**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.

Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=dbms\_lab;Data Source=DESKTOP-6COJ5NJ

**Ход работы**

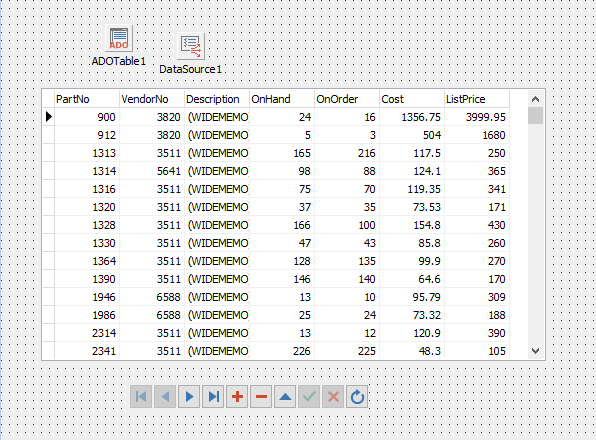


Рисунок 1. Форма типа сетка для просмотра таблицы PARTS

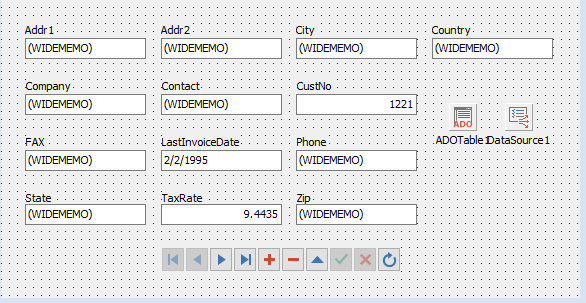


Рисунок 2. Форма типа ввод/редактирование для просмотра таблицы CUSTOMER

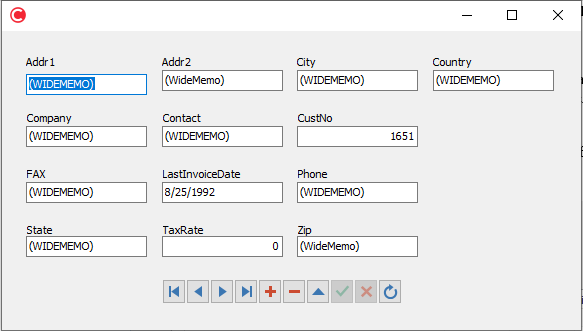


Рисунок 3.Форма типа ввод/редактирование для просмотра таблицы CUSTOMER

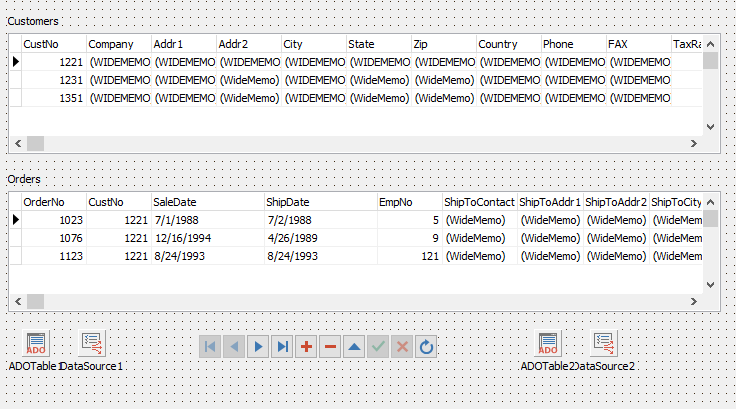


Рисунок 4. Форма типа главная/подчинённая

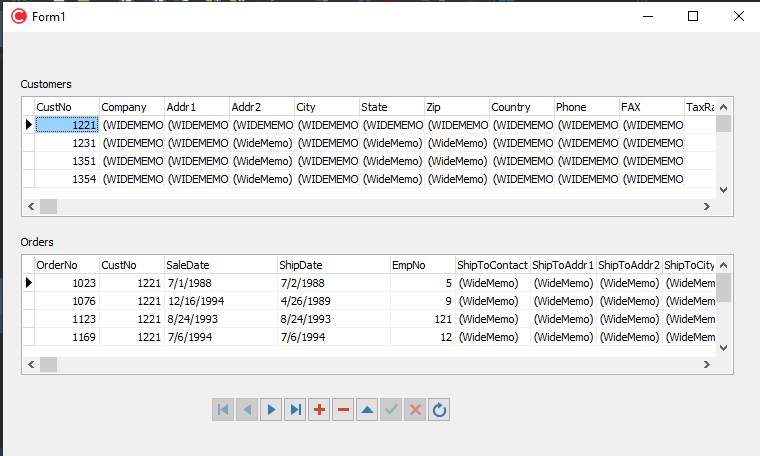


Рисунок 5. Форма типа главная/подчинённая

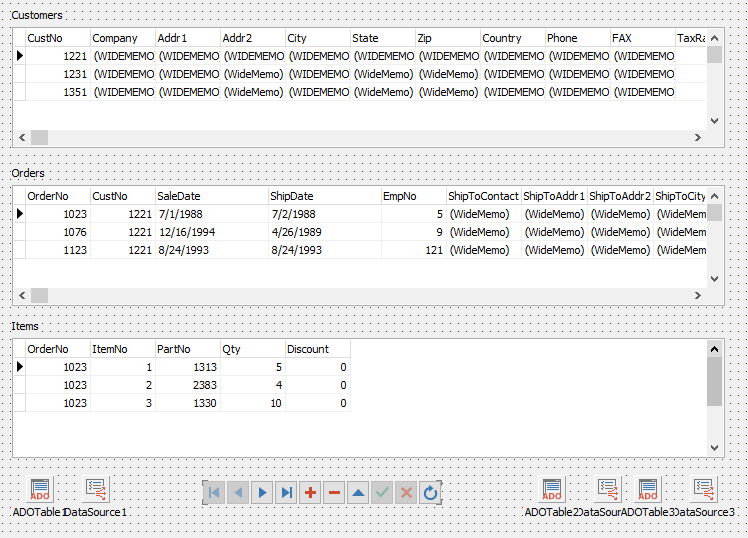


Рисунок 6. Форма типа главная/подчинённая 2

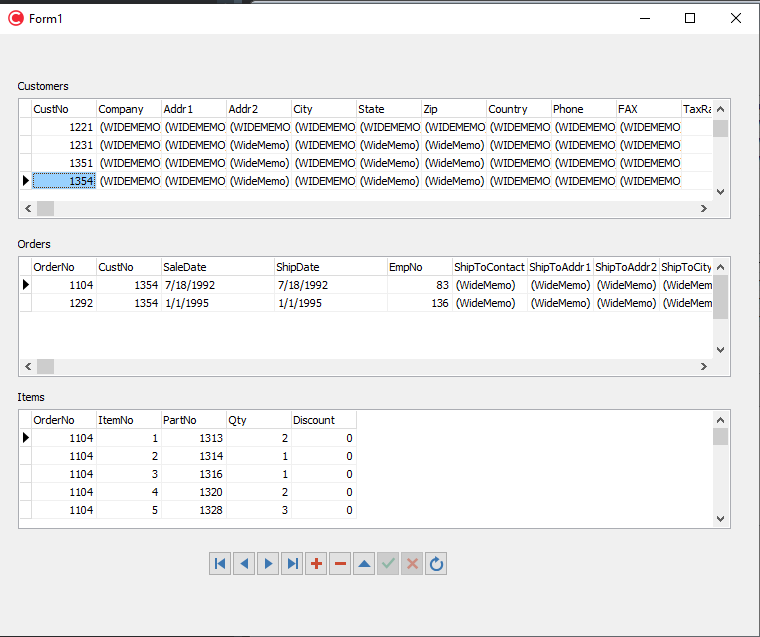


Рисунок 7. Форма типа главная/подчинённая 2

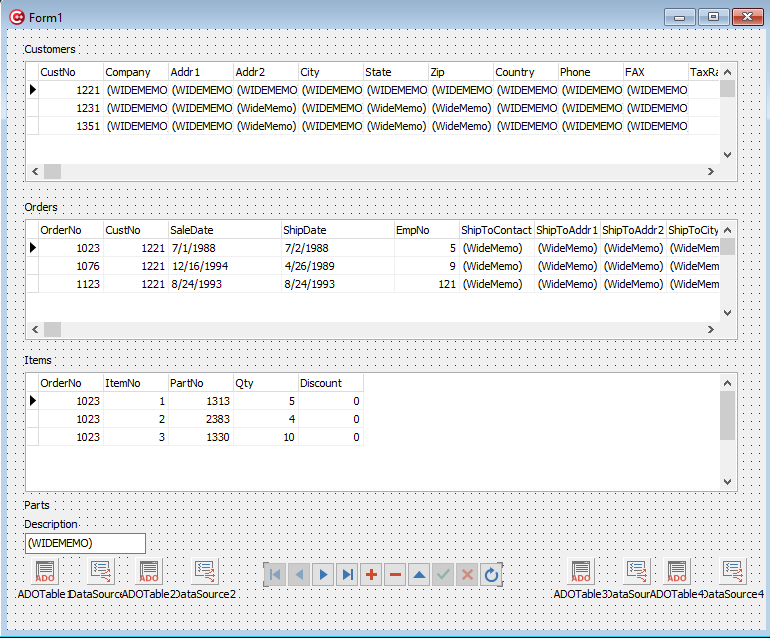


Рисунок 8. Форма типа главная/подчинённая 3

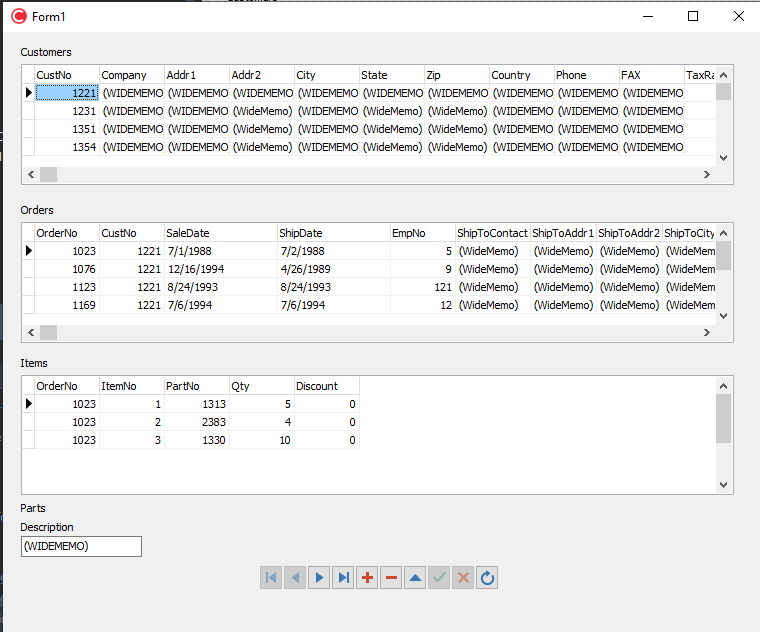


Рисунок 9. Форма типа главная/подчинённая 3

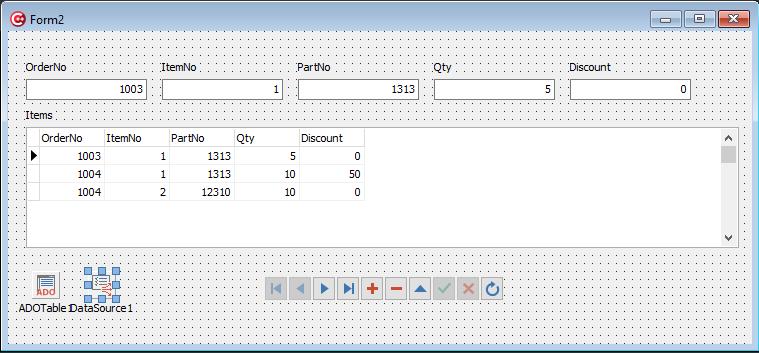


Рисунок 10. Форма типа ввод/редактирование + таблица

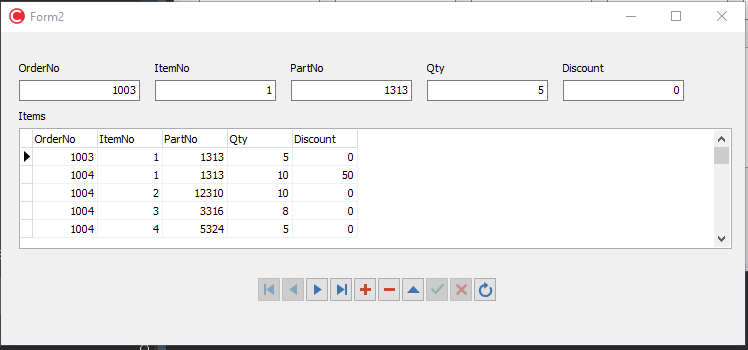


Рисунок 11. Форма типа ввод/редактирование + таблица

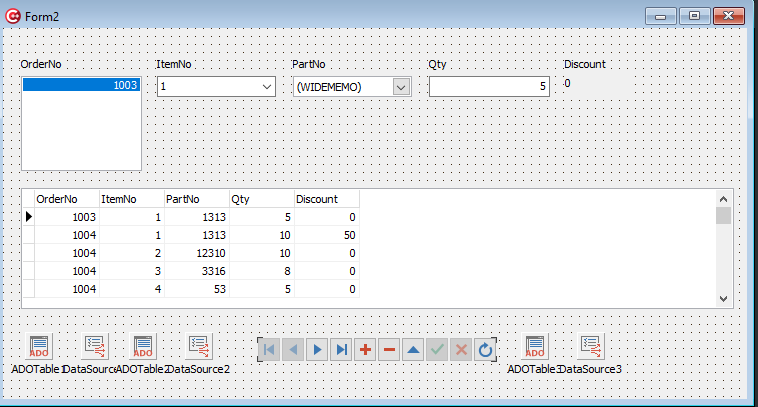


Рисунок 11. Форма типа ввод/редактирование + таблица модифицированная

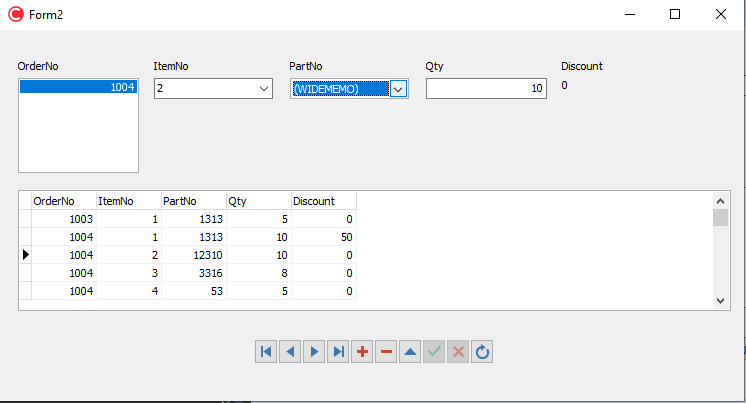


Рисунок 12. Форма типа ввод/редактирование + таблица модифицированная

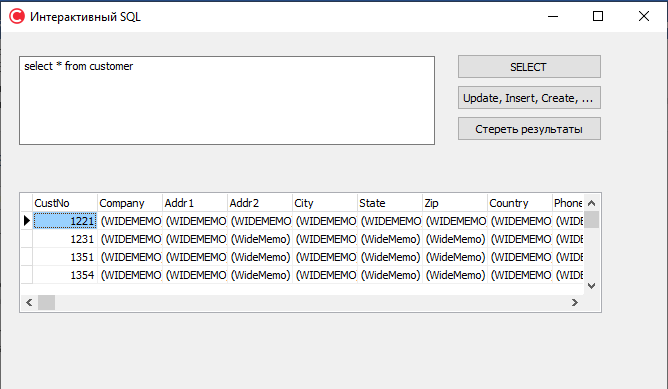


Рисунок 13. Select запрос для таблицы customer

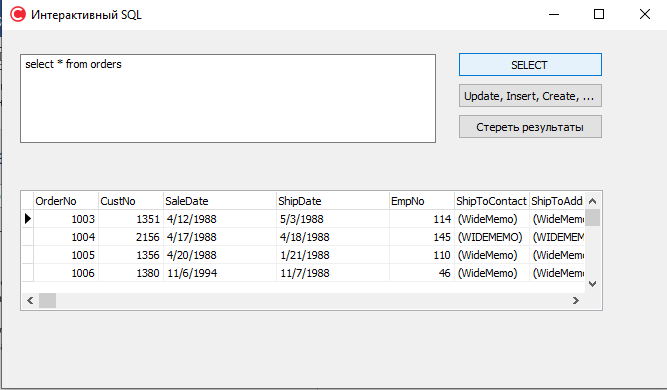


Рисунок 14. Select запрос для таблицы orders

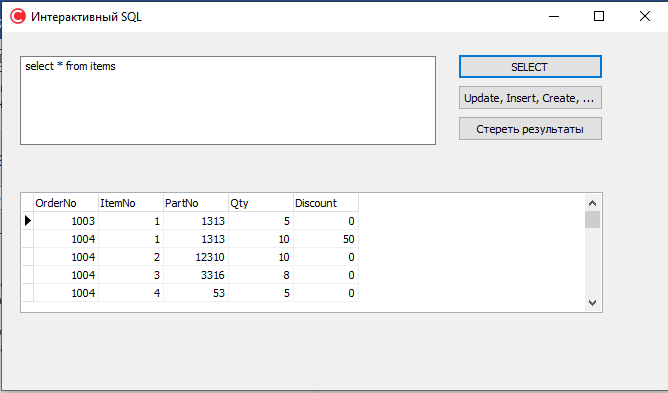


Рисунок 15. Select запрос для таблицы items

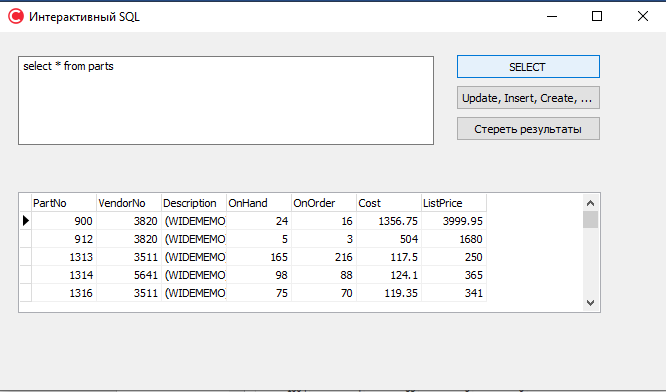


Рисунок 16. Select запрос для таблицы parts

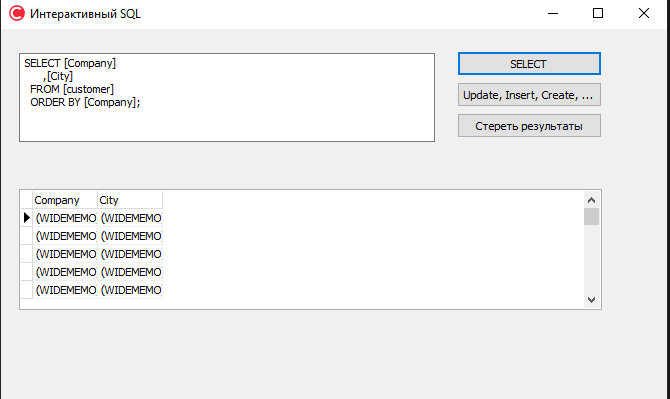


Рисунок 17. Задание 13

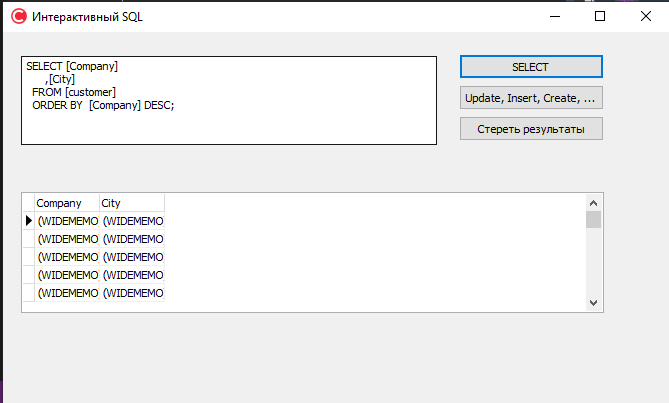


Рисунок 18. Задание 14

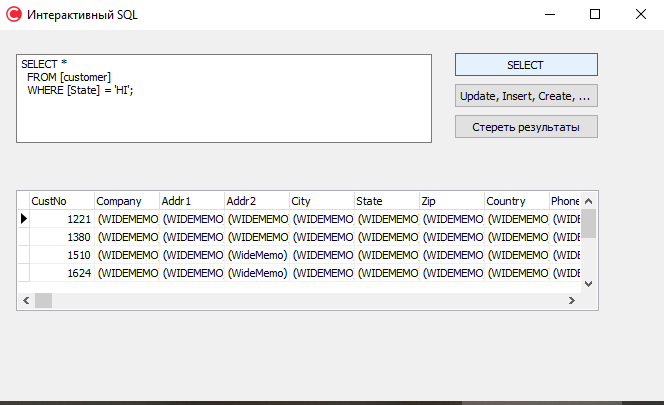


Рисунок 19. Задание 15

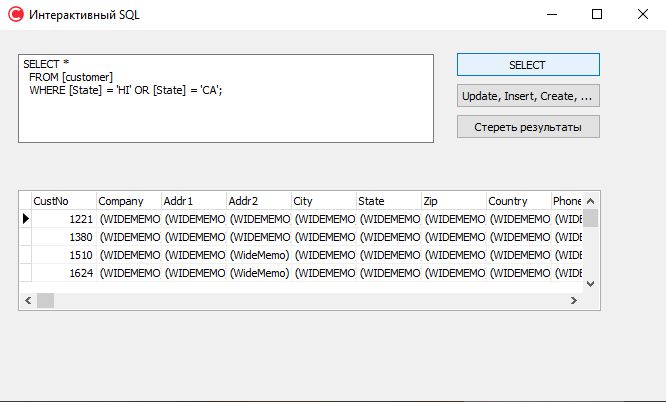


Рисунок 20. Задание 16

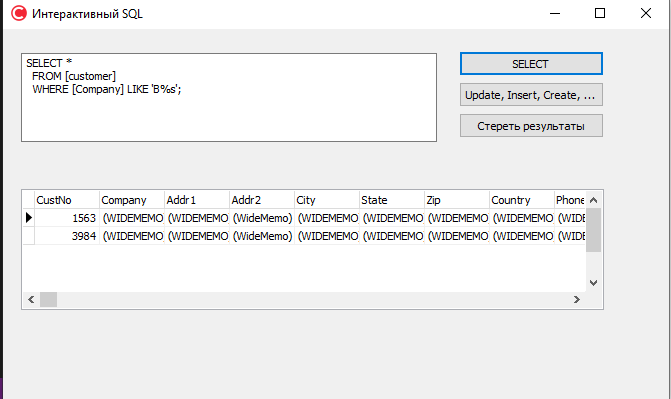


Рисунок 21. Задание 18

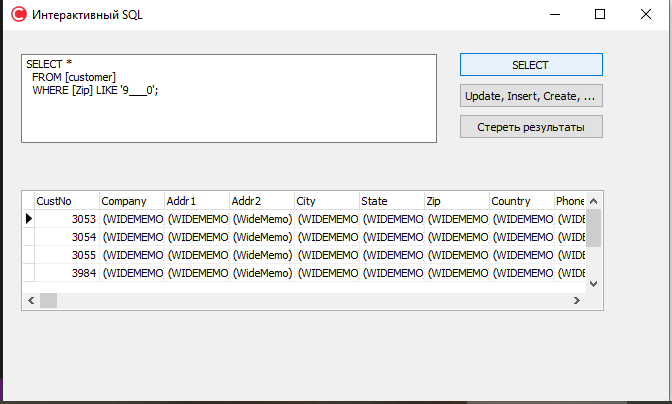


Рисунок 22. Задание 19

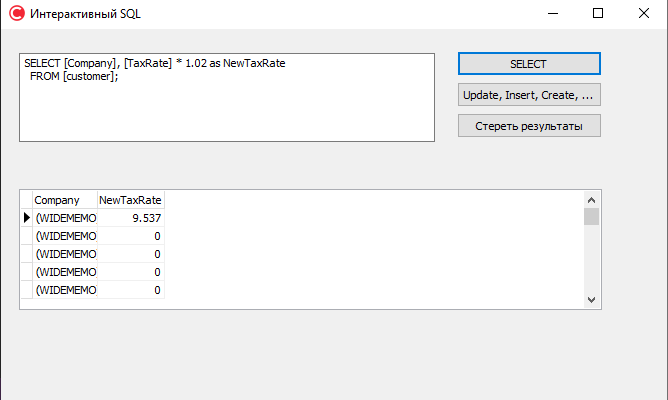


Рисунок 23. Задание 20

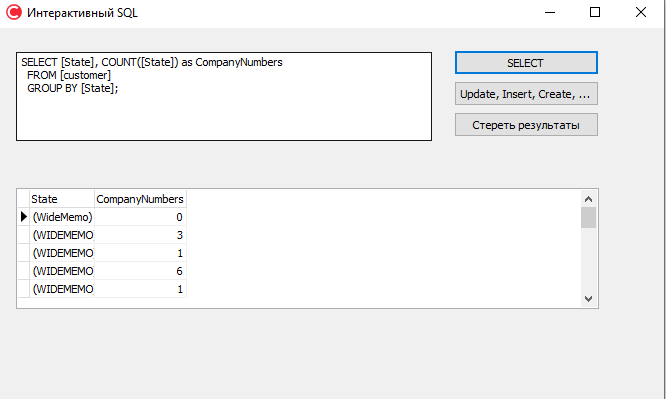


Рисунок 24. Задание 22

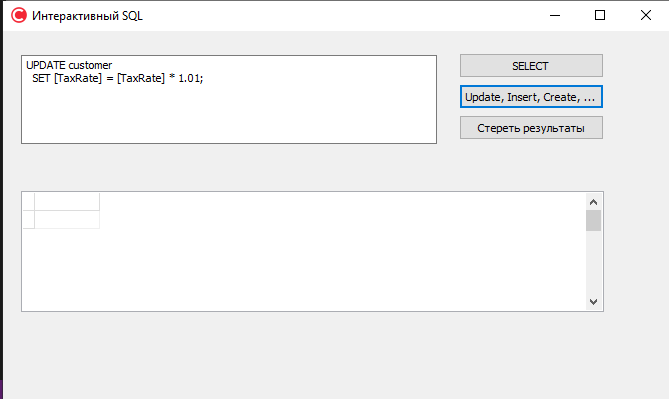


Рисунок 25. Задание 23

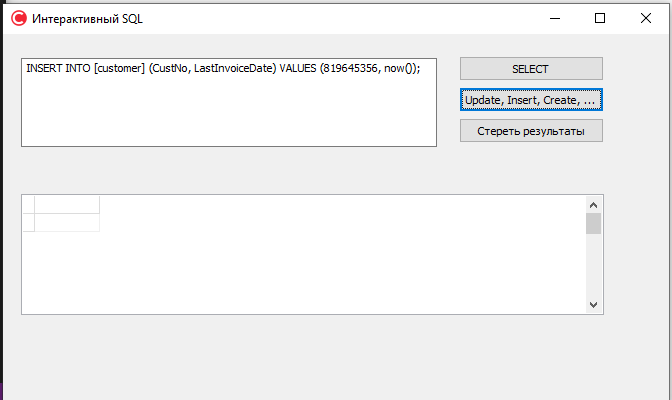


Рисунок 26. Задание 25

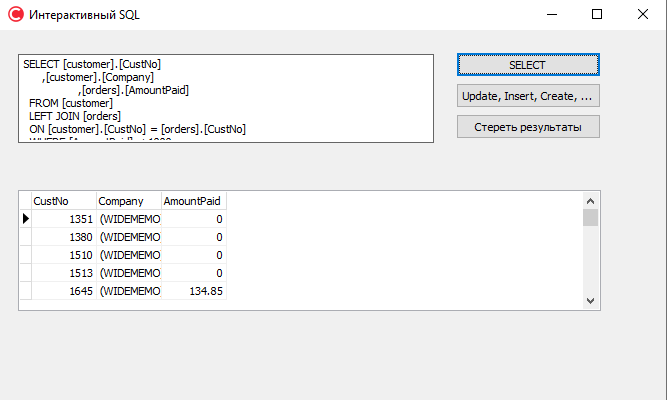


Рисунок 27. Задание 27

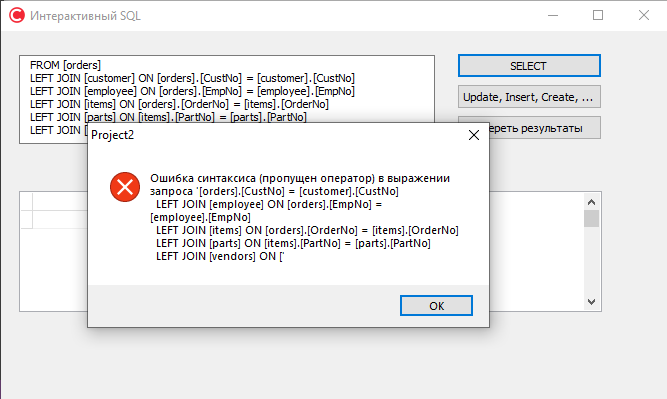


Рисунок 28. Задание 28

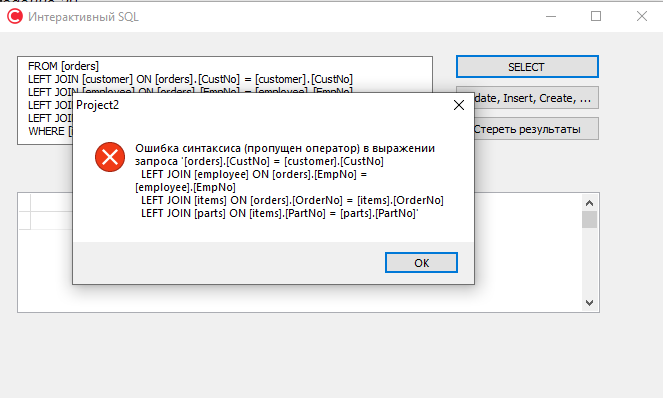


Рисунок 29. Задание 29

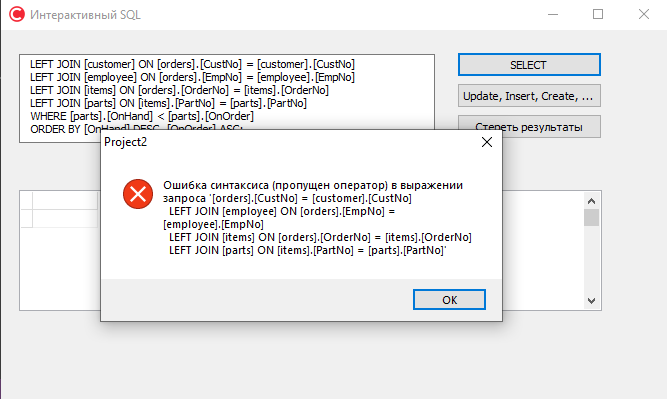


Рисунок 30. Задание 30

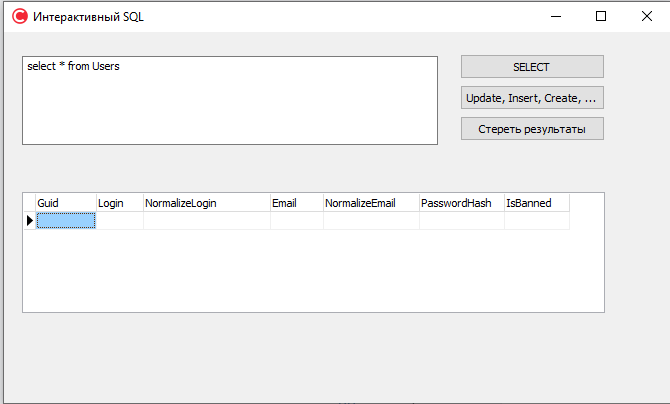


Рисунок 31. Запрос в базу данных, спроектированную в ЛР 2

**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.**

1. 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.**

1. Укажите достоинства и недостатки технологии ADO.

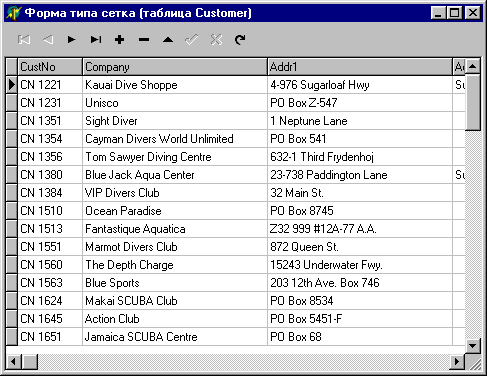
**Ответ: Основным достоинством технологии ADO является ее естественная ориентация на создание “облегченного” клиента, для которого, в отличие от технологии BDE, не требуется устанавливать специальные программные средства на клиентской машине, поскольку необходимая поддержка обеспечивается операционной системой Windows.**

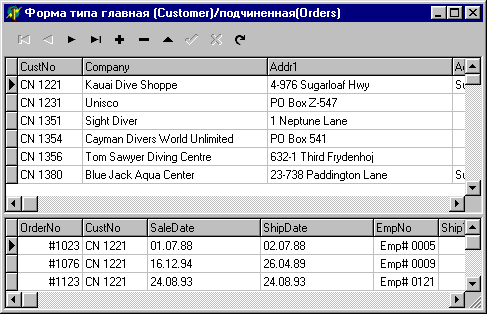
**Недостатком технологии ADO является то, что ею нельзя воспользоваться, если для соответствующей структуры данных (в частности, для БД многих популярных серверов - InterBase, Informix, DB2 и пр.) не создан нужный провайдер или ODBC-драйвер.**

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

**Ответ:**

*Рис.6.* Пример формы типа ввод/редактирование

Рис.7. Пример формы типа сетка

Рис.8. Пример формы типа главная/подчиненная

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

**Ответ: Свойство Active**

1. 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).**

1. 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).**

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

**Ответ: DataSource**

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*?

**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( ).**

1. 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.**

1. 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.**

1. 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( ).**