|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ | | | | | | | | |
|  | | |  | | |  | | |
| ФГБОУ ВО «Пермский государственный  национальный исследовательский университет» | | | | | | | | |
|  | | |  | | |  | | |
|  | | ОТЧЕТ  по лабораторной работе № 4 «Связанные элементы управления» по дисциплине «Учебная практика по БД и СУБД» | | | | |  | |
|  | | |  | | |  | | |
|  | Работу выполнил  студент гр. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Зимин И. В.\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2018 | | |  | Проверил  \_\_\_\_\_\_\_\_\_\_\_ кафедры МОВС  (доц., ст. преп., асс.)  Постаногов И. С.\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_ 2018 | | |  |
|  |  | | |  |  | | |  |
| Пермь 2018 | | | | | | | | |

1 Постановка задачи

Спроектировать и наполнить базу данных, состоящую из трёх таблиц, последовательно связанных связью типа 1:М.

Создать программу при помощи мастера, которая на графическом интерфейсе пользователя содержит три элемента управления типа выпадающий список, предоставляющих пользователю возможность выбора из элементов соответствующей таблицы, относящихся лишь к выбранному на более высоком уровне иерархии элементу.

Провести тестирование корректности работы программы.

Написать отчёт о проделанной работе.

2 Описание базы данных

Настало время воспользоваться ранее спроектированной базой данных «Автомобили», в которой содержится 3 таблицы, последовательно связанных связью типа «Один ко многим»

В данной предметной области выделены следующие сущности: «Страна-производитель», «Марка автомобиля» и «Модель автомобиля».

Страны-производители имеют название и год основания. Марки автомобилей имеют название и год основания. Модели автомобилей имеют название и год выпуска.

Схема БД «Автомобили» приведена на рисунке 2.1.

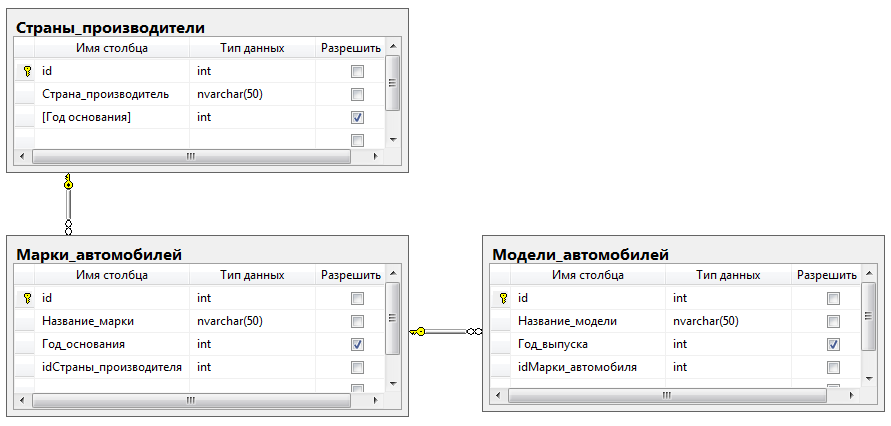


Рисунок 2.1 – Диаграмма базы данных «Автомобили»

1. Основное описание

Для решения данного задания необходимо было осуществить связь определённого языка программирования с данными из ранее спроектированной базы данных, при этом не написав ни строчки кода. Разработка приложения проводилась в среде Visual Studio 2015 на языке C#.

Приложение содержит 3 выпадающих списка:

В первом содержится полный список стран-производителей.

Во втором только те марки автомобилей, которые производятся в стране, которая на текущий момент выбрана в первом списке.

В третьем содержатся только те модели, которые выпускаются маркой, выбранной во втором списке, которая в свою очередь относится к выбранной в первом списке стране.

Интерфейс программы показан на рисунке 3.1.

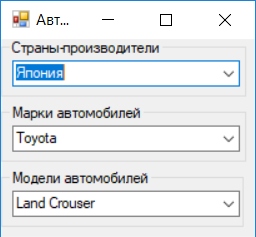


Рисунок 3.1 – Интерфейс приложения

4 Реализация на C#

Cпомощью следующих элементов управления, приложение может обращаться к базе данных, работать с ней:

DataSet – расположенное в оперативной памяти представление данных, обеспечивающее согласованную реляционную программную модель независимо от источника данных.

DataSet представляет полный набор данных, включая таблицы, содержащие, упорядочивающие и ограничивающие данные, а также связи между таблицами.

DataTable – класс, представляющий одну таблицу с данными в памяти.

BindingSource – класс, инкапсулирующий источник данных для каждого из трёх элементов управления типа выпадающий список (combobox).

Компонент BindingSource используется для двух целей. Он обеспечивает уровень косвенного обращения при выполнении привязки элементов управления в форме к данным. Это достигается путем привязки компонента BindingSource к источнику данных и последующей привязки элементов управления в форме к компоненту BindingSource. Все последующие взаимодействия с данными, включая перемещение, сортировку, фильтрацию и обновление, осуществляются с помощью вызовов компонента BindingSource.

TableAdapter – адаптер таблицы, который обеспечивает связь между приложением и базой данных путем выполнения инструкций SQL и хранимых процедур в базе данных. В дополнение к стандартным возможностям адаптера обработки данных, адаптеры таблиц предоставляют дополнительные запросы, совместно использующие общую схему со связанной типизированной таблицей данных. Адаптер таблицы загружает возвращенные данные в связанную таблицу данных в приложении или возвращает новые таблицы, уже заполненные данными.

Инициализация контейнера (контейнер – это класс, предназначенный для группировки и хранения компонентов):

this.components = new System.ComponentModel.Container();

Инициализация объекта, в котором будут храниться все данные из базы данных:

This.АвтомобилиDataSet = new lab4.АвтомобилиDataSet();

Инициализирует новый экземпляр класса BindingSource и добавляет источник данных в указанный контейнер (components):

this.страныПроизводителиBindingSource=new System.Windows.Forms.BindingSource(this.components);

Инициализация адаптера, который обеспечивает связь между таблицей из базы данных и приложением:

this.страны\_производителиTableAdapter=new lab5.АвтомобилиDataSetTableAdapters.Страны\_производителиTableAdapter();

Подает сигнал объекту о начале инициализации:

((System.ComponentModel.ISupportInitialize)(this.автомобилиDataSet)).BeginInit();

Задает источник данных для комбобокса:

this.cbCountry.DataSource = this.страныПроизводителиBindingSource;

Указывается поле записи, которое будет отображаться в данном комбобоксе (в данном случае названия стран-производителей):

this.cbCountry.DisplayMember = "Страна производитель";

Указывается поле записи, которое будет использоваться в качестве идентификатора строки выпадающего списка (в данном случае идентификатор страны-производителя):

this.cbCountry.ValueMember = "id";

Задает список в источнике данных, к которому в настоящий момент привязан соединитель:

* В случае если источник данных целиком совпадает с данными из определенной таблицы:

this.страныПроизводителиBindingSource.DataMember="Страны производители";

* В случае если источник данных представляет из себя соединение двух таблиц по внешнему ключу:

this.fKМаркиАвтомобилейСтраныПроизводителиBindingSource.DataMember = "FK\_Марки автомобилей\_Страны производители";

Задает источник данных, к которому привязан соединитель:

this.страныПроизводителиBindingSource.DataSource= this.автомобилиDataSet;

SchemaSerializationMode - свойство, которое задает тип сериализации для объекта DataSet (в данном случае «включая схему»).

Сериализация – процесс перевода какой-либо структуры данных в последовательность битов:

this.автомобилиDataSet.SchemaSerializationMode= System.Data.SchemaSerializationMode.IncludeSchema;

Подача сигнала о завершении инициализации

((System.ComponentModel.ISupportInitialize)(this.страныПроизводителиBindingSource)).EndInit();