с инсталацията). Поставете Active на true.
2. **DataSource1**: Настройте свойството DataSet на Table1.
3. **DBGrid1**: Настройте свойството DataSource на DataSource1.

---

2. Програмен код (Unit1.cpp)

Сега ще добавим логиката за управление на записите чрез събитията OnClick на бутоните.

C++

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------

// ДОБАВЯНЕ НА ЗАПИС
void __fastcall TForm1::BtnAddClick(TObject *Sender)
{
        Table1->Append(); // Отваря празен ред в края на таблицата
        DBGrid1->SetFocus(); // Премества фокуса върху таблицата за писане
}

// РЕДАКТИРАНЕ НА ТЕКУЩ ЗАПИС
void __fastcall TForm1::BtnEditClick(TObject *Sender)
{
        if (!Table1->IsEmpty()) {
                Table1->Edit(); // Поставя текущия запис в режим на редакция
                DBGrid1->SetFocus();
        }
}

// ИЗТРИВАНЕ НА ЗАПИС
void __fastcall TForm1::BtnDeleteClick(TObject *Sender)
{
        if (!Table1->IsEmpty()) {
                // Потвърждение преди изтриване
                if (Application->MessageBox("Наистина ли искате да изтриете записа?",
                        "Потвърждение", MB_YESNO | MB_ICONQUESTION) == IDYES)
```

```
            {
                Table1->Delete();
            }
        }
}

// ЗАПИСВАНЕ НА ПРОМЕНИТЕ
void __fastcall TForm1::BtnSaveClick(TObject *Sender)
{
        if (Table1->State == dsEdit || Table1->State == dsInsert) {
            Table1->Post(); // Потвърждава промените в базата
            ShowMessage("Промените са записани успешно!");
        }
}
```

---

3. Как работи приложението

- **TTable** управлява физическия достъп до данните.
- **TDataSource** е "лепилото", което позволява на **TDBGrid** автоматично да се обновява, когато се движите по записите.
- **Методите на TTable**:
- Append(): Създава нов запис.
- Edit(): Позволява промяна на съществуващ.
- Post(): Физически записва информацията на диска.
- Cancel(): Отменя текущите промени (може да добавите и такъв бутон).

Важни бележки:

1. **BDE Конфигурация**: Ако използвате Windows 10 или 11, BDE (Borland Database Engine) може да изисква администраторски права за писане в определени папки.
2. **Собствена база данни**: Ако искате да създадете своя таблица, използвайте инструмента **Database Desktop** (идва с C++ Builder), запишете я като .db (Paradox) или .dbf (dBase) и я свържете чрез Table1->TableName.
3. **Автоматизация**: В C++ Builder 6 можете да използвате и компонента TDBNavigator, който обединява всички тези бутони в един контрол, но ръчното програмиране (както е показано горе) дава много повече контрол над потребителския интерфейс.