**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.

***Продолжительность работы*** - 4 ч.

**Ответы на вопросы:**

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

-Компонент типа TTableили TQuery для связи с BDEи через него с БД либо компонент типа TADOTableили TADOQuery для подключения к БД с использованием технологии ADO (ActiveX Data Objects - объекты данных, построенные как объекты ActiveX);

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

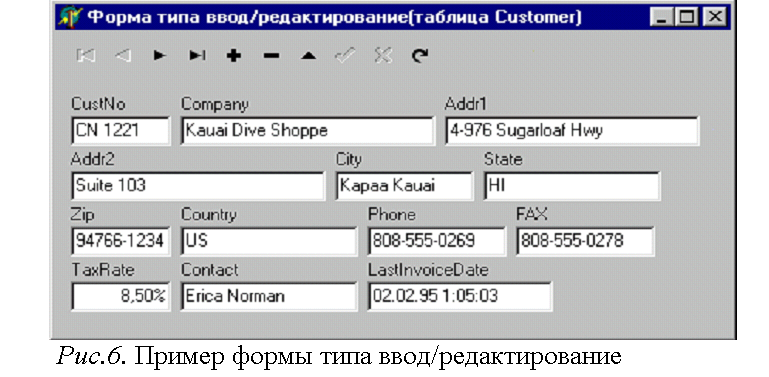
-Компонент типа TDataSource для соединения используемых приложением наборов данных с визуальными компонентами пользовательского интерфейса; визуальные компоненты для создания такого интерфейса.

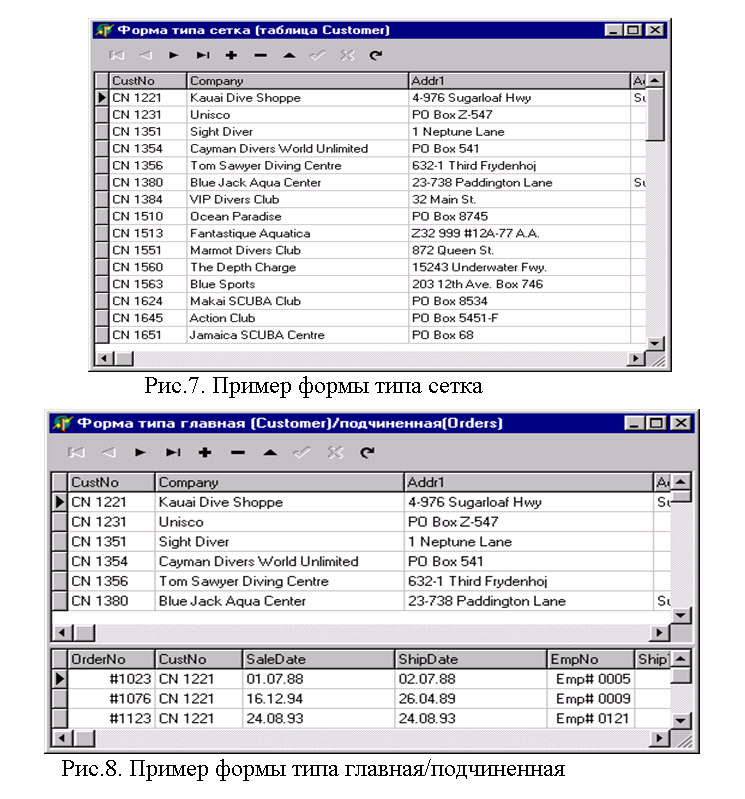
1. **Укажите достоинства и недостатки технологии ADO.**

-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.

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

**-**

****

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

**-** Active

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

**-** True\False

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).

1. **Каким способом можно изменить состав кнопок навигатора?**

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

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

*- ConnectionString*, *TableName, Dataset*

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

**-** Не знаю

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

**-** Для связи подчиненной таблицы с главной предназначены свойства *MasterSource* и *MasterFields* компонента типа TADOTable, представляющего подчиненную таблицу. Задать значение свойства *MasterFields* можно с помощью 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.

1. **Укажите назначение свойств компонентов TDBLookupListBox, TDBLookupComboBox, TDBComboBox.**

**Свойства замещающих компонентов**

| Свойство | Значение | Примечание |
| --- | --- | --- |
| DBLookupListBox1: TDBLookupListBox | | |
| DataSource | DataSource1 | Имя компонента, обеспечивающего доступ к таблице ITEMS |
| DataField | OrderNo | Имя поля в таблице ITEMS, значение которого отыскивается в таблице ORDERS |
| ListSource | DataSource2 | Имя компонента, через который осуществляется обмен и управление данными из таблицы ORDERS, связанной с таблицей ITEMS |
| ListField | OrderNo | Имя поля в таблице ORDERS, значения которого отображаются в компоненте DBLookupListBox1 |
| KeyField | OrderNo | Имя поля в таблице ORDERS, значение которого отыскивается по значению поля, указанного в свойстве DataField |
| DBLookupComboBox1: TDBLookupComboBox | | |
| DataSource | DataSource1 | Имя компонента, обеспечивающего доступ к таблице ITEMS |
| DataField | PartNo | Имя поля в таблице ITEMS, значение которого отыскивается в таблице PARTS |
| ListSource | DataSource3 | Имя компонента, через который осуществляется обмен и управление данными из таблицы PARTS , связанной с таблицей ITEMS |
| ListField | Description | Имя поля в таблице PARTS, значение которого отображается в компоненте DBLookupComboBox1 |
| KeyField | PartNo | Имя поля в таблице PARTS, значение которого отыскивается по значению поля, указанного в свойстве DataField |

**Свойства компонента DBComboBox1**

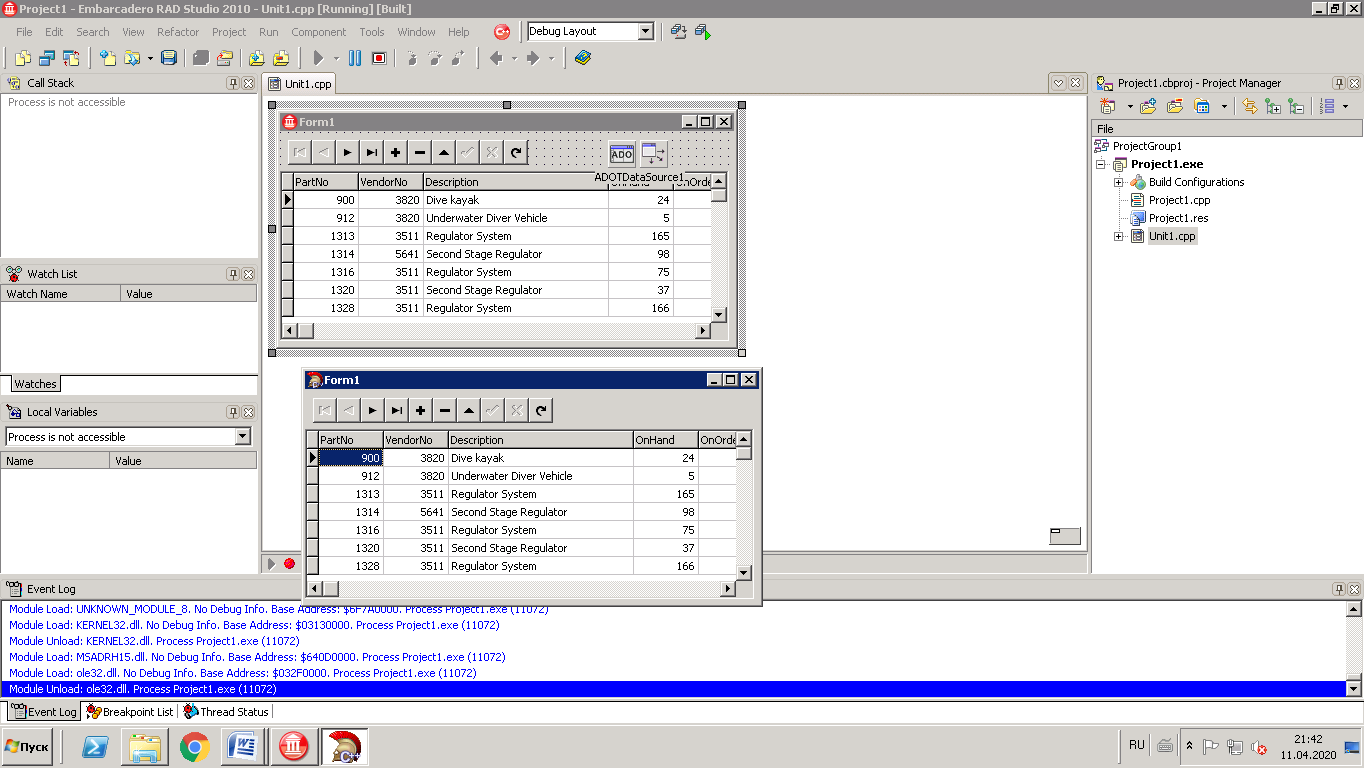
|  |  |  |
| --- | --- | --- |
| Свойство | Значение | Примечание |
| DBComboBox1: TDBComboBox | | |
| DataField | ItemNo | Имя поля, значение которого отображается и устанавливается |
| DataSource | DataSource1 | Имя компонента, обеспечивающего доступ к таблице ITEMS |
| Items | 1  2  3  4  5 | Список возможных значений, присваиваемых полю текущей записи; значения задаются в текстовом редакторе, который вызывается из инспектора объектов нажатием кнопки, расположенной в строке свойства Items |

1. **Укажите назначение свойств и методов компонента TADOQuery.**

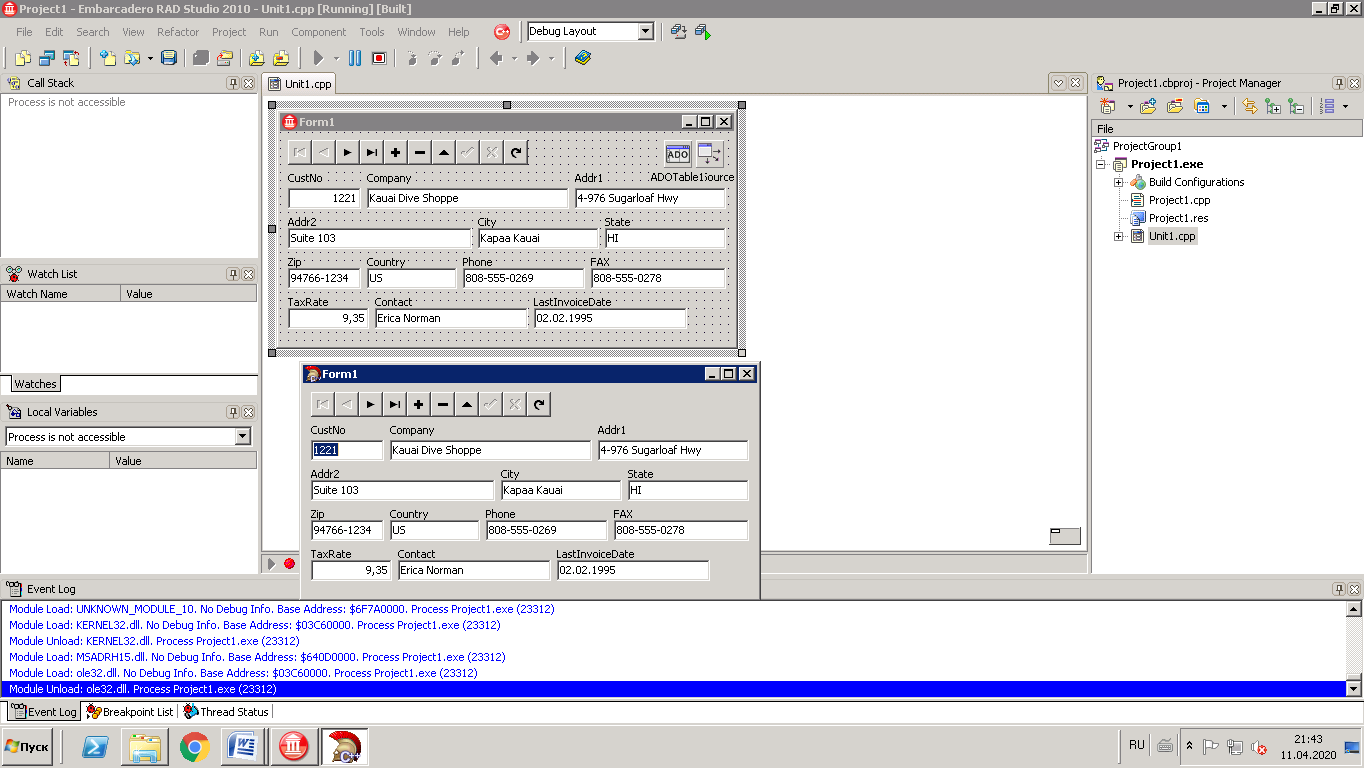
**-**Компонент TADOQuery аналогичен BDE-компоненту TQuery и предназначен для получения строк из одной или нескольких таблиц БД и внесения изменений в одну из таблиц БД. Запрос на выполнение необходимой операции задается в свойстве SQL. Значение свойства SQL устанавливается либо в окне инспектора объектов,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

**Выполнение работы**

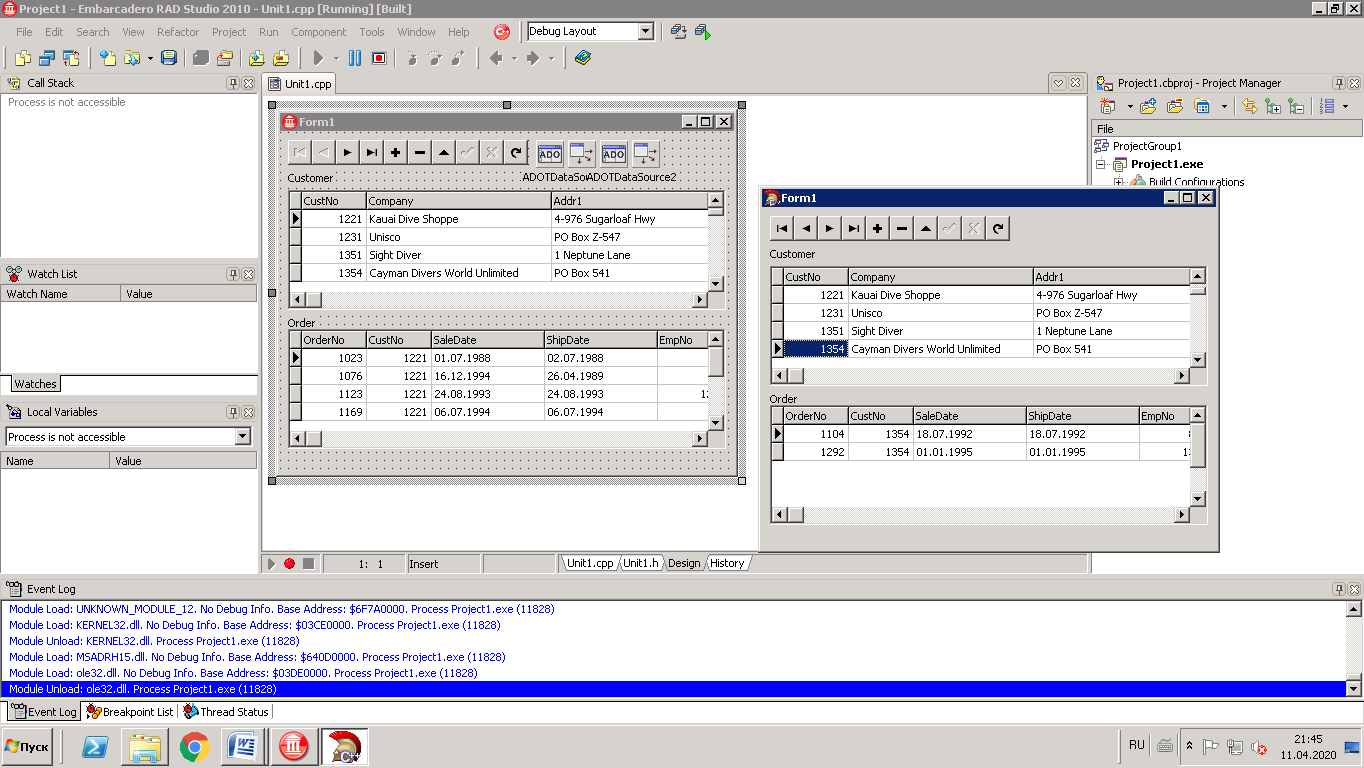
2. Создать приложение с формой типа сетка для просмотра таблицы PARTS базы данных DBdemosTest (на сервере Nebula)*,* содержащей информацию о поставках изделий, разместив в форме этого приложения все компоненты, необходимые для просмотра таблицы PARTS, и установив в окне инспектора объектов свойства размещенных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



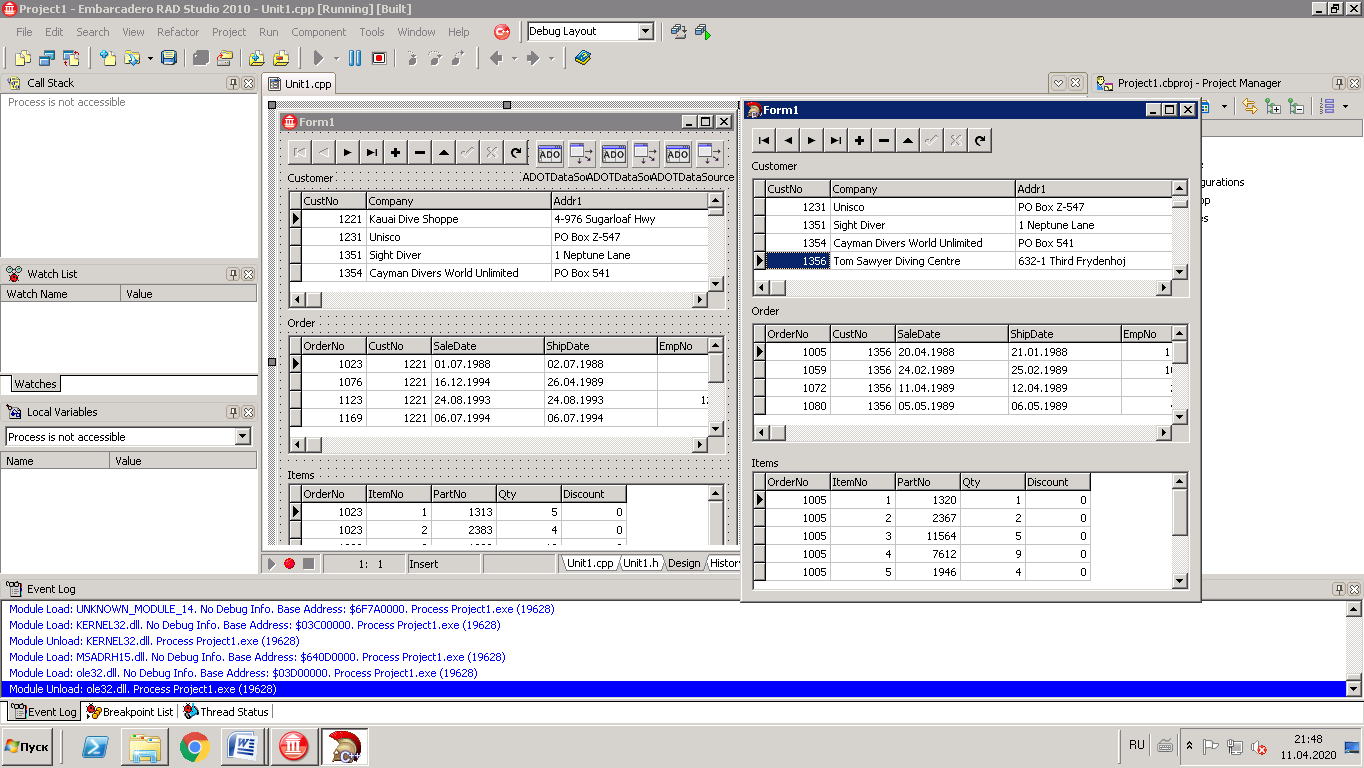
4. Создать приложение с формой типа ввод/редактирование для просмотра таблицы CUSTOMER, содержащей информацию о компаниях-покупателях (см. рис.6). В экранной форме для отображения значений полей текущей записи использовать компоненты типа TDBEdit,которые связаны с компонентом DataSource1 через свои свойства *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.



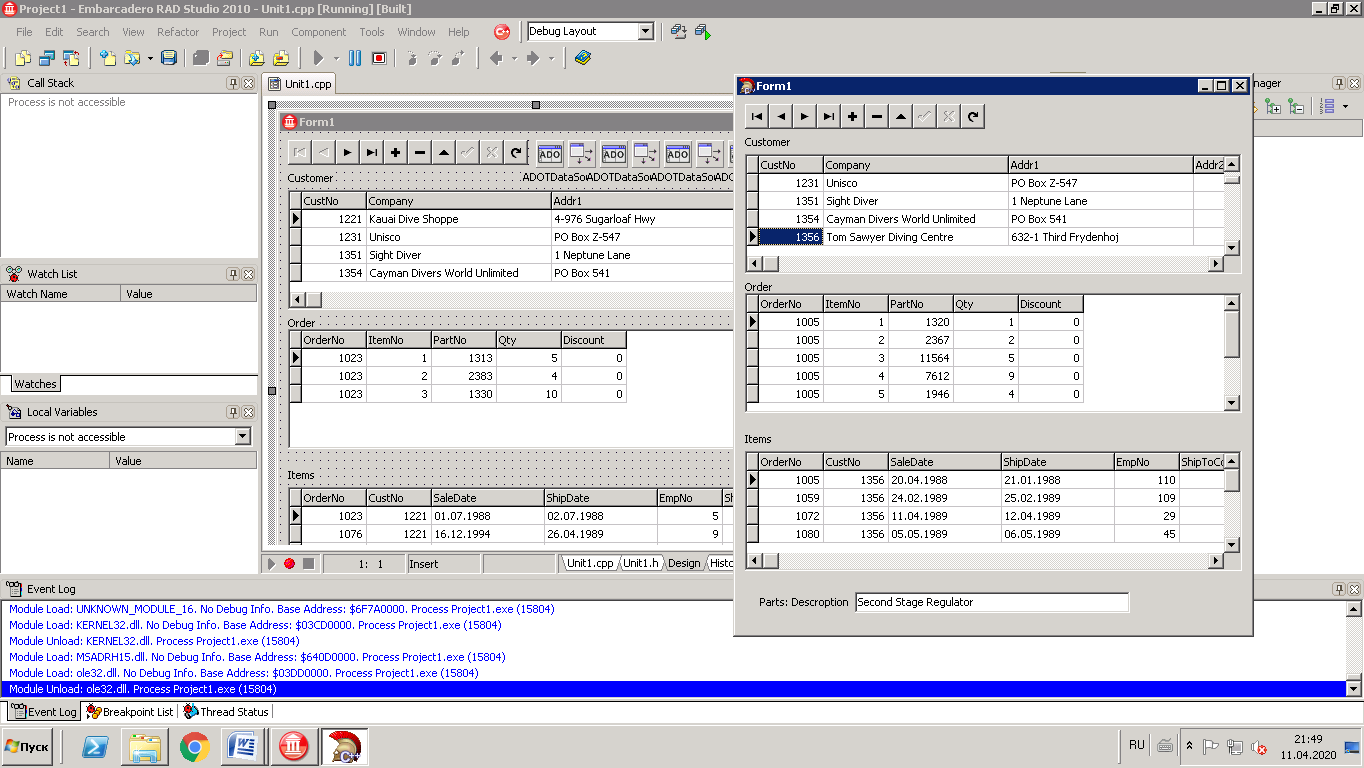
6. Создать приложение с формой типа главная/подчиненная для просмотра таблицы CUSTOMER и связанной с ней таблицы ORDERS,в которых содержится информация о компаниях-покупателях и сделанных заказах (см. рис.8). Информацию из каждой таблицы отобразить в компоненте типа TDBGrid.

****

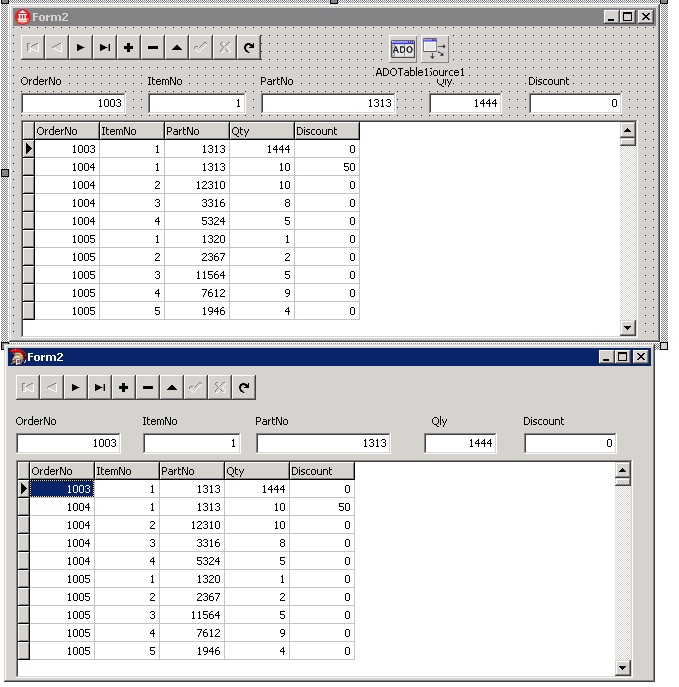
9. Приложение, созданное при выполнении п. 6, дополнить компонентами, обеспечивающими доступ, управление и отображение данных из таблицы ITEMS,и связать эту таблицу в качестве подчиненной с таблицей ORDERS.

****

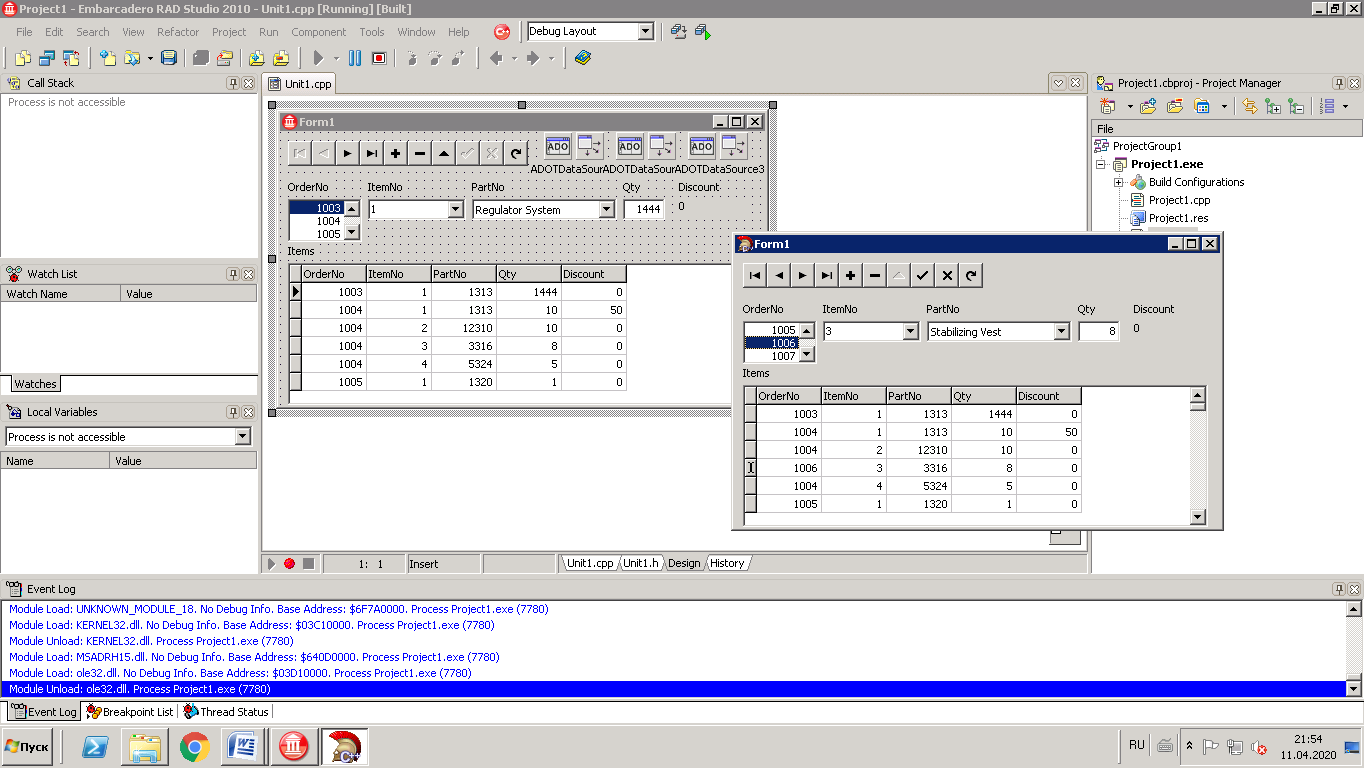
11. Приложение, созданное при выполнении п. 8, дополнить компонентами, обеспечивающими доступ, управление и отображение данных из таблицы PARTS,и связать эту таблицу в качестве подчиненной с таблицей ITEMS.Из таблицы PARTS требуется отображать только содержимое поля *Description* (описание поставляемого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



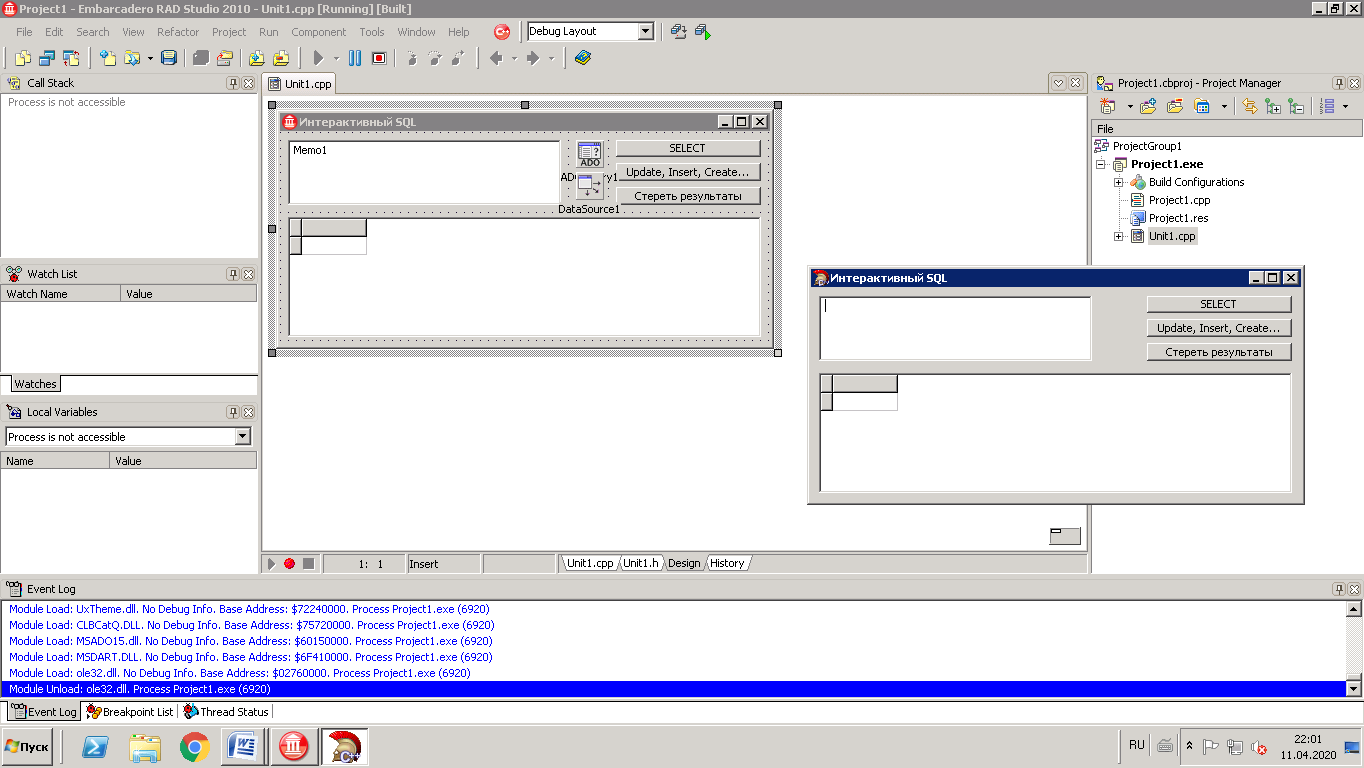
13. Создать приложение с формой типа ввод/редактирование для просмотра таблицыITEMS и затемдополнить созданное приложение компонентом типа TDBGrid,чтобы продублировать отображение содержимого таблицыITEMS



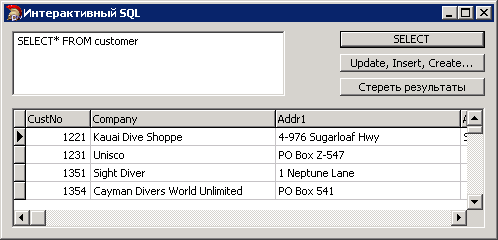
15. Модифицировать приложение, созданное при выполнении п. 12, чтобы обеспечить целостность БД с учетом того, чтов таблице ITEMS значения поля *OrderNo* не должны отличаться от номеров заказов, зафиксированных в таблице ORDERS; значения поля *PartNo* не должны отличаться от номеров поставок изделий,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

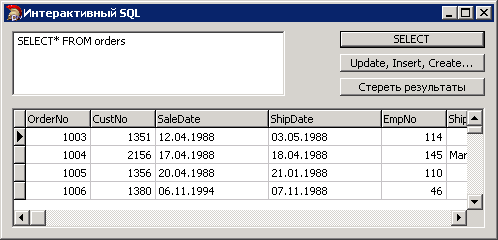


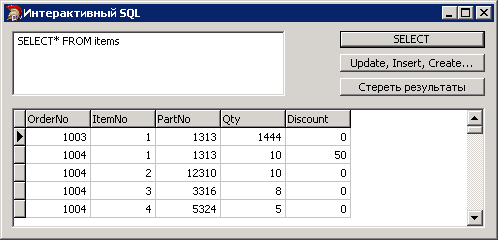
18. Разработать приложение для выполнения операторов языка SQL.

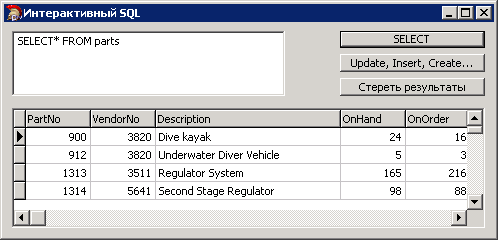


19. Запустить созданное приложение на выполнение и с помощью операторов SELECT вывести содержимое таблиц CUSTOMER, ORDERS, ITEMS, PARTS.

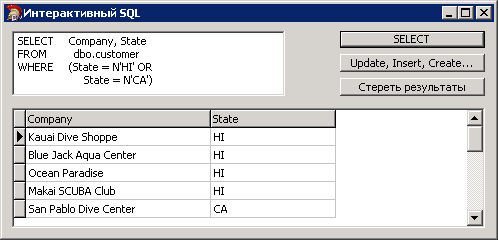


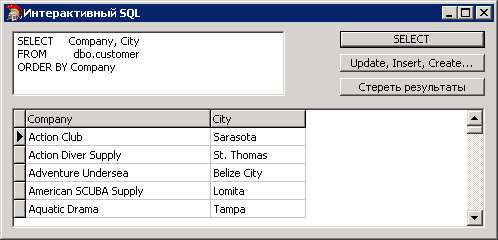


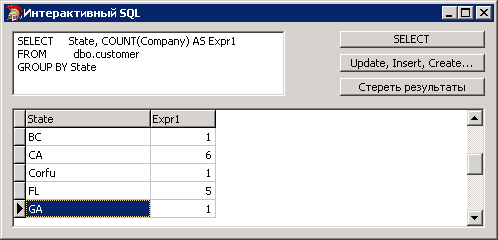




20. Для указанных таблиц выполнить записанные в отчет по лабораторной работе № 1 операторы языка SQL с выборкой, сортировкой, группировкой, изменением и добавлением данных.

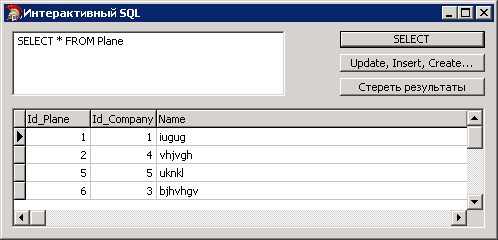


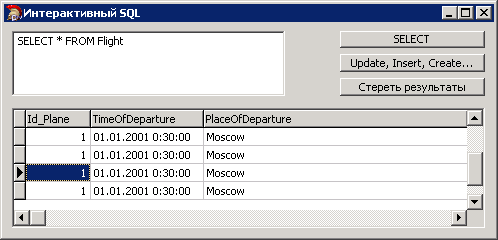


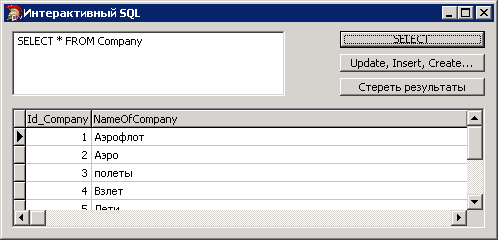


.

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.