



Ado.Net



Дмитрий Шалымов

Ведущий курса

dmitry.shalymov@veeamacademy.ru

Введение в Ado.Net

- Набор классов для взаимодействия с реляционными базами данных
- Унифицированный интерфейс
 - MS SQL Server, MySQL, Oracle, Access, DB2
- Гибкий выбор поставщика данных

Способы работы с Ado.Net

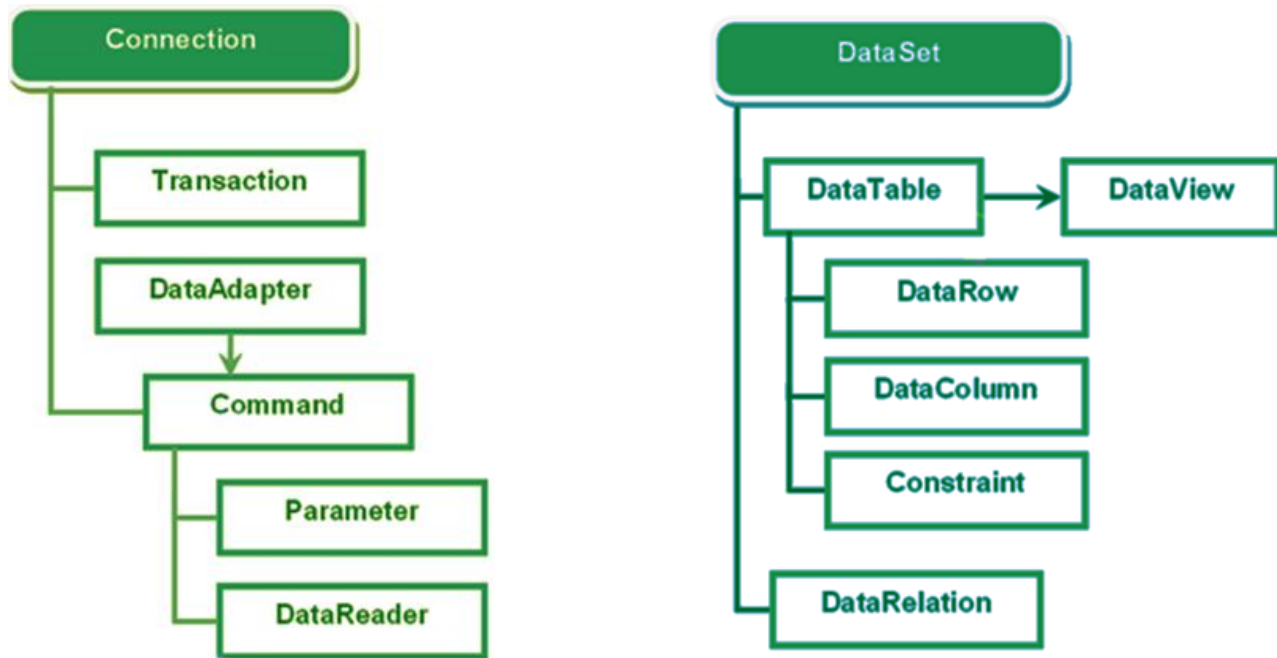
- Подключенный уровень
 - объект подключения, объект команды, объект чтения данных
- Автономный уровень
 - DataSet, DataTable
- Entity Framework (EF)



Основные объекты для работы

- Connection
- Command
- DataReader
- DataSet
- DataAdapter

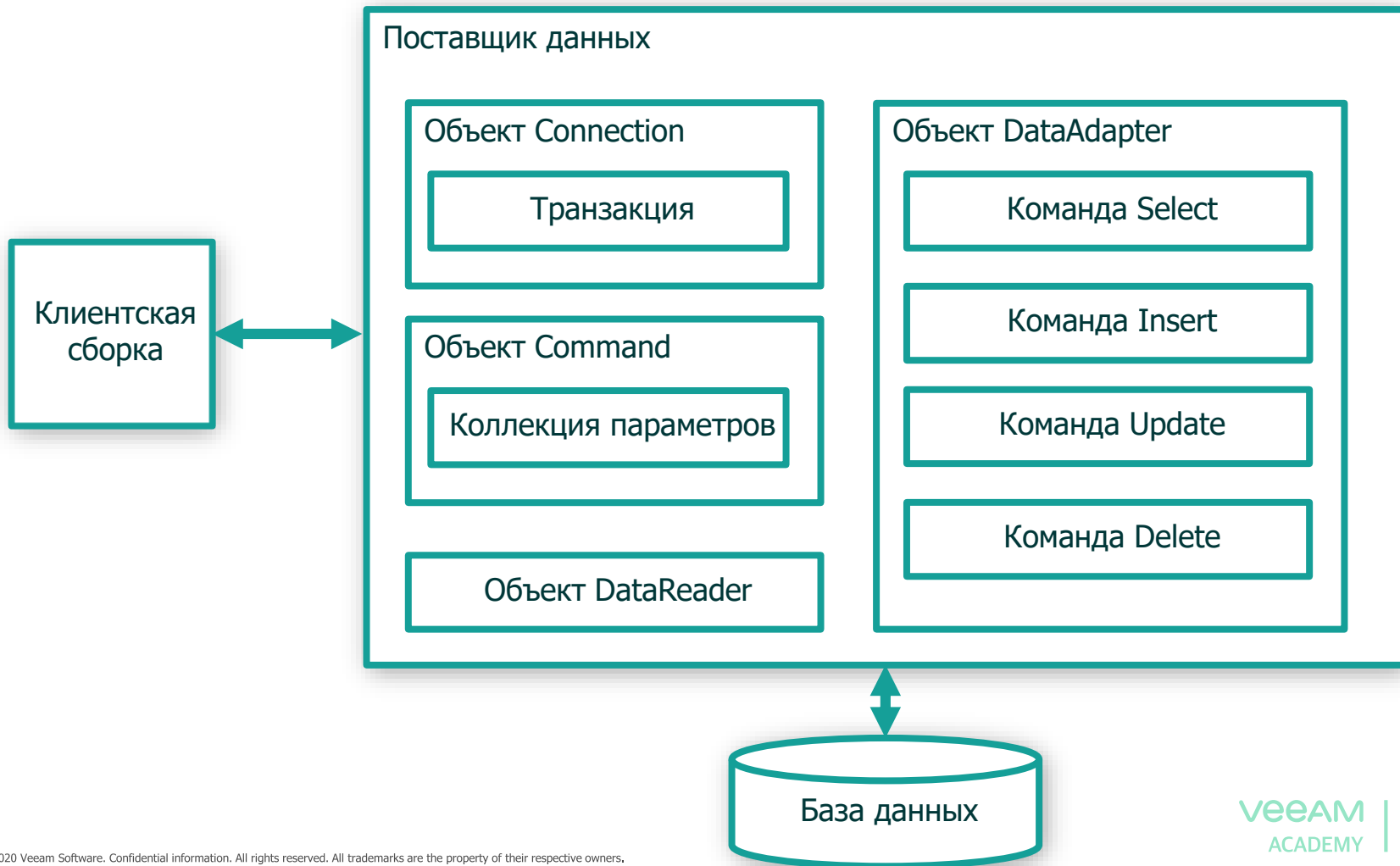
Объектная модель ADO.NET



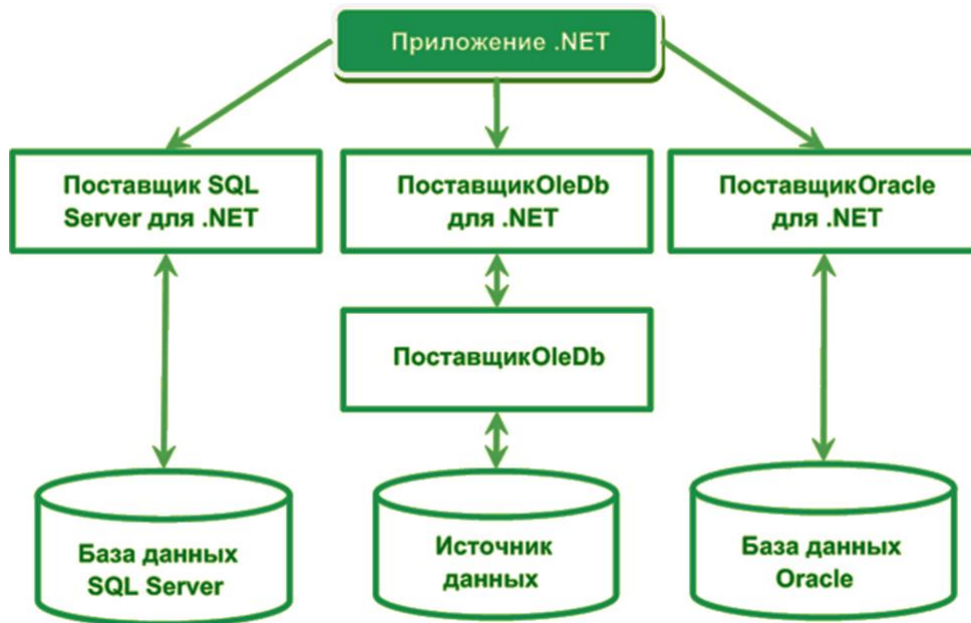
Поставщики данных

- Нет единого набора типов
- Есть различные поставщики данных
- Основные объекты поставщиков данных
 - Connection
 - Command
 - Parameter
 - DataReader
 - DataAdapter
 - Transaction





Уровни моделей поставщиков ADO.NET



The background is a dark teal color. It features several large, flowing, wavy shapes in a lighter teal shade. These shapes are composed of a dense pattern of small dots, creating a textured, organic feel. The text is centered horizontally and partially overlaid by these wavy shapes.

Подключенный уровень

Понятие подключенного уровня

Шаги для работы с БД:

1. Создать объект подключения
2. Создать объект команды
3. Вызвать метод ExecuteReader() у объекта команды
4. Обработать записи с помощью метода Read()

```
using (SqlConnection cn = new SqlConnection())
{
    cn.ConnectionString = connectionString;

    cn.Open();

    SqlCommand command = new SqlCommand();
    command.CommandText = sqlExpression;
    command.Connection = connection;

    SqlDataReader reader = command.ExecuteReader();

    while (reader.Read()) // построчно считываем данные
    {
        object id = reader.GetValue(0);
        object name = reader.GetValue(1);
        object age = reader.GetValue(2);
    }
}
```

```
// Создать и открыть подключение
using (SqlConnection cn = new SqlConnection())
{
    cn.ConnectionString =
        @"Data Source=(local)\SQLEXPRESS;Integrated Security=SSPI;" +
        "Initial Catalog=TestDB";
    cn.Open();

    // Создать объект команды SQL
    string strSQL = "Select * From Department";
    SqlCommand myCommand = new SqlCommand(strSQL, cn);

    // Получить объект чтения данных с помощью ExecuteReader()
    using (SqlDataReader myDataReader = myCommand.ExecuteReader())
    {
        // Организовать цикл по результатам
        while (myDataReader.Read())
        {
            Console.WriteLine(
                $"Office name: {myDataReader["Name"]}, Location: {myDataReader["Location"]}");
        }
    }
}
```

SqlCommand

Чтобы выполнить команду, необходимо применить один из методов:

- ExecuteNonQuery
 - INSERT, UPDATE, DELETE
- ExecuteReader
 - SELECT
- ExecuteScalar
 - Min, Max, Sum, Count

Параметризированные объекты команд

Использование SQL параметров в виде объектов

- Сокращает количество опечаток
- Обычно выполняются быстрее
- Защищают от SQL injection атак



```
private void InsertColor(int id, string name, SqlConnection connection)
{
    string sql = string.Format("Insert Into Colors" +
    " (ColorID, Name) Values" +
    "(@ColorID, @Name)");

    using (SqlCommand cmd = new SqlCommand(sql, connection))
    {
        SqlParameter param = new SqlParameter();
        param.ParameterName = "@ColorID";
        param.Value = id;
        param.SqlDbType = SqlDbType.Int;
        cmd.Parameters.Add(param);

        param = new SqlParameter();
        param.ParameterName = "@Name";
        param.Value = name;
        param.SqlDbType = SqlDbType.Char;
        param.Size = 10;
        cmd.Parameters.Add(param);

        cmd.ExecuteNonQuery();
    }
}
```

Пул подключений

- Использование ранее созданных подключений
- Поддерживается несколько пулов одновременно
- Min Pool Size
- Менеджер подключений



The background is a solid dark teal color. Overlaid on this are several large, flowing, wavy shapes in a lighter teal shade. These shapes are composed of numerous small dots, creating a halftone or mesh-like texture. The overall effect is modern and technological.

Автономный уровень

Автономный уровень в Ado.Net

Можно использовать объектную модель Ado.Net **автономно**

Моделирование реляционных данных с помощью модели объектов, **находящихся в памяти:**
таблицы, отношения между таблицами, ограничения, первичные ключи и др.

Адаптер данных

Выбирает и обновляет данные с помощью объектов DataSet

DataSet -> DataTable -> DataRow, DataColumn

Адаптеры данных удерживают подключение открытым в течение минимально возможного времени



Dummy Shop и Ado.Net

Безопасность базы данных

Безопасность базы данных

- Создавайте учетные записи с **минимально возможными правами** доступа
- Не предоставляйте пользователям доступ к **административным** учетным записям
- Не возвращайте **серверные сообщения об ошибках** в клиентские приложения
- **Проверяйте весь ввод** как на сервере, так и на клиенте
- Используйте **параметризованные команды**
- Включайте **аудит безопасности** и ведение журнала в используемой базе данных

Thank you

veeam
ACADEMY

