**Лабораторная работа № 4**

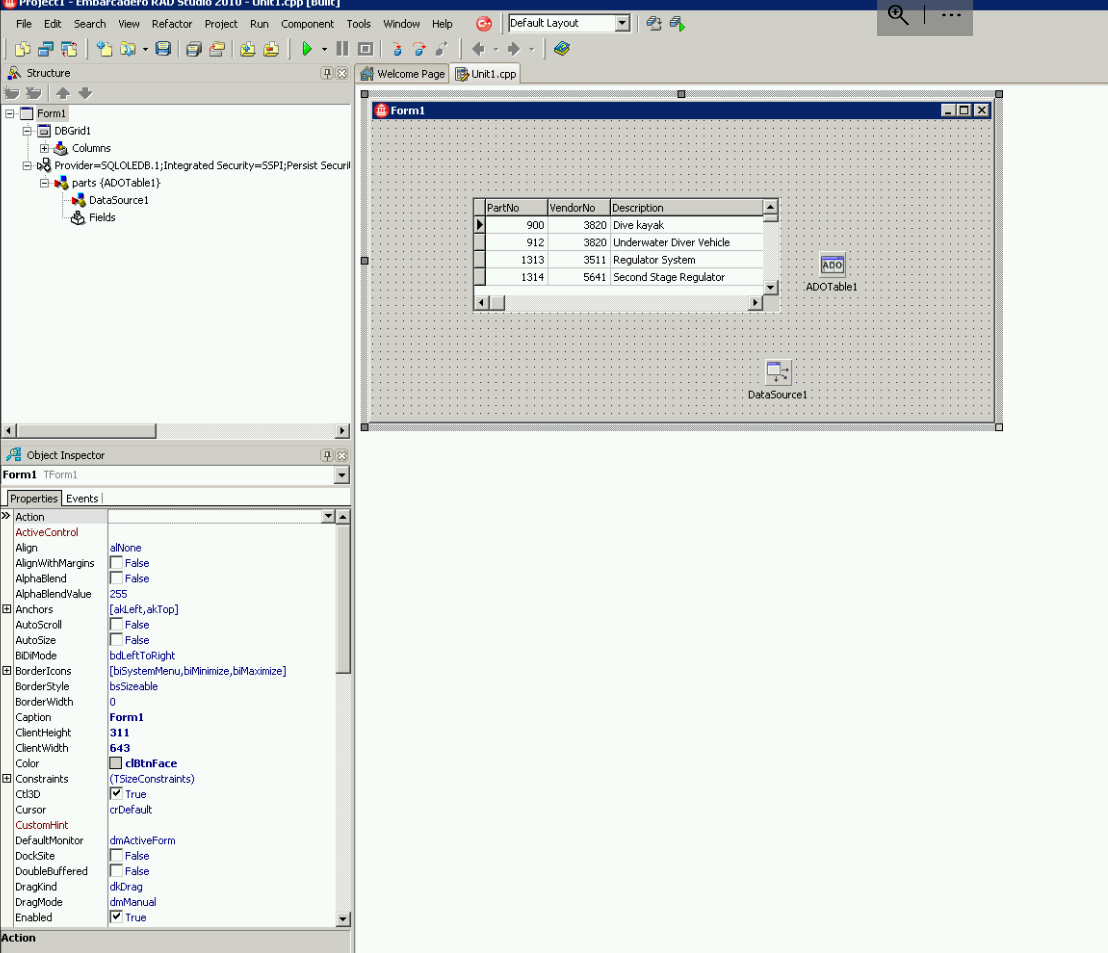
**Разработка приложений в системе C++Builder с использованием компонентов доступа к данным и отображения данных**

**Цель работы:** научиться разрабатывать приложения с использованием компонентов, обеспечивающих доступ к данным и отображение данных, хранящихся в базе данных, управляемой системой MS SQL Server.

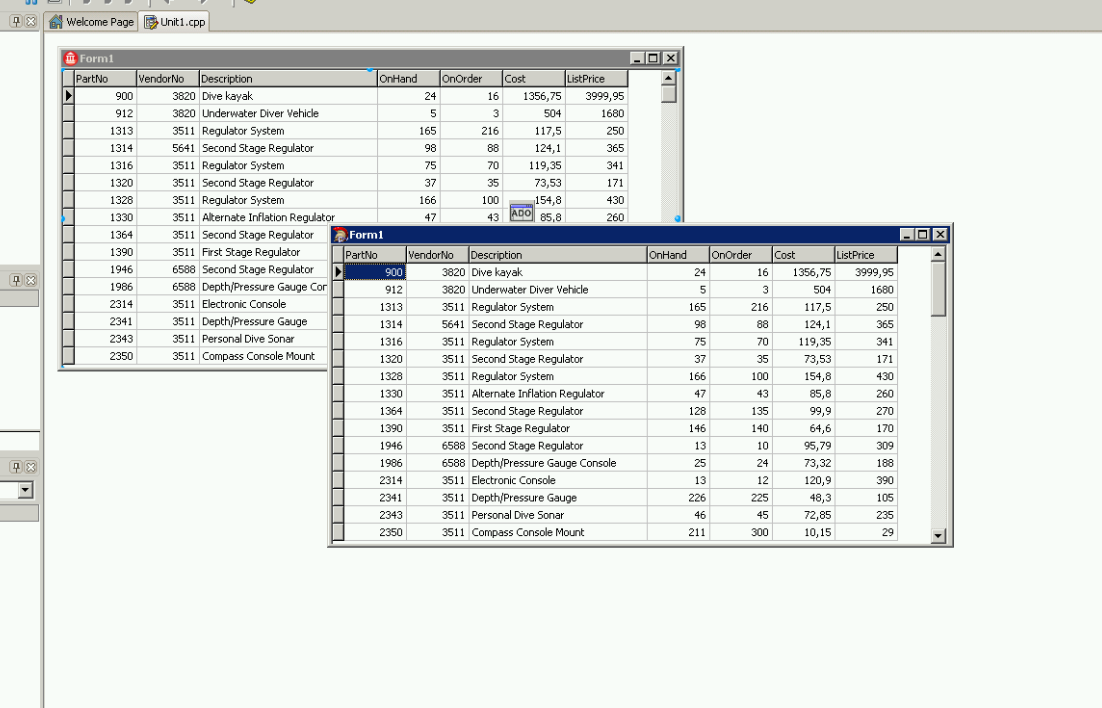
2. Создать приложение с формой типа сетка для просмотра таблицы PARTS базы данных DBdemosTest (на сервере Nebula), содержащей информацию о поставках изделий, разместив в форме этого приложения все компоненты, необходимые для просмотра таблицы PARTS (рис.9), и установив в окне инспектора объектов свойства размещенных компонентов, указанные в табл.1.

Use Connection String:

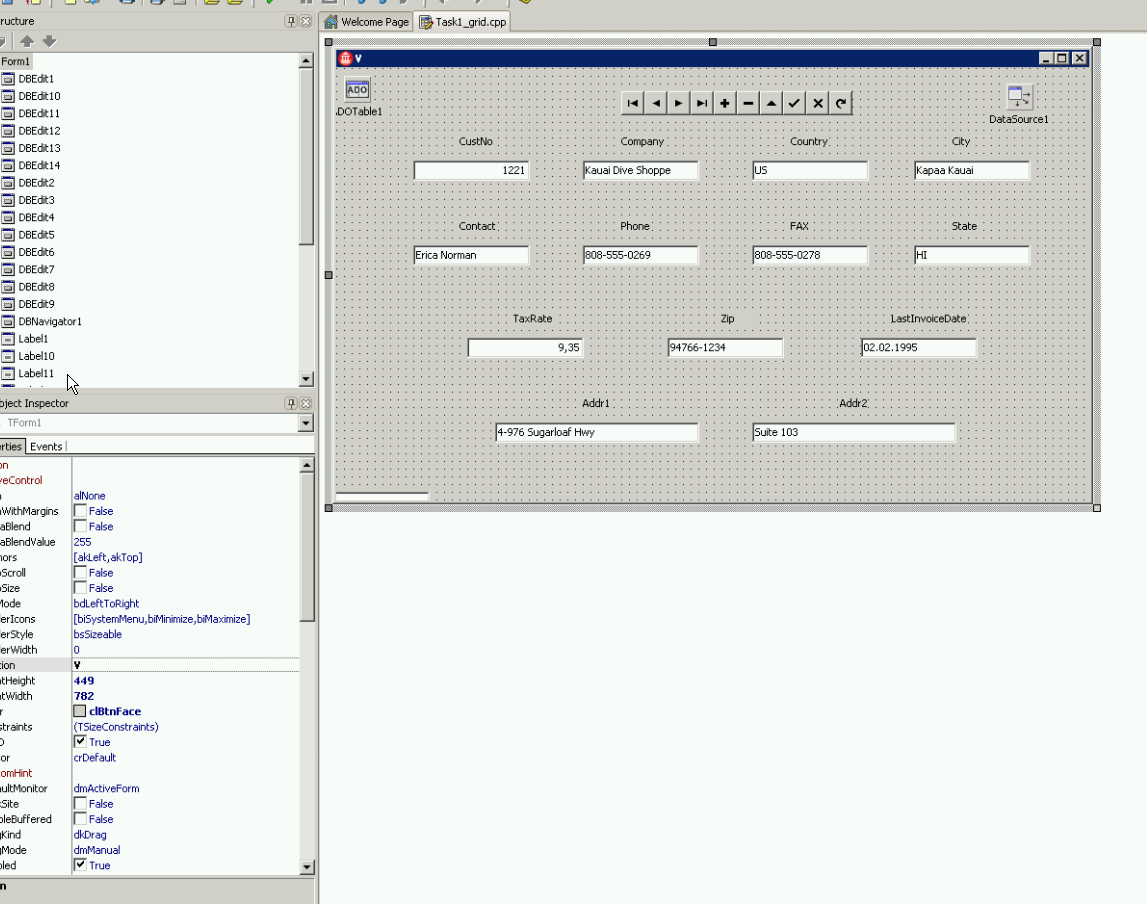
Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=DBdemosTest;Data Source=Nebula



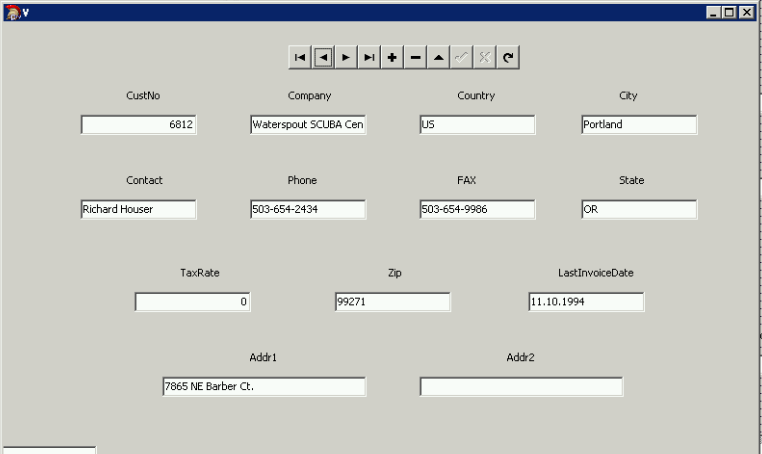
3. Проверить работу приложения и сохранить приложение командой File | Save Project As в папке Lab4-Task1 на устройстве С терминального компьютера в папке Мои документы.



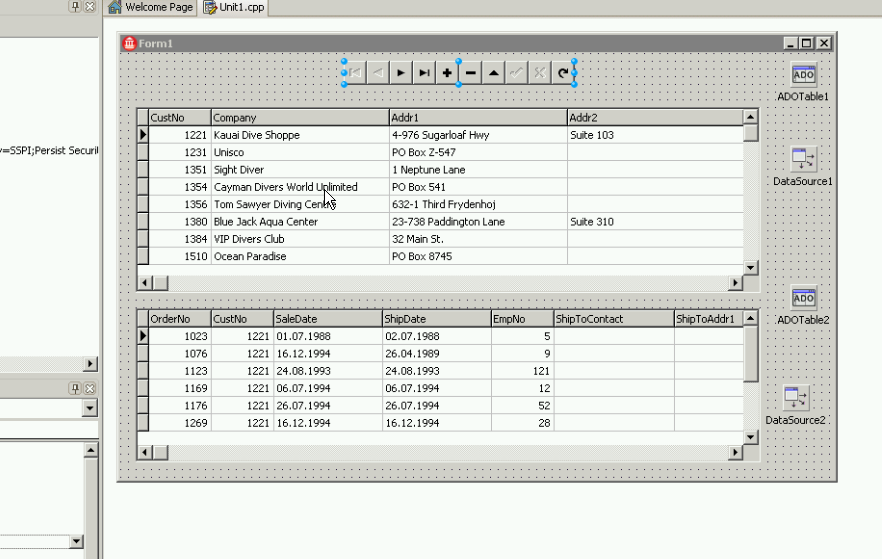
4. Создать приложение с формой типа ввод/редактирование для просмотра таблицы CUSTOMER, содержащей информацию о компаниях-покупателях (см. рис.6). В экранной форме для отображения значений полей текущей записи использовать компоненты типа TDBEdit, которые связаны с компонентом DataSource1 через свои свойства DataSource. Имя поля, значение которого отображается в компоненте типа TDBEdit, задать свойством DataField этого компонента. Надпись к полю задать свойством Caption компонента типа TLabel, принадлежащего группе Standard.



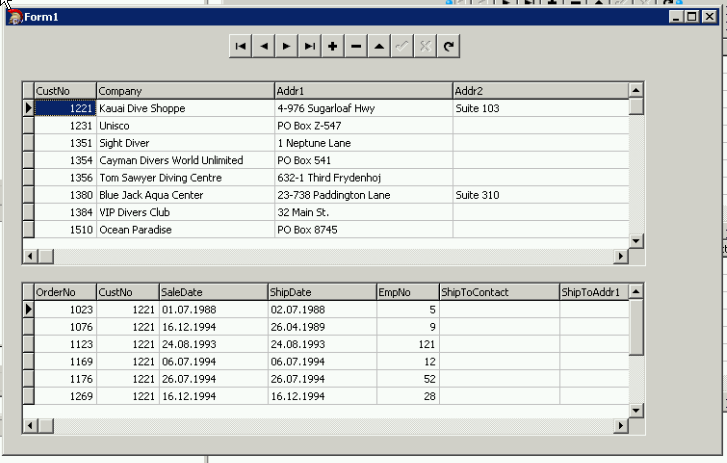
5. Проверить работу приложения и сохранить приложение в папке Lab4-Task2 на устройстве С терминального компьютера.



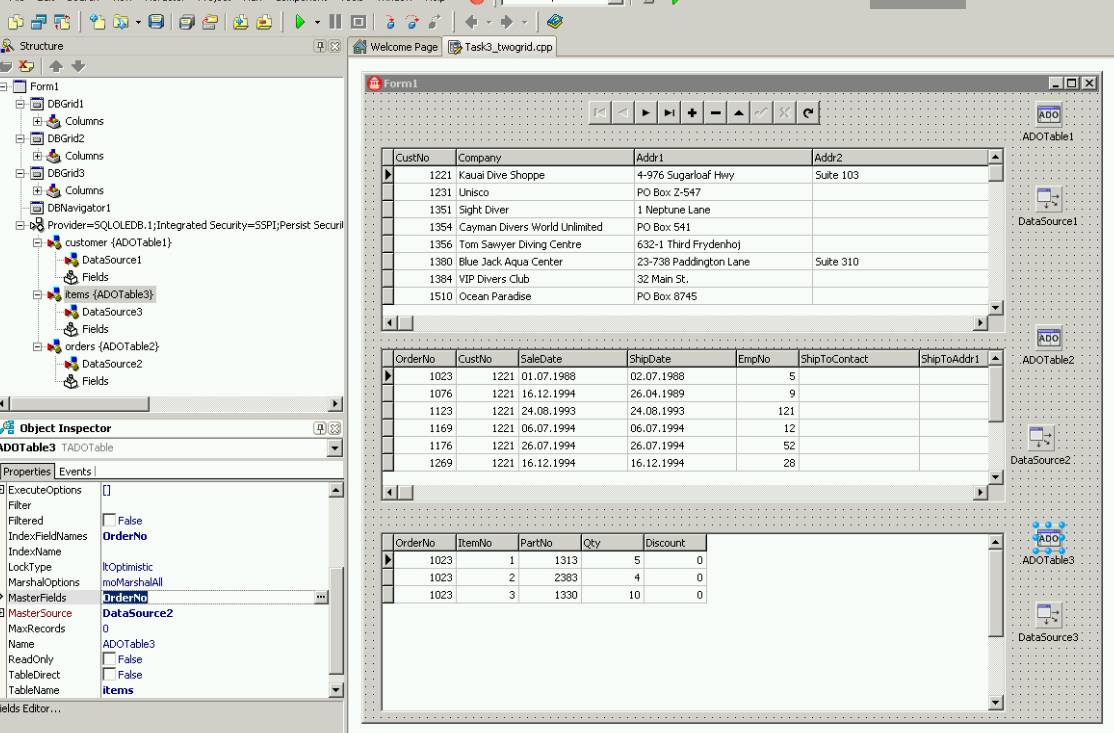
6. Создать приложение с формой типа главная/подчиненная для просмотра таблицы CUSTOMER и связанной с ней таблицы ORDERS, в которых содержится информация о компаниях-покупателях и сделанных заказах (см. рис.8). Информацию из каждой таблицы отобразить в компоненте типа TDBGrid.



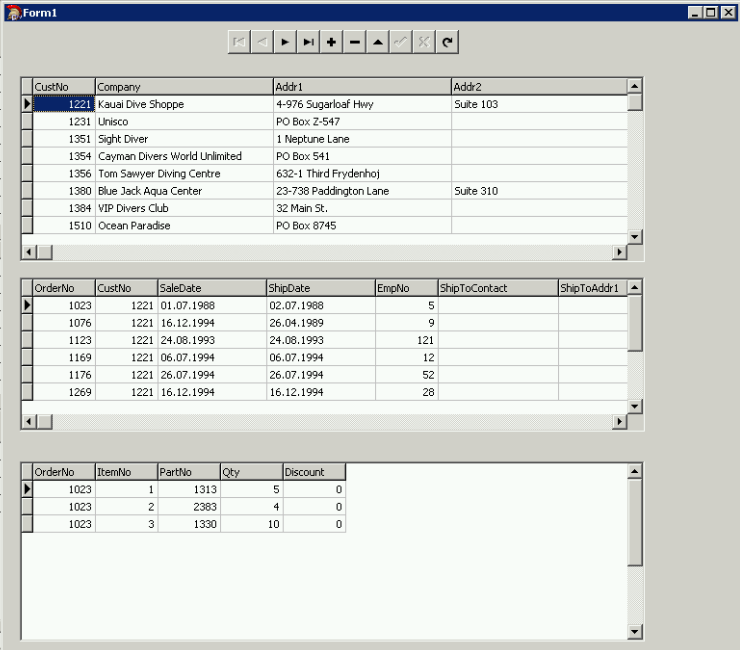
7. Воспользоваться табл.2, в которой приведены значения свойств используемых компонентов, устанавливаемые при разработке приложения.

8. Проверить работу приложения и сохранить приложение в папке Lab4-Task3 на устройстве С терминального компьютера.

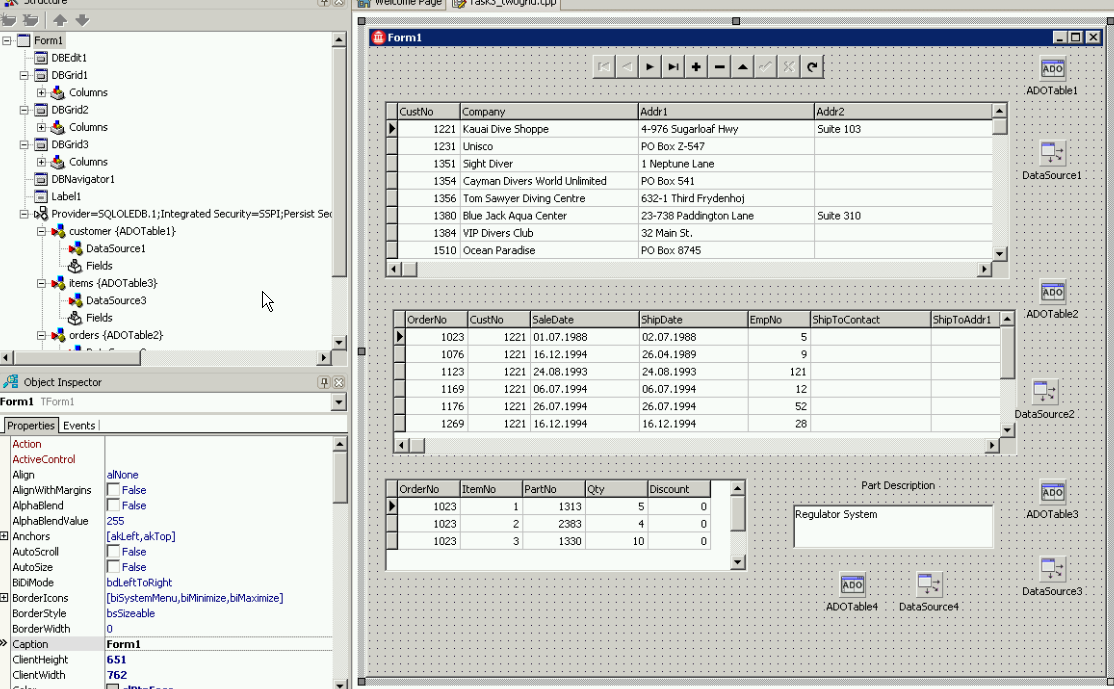
9. Приложение, созданное при выполнении п. 6, дополнить компонентами, обеспечивающими доступ, управление и отображение данных из таблицы ITEMS, и связать эту таблицу в качестве подчиненной с таблицей ORDERS.



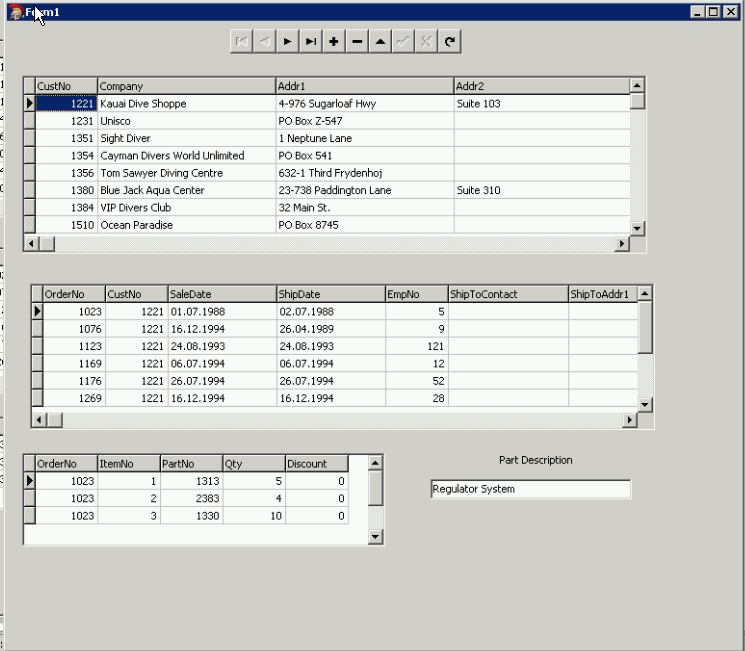
10. Проверить работу приложения, которое должно отображать данные из трех связанных таблиц. Сохранить приложение в папке Lab4-Task4 на устройстве С терминального компьютера.



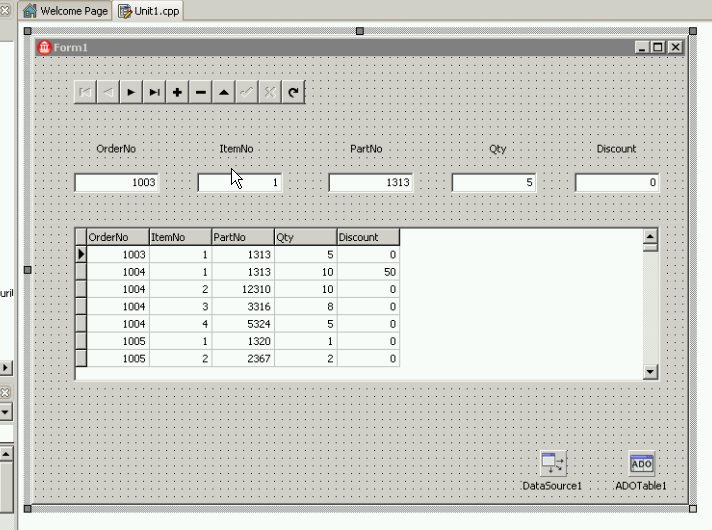
11. Приложение, созданное при выполнении п. 8, дополнить компонентами, обеспечивающими доступ, управление и отображение данных из таблицы PARTS, и связать эту таблицу в качестве подчиненной с таблицей ITEMS. Из таблицы PARTS требуется отображать только содержимое поля Description (описание поставляемого изделия) в компоненте типа TDBEdit.



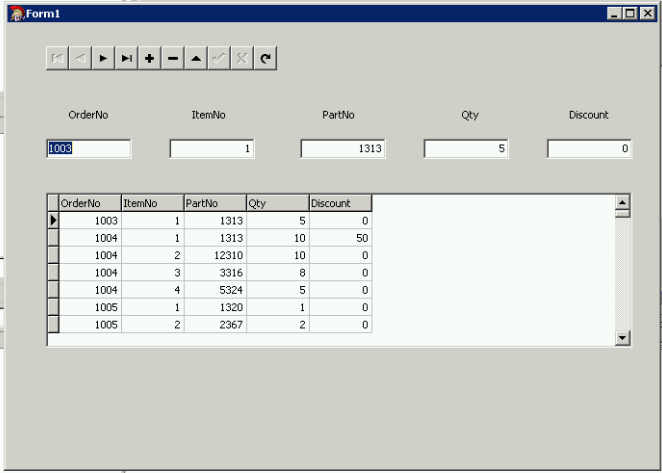
12. Проверить работу приложения, которое должно отображать данные из четырех связанных таблиц, и сохранить приложение в папке Lab4-Task5 на устройстве С терминального компьютера.



13. Создать приложение с формой типа ввод/редактирование для просмотра таблицы ITEMS и затем дополнить созданное приложение компонентом типа TDBGrid, чтобы продублировать отображение содержимого таблицы ITEMS (рис.10).

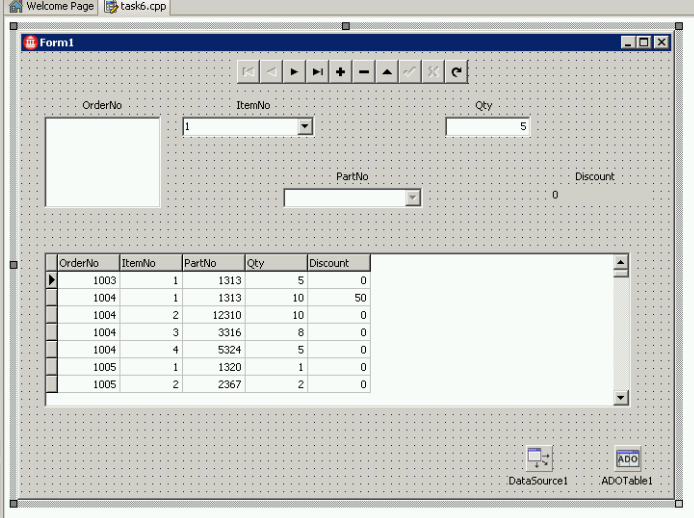


14. Проверить работу приложения и сохранить приложение в папке Lab4-Task6 на устройстве С терминального компьютера.



15. Модифицировать приложение, созданное при выполнении п. 12, чтобы обеспечить целостность БД с учетом того, что в таблице ITEMS значения поля OrderNo не должны отличаться от номеров заказов, зафиксированных в таблице ORDERS; значения поля PartNo не должны отличаться от номеров поставок изделий, зафиксированных в таблице PARTS; значения поля ItemNo должны находиться в определенном диапазоне (например, 1 - 5).

Для обеспечения ввода только допустимых значений в поля OrderNo и PartNo таблицы ITEMS, а также установки номера поставки путем выбора названия изделия и запрета доступа к отображаемому значению поля Discount, следует в форме, созданной при выполнении п. 12, заменить компоненты типа TDBEdit компонентами других типов, указанными в табл.3 (рис.11).



16. Настроить замещающие компоненты, чтобы:

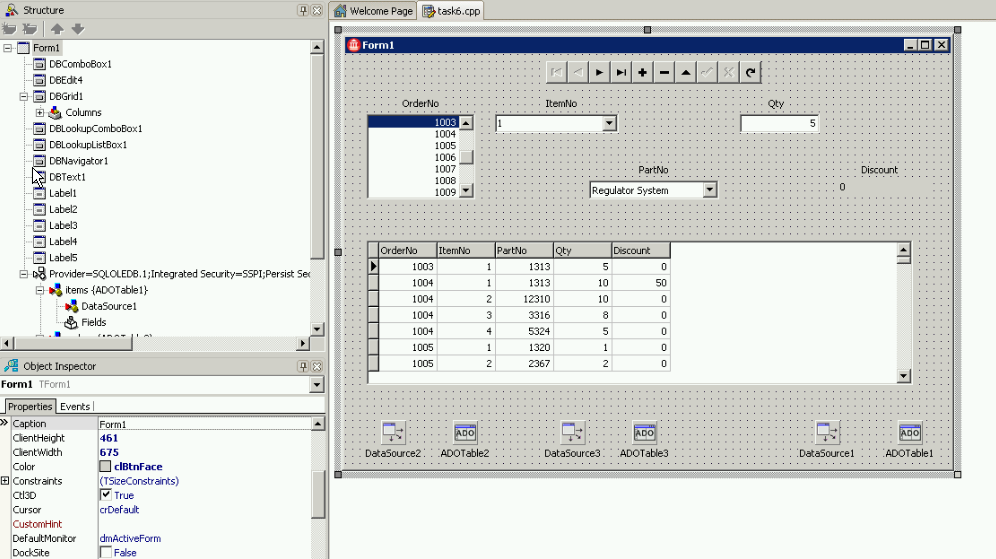
• поле OrderNo (номер заказа) содержало значение, имеющееся в таблице ORDERS;

• поле ItemNo (порядковый номер изделия в заказе) содержало значение, выбираемое из диапазона 1 - 5;

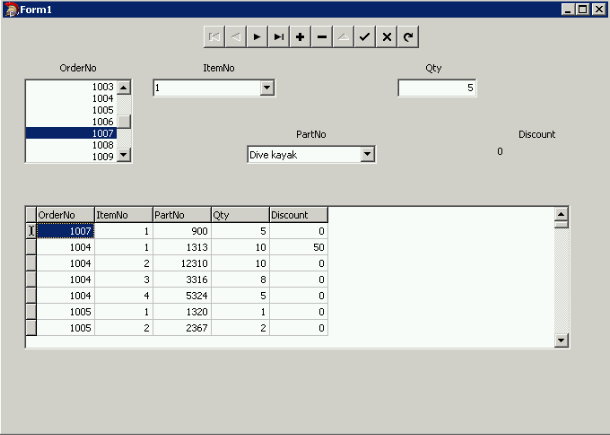
• поле PartNo по номеру поставки находило в таблице PARTS и отображало в рабочем поле формы название соответствующего поставляемого изделия (товара);

• поле Discount (скидка) было доступно только для просмотра.

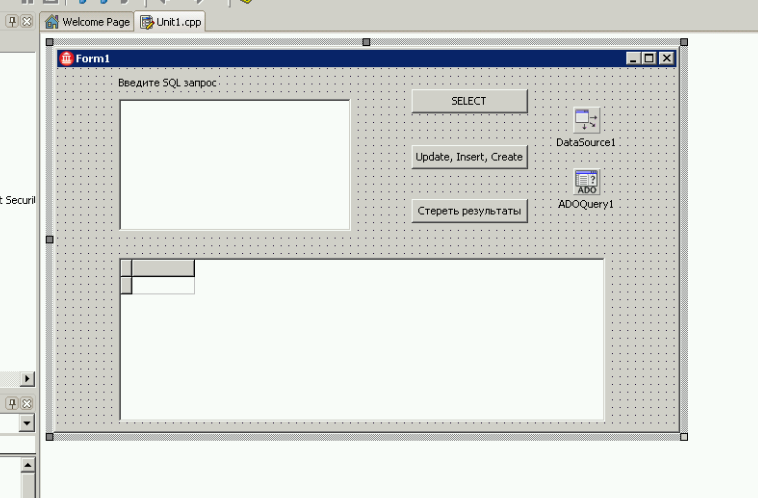
Для такой настройки таблицу (ORDERS или PARTS), связанную с таблицей ITEMS, представить в рабочем поле формы компонентами типа TADOTable и TDataSource, взаимосвязь таблиц задать установкой перечисленных в табл.4 свойств компонента типа TDBLookupListBox или TDBLookupComboBox, а свойства компонента типа TDBComboBox установить в соответствии с табл.5.



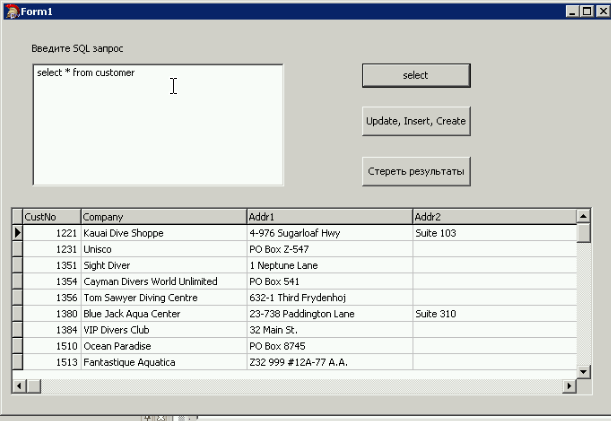
17. Проверить работу приложения, осуществив корректировку имеющихся и вставку новых строк в таблицу ITEMS , выбирая значения полей OrderNo и PartNo с помощью компонентов типа TDBLookupListBox и TDBLookupComboBox; изменение данных наблюдать в дополнительно размещенном компоненте типа TDBGrid

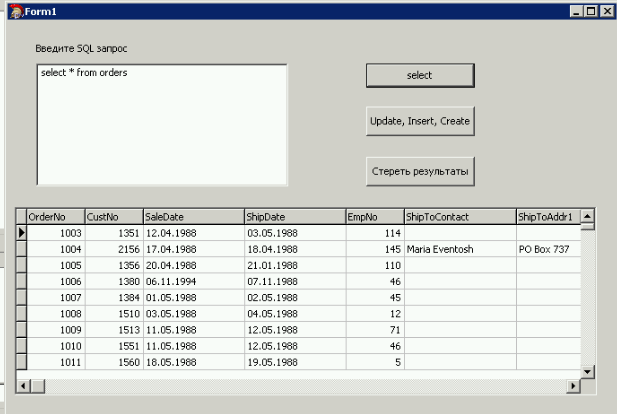


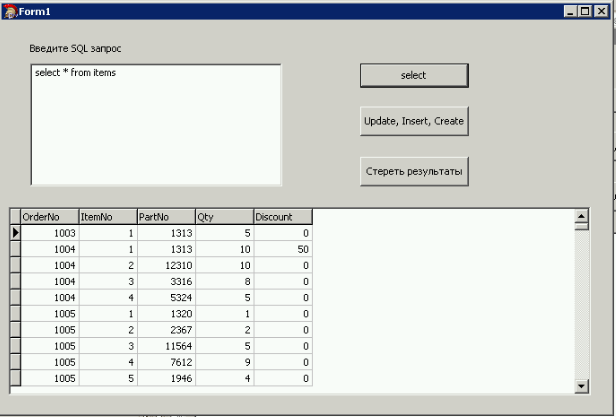
18. Разработать приложение для выполнения операторов языка SQL.

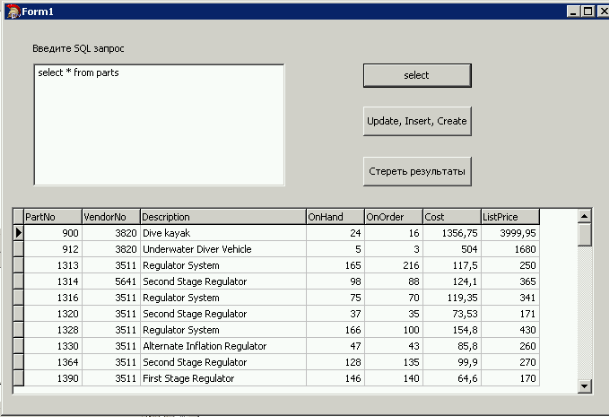


19. Запустить созданное приложение на выполнение и с помощью операторов SELECT вывести содержимое таблиц CUSTOMER, ORDERS, ITEMS, PARTS.

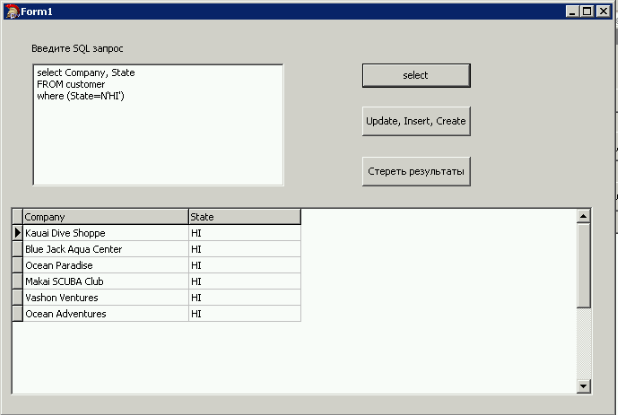


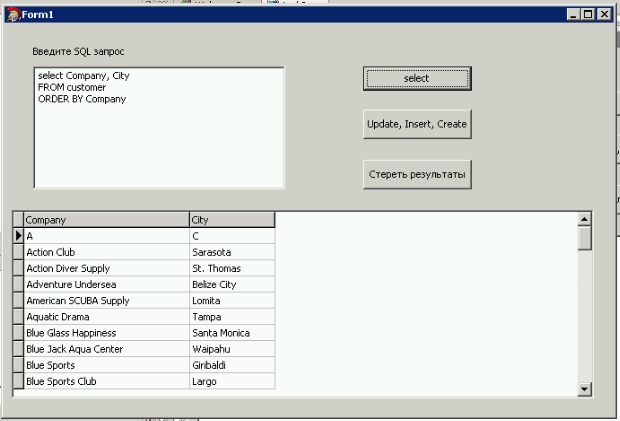


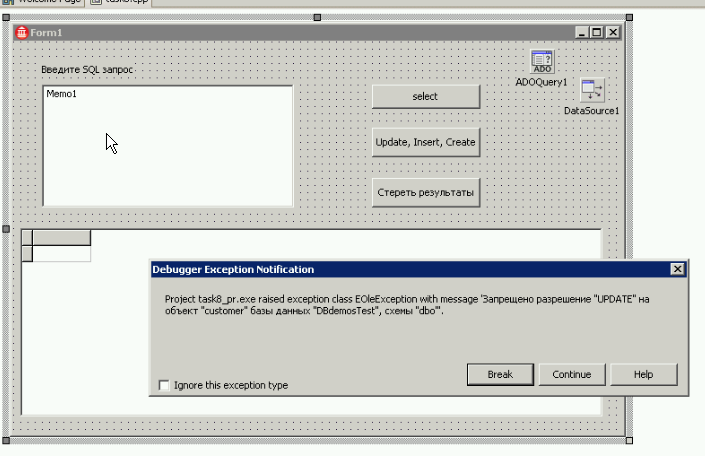




20. Для указанных таблиц выполнить записанные в отчет по лабораторной работе № 1 операторы языка SQL с выборкой, сортировкой, группировкой, изменением и добавлением данных.

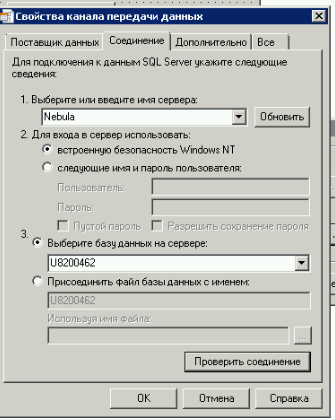




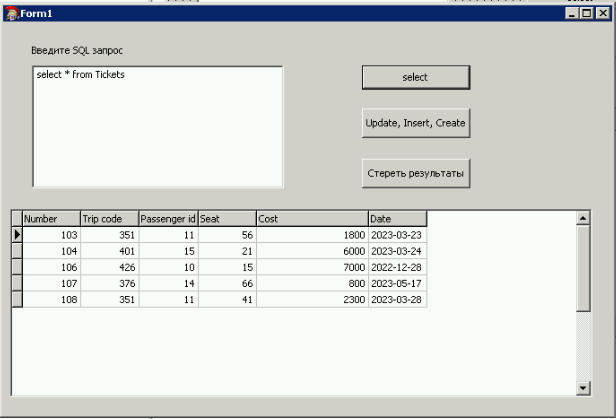


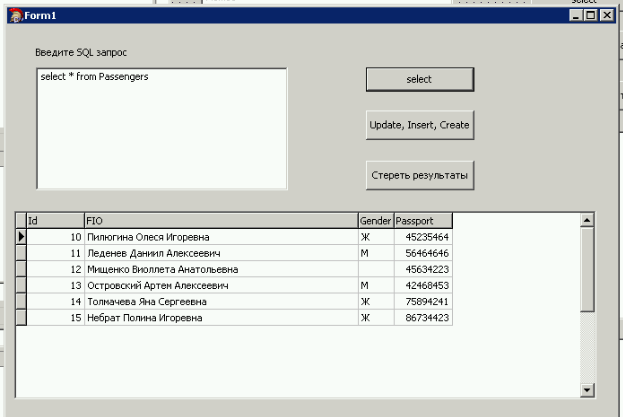
21. Сохранить приложение в папке Lab4-Task7 на устройстве С терминального компьютера.

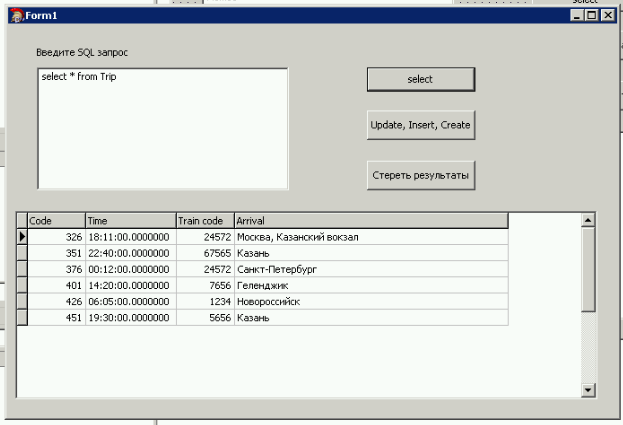
22. Скорректировать приложение, чтобы иметь доступ к базе данных, спроектированной и созданной на сервере Nebula в лабораторной работе № 2.



23. Проверить работу скорректированного приложения, выполнив включенные в отчет при домашней подготовке SQL-запросы для спроектированной базы данных.







24. Сохранить приложение в папке Lab4-Task8 на устройстве С терминального компьютера.

**Контрольные вопросы**

1. Какие компоненты системы C++ Builder предназначены для связи с БД?

*Ответ: Компонент типа TTable или TQuery для связи с BDE и через него с БД, компонент типа TADOTable или TADOQuery для подключения к БД с использованием технологии ADO.*

2. Какие компоненты системы C++ Builder предназначены для отображения данных и управления ими?

*Ответ: Компонент типа TDBGrid отображает содержимое таблицы в виде строк и столбцов, компонент типа TDBEdit - содержимое одного поля текущей строки таблицы, а компонент типа TDBComboBox отображает значение поля и задает список возможных значений этого поля.*

3. Укажите достоинства и недостатки технологии ADO.

*Ответ: Основным достоинством технологии ADO является ее естественная ориентация на создание “облегченного” клиента, для которого, в отличие от технологии BDE, не требуется устанавливать специальные программные средства на клиентской машине, поскольку необходимая поддержка обеспечивается операционной системой Windows.Недостатком технологии ADO является то, что ею нельзя воспользоваться, если для соответствующей структуры данных (в частности, для БД многих популярных серверов - InterBase, Informix, DB2 и пр.) не создан нужный провайдер или ODBC-драйвер.*

4. Приведите примеры форм типа ввод/редактирование, сетка, главная таблица/подчиненная таблица.

*Ответ: Рисунок 1, Рисунок 2 и Рисунок 3*

5. Какое свойство компонента типа TADOTable управляет открытием таблицы базы данных?

*Ответ: Свойство Active*

6. Какие значения может принимать свойство компонента типа TADOTable, которое управляет открытием таблицы базы данных, и как задавать эти значения вручную и программно?

*Ответ: Свойство Active компонента типа TADOTable, управляющее открытием таблицы, может устанавливаться вручную в окне инспектора объектов или программно методами Open (соответствует Active=True) и Close (соответствует Active=False).*

7. Каким способом можно изменить состав кнопок навигатора?

*Ответ: Наличие конкретной кнопки задается в свойстве VisibleButtons установкой соответствующего идентификатора nbXxx в значение True (есть кнопка) или False (нет кнопки).*

8. Какие свойства компонента типа TADOTable задают связь этого компонента с конкретной таблицей базы данных?

*Ответ: DataSource*

9. Какие свойства компонентов типа TDataSource и TDBGrid (TDBEdit, TDBNavigator) и с какими значениями нужно задать, чтобы образовать связь компонентов, показанную на рис.2,*а*?

*Ответ: Таким образом, компонент TADOConnection играет роль концентратора соединений с источником данных компонентов-наборов. Компонент TADOTable является аналогом BDE-компонента TTable и представляет в клиентской программе набор данных, состоящий из строк таблицы БД, имя которой содержит его свойство TableName.*

*Компонент TADOQuery аналогичен BDE-компоненту TQuery и предназначен для получения строк из одной или нескольких таблиц БД и внесения изменений в одну из таблиц БД. Запрос на выполнение необходимой операции задается в свойстве SQL. Значение свойства SQL устанавливается либо в окне инспектора объектов, либо в программе методами Clear( ) и Add( ). Для управления компонентом ADOQuery используются специальные методы Close( ), Open( ), ExecSQL( ).*

10. Какие свойства компонента типа TADOTable для подчиненной таблицы необходимо установить, чтобы задать связь с главной таблицей?

*Ответ: Для связи подчиненной таблицы с главной предназначены свойства MasterSource и MasterFields компонента типа TADOTable, представляющего подчиненную таблицу. Задать значение свойства MasterFields можно с помошью конструктора связанных полей (Field Link Designer), который вызывается из инспектора объектов нажатием кнопки, расположенной в строке свойства MasterFields. В окне конструктора следует выбрать нужный индекс (индексный ключ) и установить связь между полями подчиненной (detail) и главной (master) таблиц.*

11. Укажите назначение свойств компонентов TDBLookupListBox, TDBLookupComboBox, TDBComboBox.

*Ответ: Компонент типа TDBLookupListBox или TDBLookupComboBox отыскивает в таблице, связанной с ListSource, строку, в которой значение поля с именем, указанным в KeyField, совпадает со значением поля с именем, указанным в DataField, и отображает из найденной строки значение поля, имя которого указано в ListField. Такие функциональные возможности позволяют при добавлении строки в таблицу ITEMS выбирать номер заказа из поля OrderNo таблицы ORDERS и записывать его в одноименное поле таблицы ITEMS.*

12.Укажите назначение свойств и методов компонента TADOQuery.

*Ответ: Компонент TADOQuery аналогичен BDE-компоненту TQuery и предназначен для получения строк из одной или нескольких таблиц БД и внесения изменений в одну из таблиц БД. Запрос на выполнение необходимой операции задается в свойстве SQL. Значение свойства SQL устанавливается либо в окне инспектора объектов, либо в программе методами Clear( ) и Add( ). Для управления компонентом ADOQuery используются специальные методы Close( ), Open( ), ExecSQL( )*