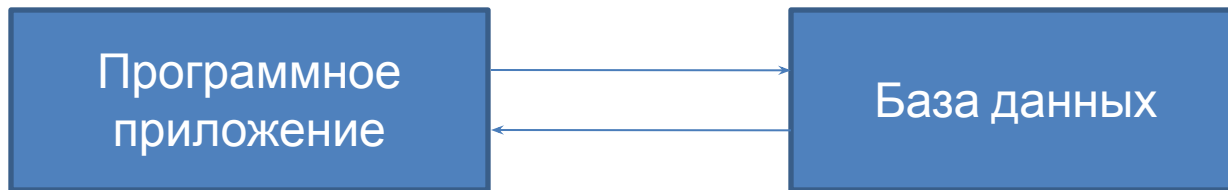


# Разработка программных приложений для работы с базами данных

- использование технологии ADO (ActiveX Data Object)



# Использование технологии ADO

В приложениях Delphi можно применять технологию Microsoft ActiveX Data Objects (ADO), которая основана на возможностях интерфейсов OLE DB.

**OLE DB** ([англ. Object Linking and Embedding, Database](#)) — набор [интерфейсов](#), основанных на [COM](#), которые позволяют [приложениям](#) обращаться к [данным](#), хранимым в разных источниках информации или хранилищах данных с помощью унифицированного доступа.

**COM** ([англ. Component Object Model](#) — модель компонентного объекта;) — это [технологический](#) стандарт от компании [Microsoft](#), предназначенный для создания [программного обеспечения](#) на основе взаимодействующих компонентов объекта, каждый из которых может использоваться во многих программах одновременно.

Технология ADO завоевала популярность у разработчиков, благодаря универсальности — базовый набор интерфейсов OLE DB имеется в каждой современной версии операционной системы Microsoft.

Технология Microsoft ActiveX Data Objects обеспечивает универсальный доступ к источникам данных из приложений БД. Такую возможность предоставляют функции набора интерфейсов, созданные на основе общей модели объектов COM и описанные в спецификации OLE DB.

Согласно терминологии ADO, любой источник данных (база данных, электронная таблица, файл) называется хранилищем данных, с которым при помощи провайдера данных взаимодействует приложение.

Так как технология ADO основана на стандартных интерфейсах COM, которые являются системным механизмом Windows, это сокращает общий объем работающего программного кода и позволяет распространять приложения БД без вспомогательных программ и библиотек.



Технология ADO в целом включает в себя не только сами объекты OLE DB, но и механизмы, обеспечивающие взаимодействие объектов с данными и приложениями. На этом уровне важнейшую роль играют провайдеры ADO, координирующие работу приложений с хранилищами данных различных типов.

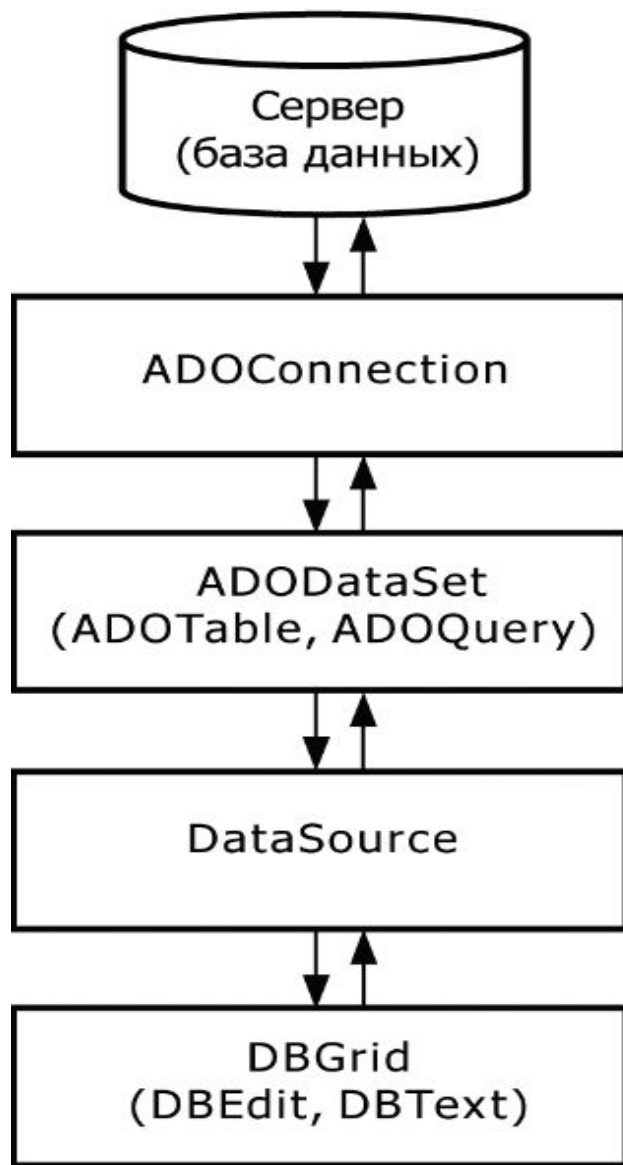
Список установленных в данной операционной системе провайдеров доступен для выбора при установке соединения через компонент **TADOCconnection.**

# Компоненты доступа к данным

Компоненты, обеспечивающие работу с базами данных находятся на вкладках

**DataControls, DataAccess, dbGo,  
dbExpress, InterBase.**





Компоненты, обеспечивающие доступ к данным

Компоненты, обеспечивающие отображение данных

# ADOConnection

Form1.ADOConnection1 ConnectionString

Source of Connection

Use Data Link File

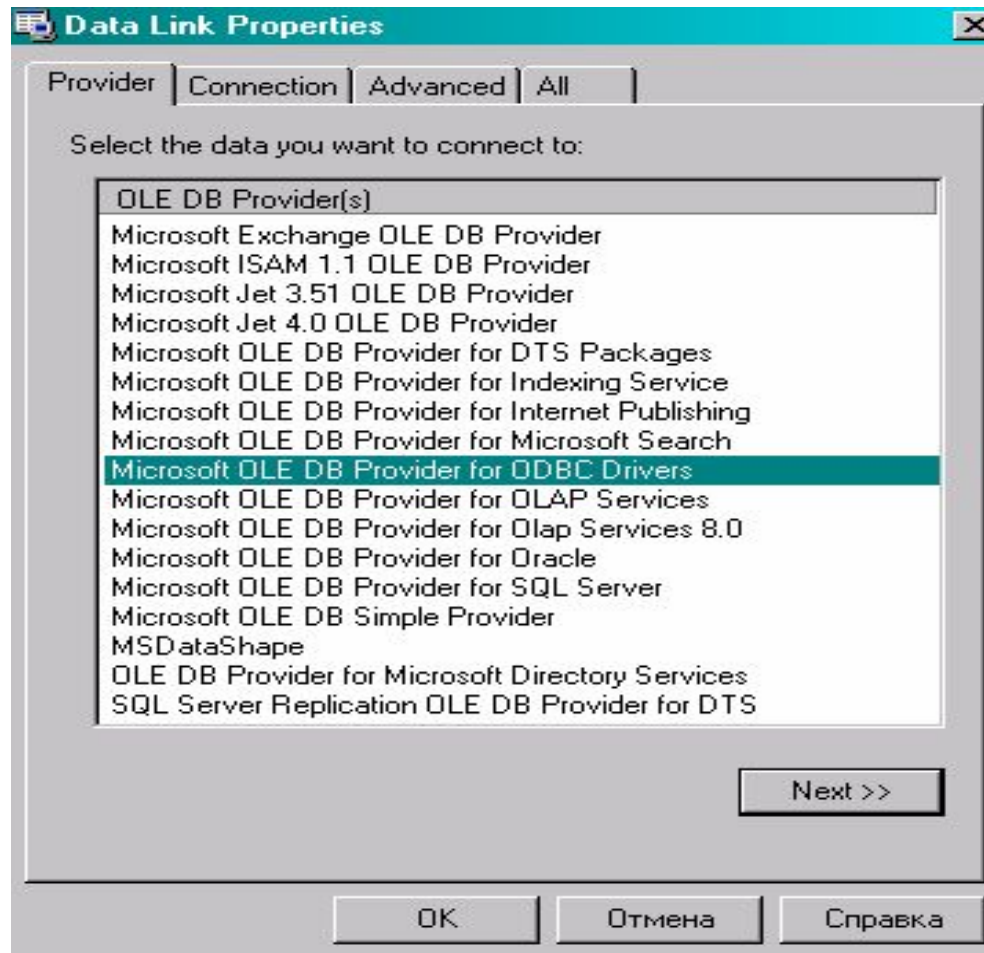
Browse...

Use Connection String

Build...

OK Cancel Help

# ADOConnection



# ADOConnection

1. В свойствах компонента ***ADOConnection*** отключить свойство ***LoginPrompt***, выставив его в ***False***. Это нужно для того, чтобы при каждом обращении к базе не нужно было вводить пароль.
2. Выставить свойство ***Connected*** в ***True***, чтобы произошло соединение с базой данных.

# ADOTable

1. Щёлкнуть по выпадающему списку в свойстве *Connection* и выбрать там единственный пункт *ADODConnection1*.
2. В свойстве *TableName* нужно выбрать *ИМЯ* своей таблицы.
3. Для подключения выставить свойство *Active* в *True*.

# DataSource

Для отображения данных из таблицы надо установить на форму компонент ***DataSource*** с закладки ***Data Access*** палитры компонентов. Теперь этому компоненту надо указать, какую именно таблицу он должен отображать. Для этого в свойстве **DataSet** нужно из выпадающего списка выбрать свою таблицу.

# DBGrid

Самый простой способ отобразить таблицу – установить компонент **DBGrid**. Это компонент-сетка, которая может отображать данные в виде таблицы. В этом же компоненте можно добавлять, удалять и редактировать строки нашей таблицы.

Для связывания компонента сетки с компонентом отображения таблицы нужно в свойстве ***DataSource*** компонента ***DBGrid*** указать созданный нами компонент ***DataSource1***.

# СОРТИРОВКА ДАННЫХ

```
Procedure TForm1.Button1Click(Sender:  
  TObject);  
BEGIN  
FORM1.ADOTABLE1.INDEXFIELDNAMES:=  
'Цена';  
End;
```



# ФИЛЬТРАЦИЯ ДАННЫХ

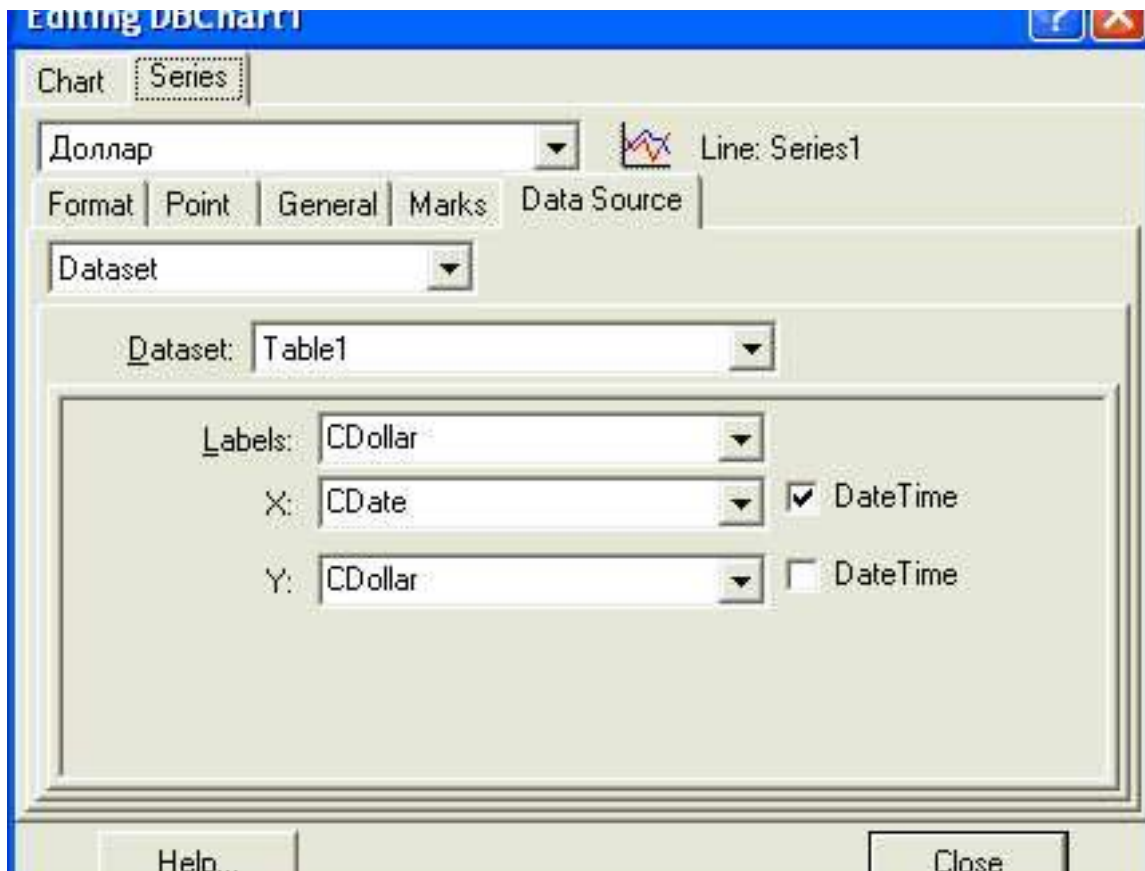
```
Procedure TForm1.Button2Click(Sender:  
  TObject);  
BEGIN  
FORM1.ADOTABLE1.FILTERED:=TRUE;  
FORM1.ADOTABLE1.FILTER:='ТОВАР=  
  КНИГИ''  
End;
```

# DBChart

Компонент DBChart является аналогом компонента Chart, но предназначенным для отображения данных из базы данных.

Серии компонента DBChart могут быть связаны с полями базы данных.

Чтобы использовать DBChart, на форме должен быть размещен один из компонентов набора данных: Table, Query и т.п. Он должен быть связан с таблицей базы данных, по которой намечено строить графики или диаграммы. Если вы хотите в процессе настройки сразу видеть в DBChart отображение данных, свойство **Active** в наборе данных установите в **true**. После окончания настройки соединение с базой данных можно разорвать, чтобы в дальнейшем устанавливать его программно во время выполнения.



## Компонент DBChart

# DBNavigator

- Свойства компонента

- **Name**

Имя компонента. Используется для доступа к свойствам компонента

- **DataSource**

Имя компонента, являющегося источником данных.

- **VisibleButtons**

Видимые командные кнопки

