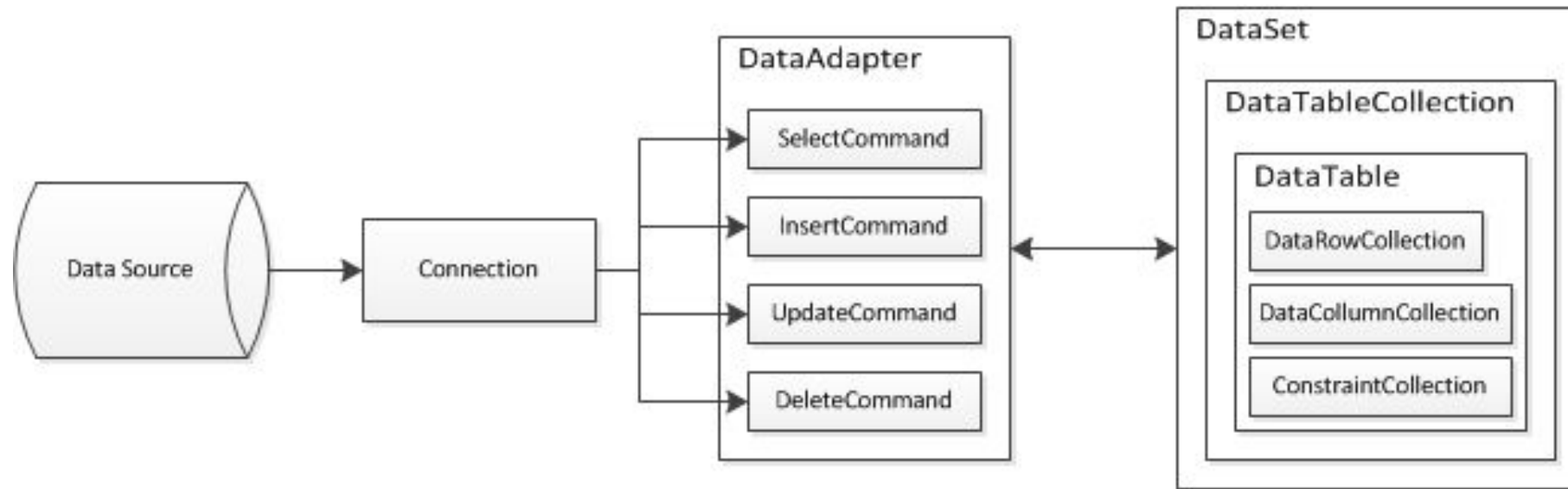


# **ADO.NET DISCONNECTED MODEL**

- 
- In-memory RDBMS
  - Work with real RDBMS
  - Typed DataSets

# DISCONNECTED MODEL КОМПОНЕНТЫ

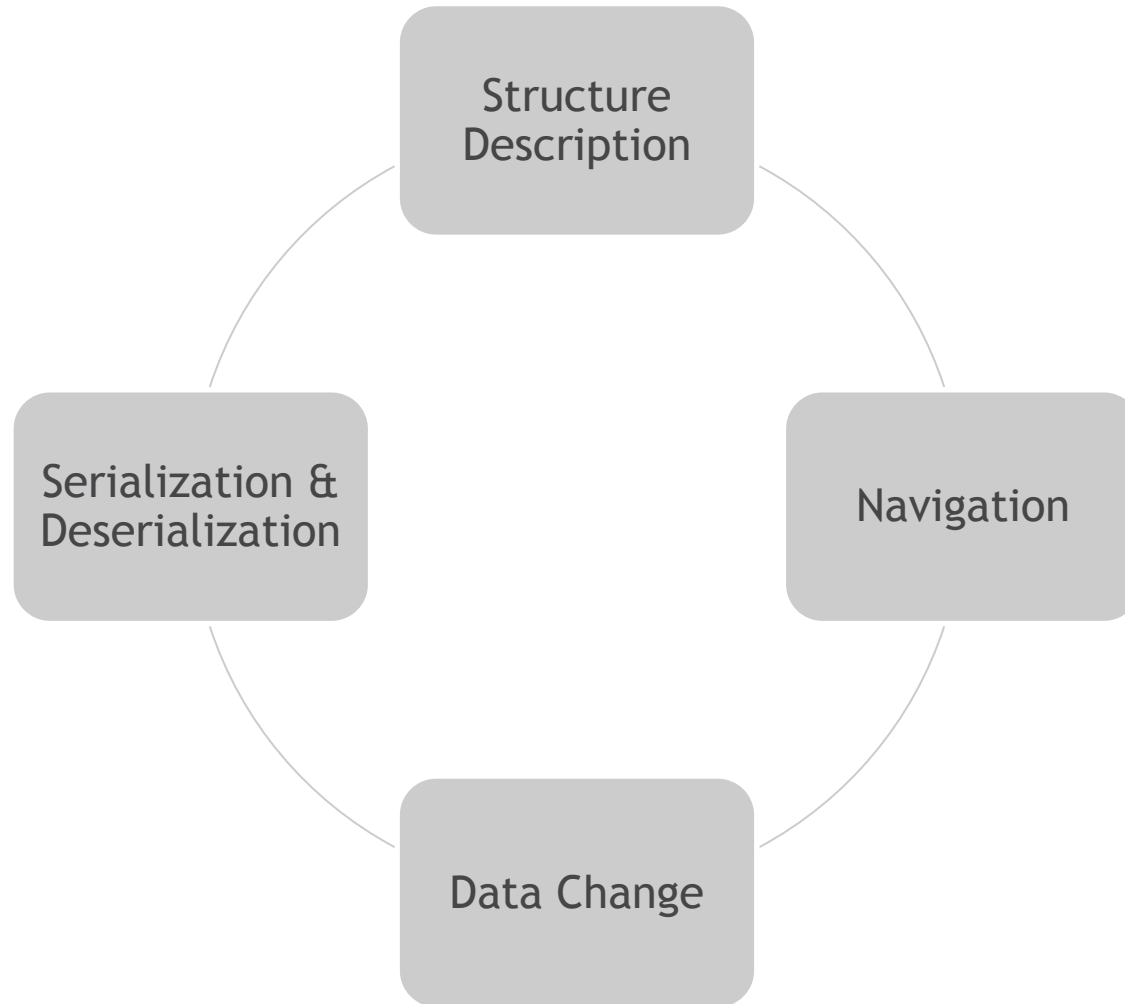


- Connection
- Command
- DataAdapter
- DataSet

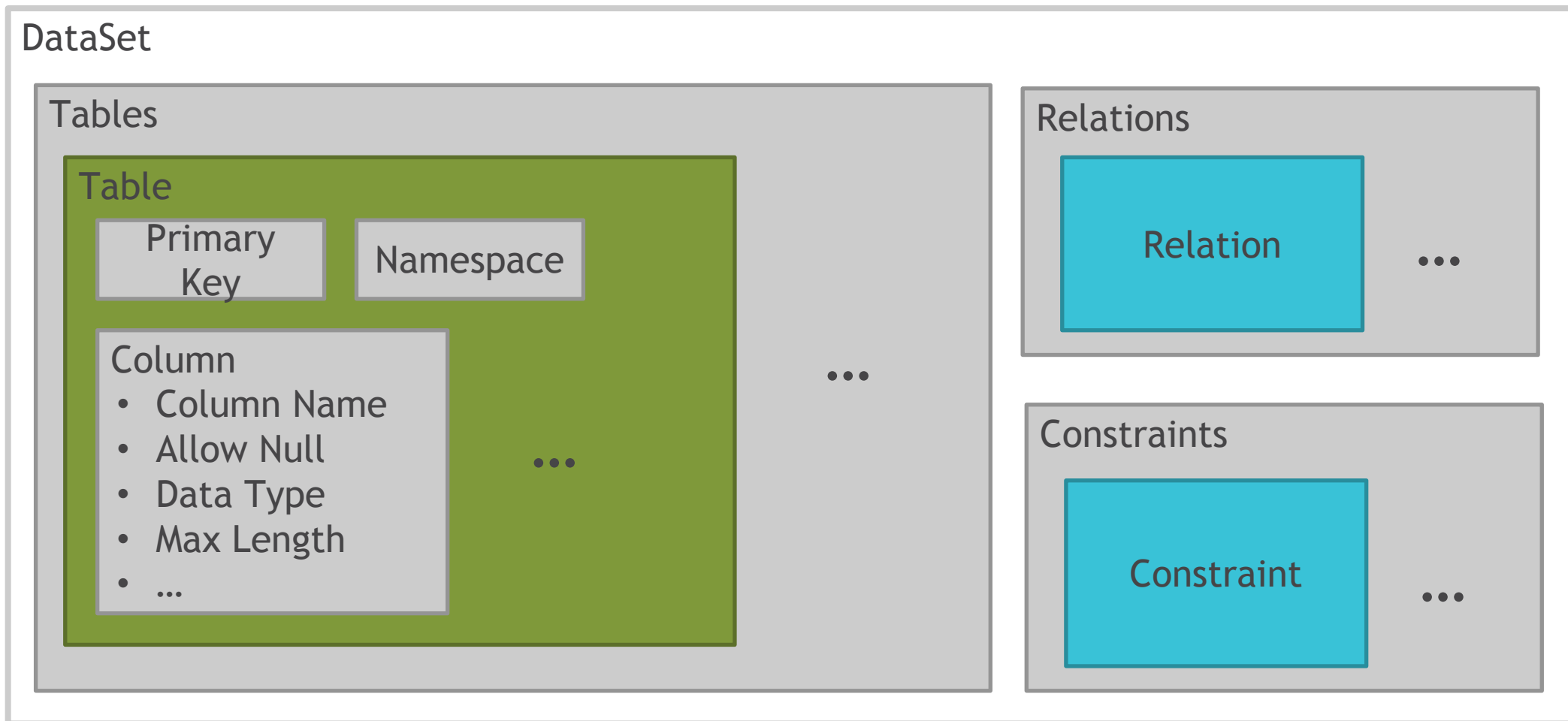
**DATA SET**

# DATASET = RDBMS В ПАМЯТИ

---



# ОПИСАНИЕ “DB” СТРУКТУРЫ



# ОПИСАНИЕ “DB” СТРУКТУРЫ. ТАБЛИЦЫ

```
var ds = new DataSet("Northwind") { Namespace = "Northwind" };

// Create table
var customerTable = ds.Tables.Add("Categories");

// Create autoincrement Id (as primary key)
var categoryId = new DataColumn()
{
    ColumnName = "CategoryID", AllowDBNull = false, DataType = typeof(int),
    AutoIncrement = true, AutoIncrementStep = 1, AutoIncrementSeed = 1
};
customerTable.Columns.Add(categoryId);
customerTable.PrimaryKey = new DataColumn[] { categoryId };

// Other columns
customerTable.Columns.Add(new DataColumn()
{
    ColumnName = "CategoryName", AllowDBNull = false, DataType = typeof(string), MaxLength = 15
});
customerTable.Columns.Add("Description", typeof (string));
customerTable.Columns.Add("Picture", typeof (byte[]));
```

# СВЯЗИ И ОТНОШЕНИЯ

```
ds.Relations.Add(  
    new DataRelation("FK_Products_Categories",  
        categoryIdColumn,  
        productTable.Columns["CategoryID"]));  
  
categoryTable.Constraints.Add(  
    new UniqueConstraint(categoryIdColumn, true));  
  
productTable.Constraints.Add(  
    new ForeignKeyConstraint(categoryIdColumn, productTable.Columns["CategoryID"])  
    {  
        DeleteRule = Rule.SetNull,  
        UpdateRule = Rule.Cascade  
    });
```



# ДОБАВЛЕНИЕ И МОДИФИКАЦИЯ

Method	Description
<code>DataRowCollection.Add(DataRow row)</code> <code>DataRowCollection.Add(params Object[] values)</code>	Add new Row in table
<code>DataRowCollection.Remove(DataRow row)</code> <code>DataRowCollection.RemoveAt(int index)</code> <code>DataRowCollection.Clear()</code>	Permanent delete row (without rollback possibility)
<code>DataRow[DataColumn column]</code> <code>DataRow[Int32 columnIndex]</code> <code>DataRow[string columnName]</code>	Change data value
<code>DataRow.Delete()</code>	Mark row as deleted (but not physical delete)

# ADD/CHANGE/DELETE ИЗ КОДА

```
var categoriesTable = ds.Tables["Categories"];
var productsTable = ds.Tables["Products"];

// Add new data
categoriesTable.Rows.Add(null, "Beverages", "Soft drinks, coffees, teas, beers, and ales", null);
categoriesTable.Rows.Add(null, "Condiments", "Sweet and savory sauces, relishes, spreads, and seasonings", null);

productsTable.Rows.Add(null, "Chai", "10 boxes x 20 bags", 18.00m, 39, 0, 10, false, 1);
productsTable.Rows.Add(null, "Chang", "24 - 12 oz bottles", 19.00m, 17, 40, 25, false, 1);
productsTable.Rows.Add(null, "Chef Anton's Cajun Seasoning", "48 - 6 oz jars", 22.00m, 53, 0, 0, false, 2);

// Change price for one product
var chaiRow = productsTable.Rows[1];
chaiRow["UnitPrice"] = (decimal)chaiRow["UnitPrice"] + 1;

// Delete one product
productsTable.Rows[2].Delete();
```

# ФИЛЬТРАЦИЯ

```
foreach (DataRow product in productsTable.Select("UnitPrice >= 20"))  
{  
    Console.WriteLine(product["ProductName"]);  
}
```

Operators	Sample
Comparison and logical operators	City <> 'Tokyo' AND City <> 'Paris'
IN	Id IN (1, 2, 3)
LIKE	Name LIKE 'j*'
Arithmetic and string operators	Age % 10 = 0
Aggregate Functions	Salary > AVG(Salary)

# MASTER / DETAIL ОТНОШЕНИЯ

```
ds.Relations.Add(  
    new DataRelation("FK_Products_Categories",  
        categoryIdColumn,  
        productTable.Columns["CategoryID"]));
```

Describe relation

```
foreach (DataRow category in categoriesTable.Rows)  
{  
    Console.WriteLine(category[1]);  
  
    foreach (DataRow product in category.GetChildRows("FK_Products_Categories"))  
        Console.WriteLine("\t{0}", product[1]);  
}
```

Navigate

# РАБОТА С XML

## Read / Write XML

```
var ds = CreateNorthwindDataSet();  
  
ds.ReadXml("data.xml");  
var productsTable = ds.Tables["Products"];  
productsTable.Rows.Add(null, "New product", "1", 122.00m, 1, 0, 0, false, 2);  
  
ds.WriteXml("new_data.xml");
```

## Read / Write XML Schema

```
var ds = new DataSet();  
ds.ReadXmlSchema("schema.xsd");  
  
var categoriesTable = ds.Tables["Categories"];  
categoriesTable.Columns.Add("Favorite", typeof(bool));  
  
ds.WriteXmlSchema("new_schema.xsd");
```

РАБОТА С **RDBMS**

# ЗАГРУЗКА ДАННЫХ

---

```
var command = connection.CreateCommand();

command.CommandText =
    "select CategoryID, CategoryName, Description, Picture " +
    " from Northwind.Categories";

ds.Tables["Categories"].Load(command.ExecuteReader());
```

# DATA ADAPTER

---

```
var adapter = new SqlDataAdapter(  
    "select * from Northwind.Categories as Categories; " +  
    " select * from Northwind.Products",  
    connection);  
  
adapter.TableMappings.Add("Table", "Categories");  
adapter.TableMappings.Add("Table1", "Products");  
  
adapter.Fill(ds);
```



# МАППИНГ

---

```
var adapter = new SqlDataAdapter(  
    "select ProductName, " +  
    " UnitPrice * (UnitsInStock + UnitsOnOrder) * (1 - Discontinued) TotalCost " +  
    " from Northwind.Products", connection);  
  
var mapping = adapter.TableMappings.Add("Table", "ProductCost");  
  
mapping.ColumnMappings.Add("ProductName", "Name");  
mapping.ColumnMappings.Add("TotalCost", "Cost");  
  
adapter.Fill(ds);
```

# UPDATE

```
var updateCommand = new SqlCommand(
    "update Northwind.Categories " +
    " set CategoryName = @p1, Description = @p2, Picture = @p3 " +
    " where CategoryID = @p4", connection);

updateCommand.Parameters.Add("@p1", SqlDbType.NVarChar, 15, "CategoryName");
updateCommand.Parameters.Add("@p2", SqlDbType.NText, 0, "Description");
updateCommand.Parameters.Add("@p3", SqlDbType.Image, 0, "Picture");

updateCommand.Parameters.Add(
    new SqlParameter()
    {
        ParameterName = "@p4",
        SourceColumn = "CategoryID",
        SourceVersion = DataRowVersion.Original
    });

adapter.UpdateCommand = updateCommand;
```

```
adapter.Fill(ds);

ds.Tables["Categories"].Rows[12]["CategoryName"] = "!!!";

adapter.Update(ds);
```

# INSERT

```
var insertCommand = new SqlCommand(
    "insert into Northwind.Categories " +
    " (CategoryName, Description, Picture) " +
    " values(@p1, @p2, @p3); " +

    "select SCOPE_IDENTITY() as CategoryID", connection);

insertCommand.Parameters.Add("@p1", SqlDbType.NVarChar, 15, "CategoryName");
insertCommand.Parameters.Add("@p2", SqlDbType.NText, 0, "Description");
insertCommand.Parameters.Add("@p3", SqlDbType.Image, 0, "Picture");

insertCommand.UpdatedRowSource = UpdateRowSource.FirstReturnedRecord;

adapter.InsertCommand = insertCommand;
```

```
adapter.Fill(ds);

var newRow = ds.Tables["Categories"].Rows.
    Add(null, "New Category", "New Category description");

adapter.Update(ds);

Console.WriteLine(newRow["CategoryID"]);
```

# COMMAND BUILDER

```
var categoriesAdapter = new SqlDataAdapter(  
    "select * from Northwind.Categories as Categories", connection);  
categoriesAdapter.TableMappings.Add("Table", "Categories");  
  
var categoriesCommandBuilder = new SqlCommandBuilder(categoriesAdapter);  
  
categoriesAdapter.InsertCommand = categoriesCommandBuilder.GetInsertCommand();  
categoriesAdapter.UpdateCommand = categoriesCommandBuilder.GetUpdateCommand();  
categoriesAdapter.DeleteCommand = categoriesCommandBuilder.GetDeleteCommand();  
  
categoriesAdapter.Fill(ds);
```