

# Модели данных Data Access Layer

# Модели данных DAL

- Table Data Gateway
- Row Data Gateway
- Active Record
- Data Mapper

# Table Data Gateway

Класс, который является Gateway для одной таблицы в базе данных. Один экземпляр работает со всеми строками в этой таблице

Инкапсулирует логику преобразования запросов к таблице или представлению этой таблицы в реляционной базе данных в SQL запросы: SELECT, INSERT, UPDATE, DELETE и т.д.

# Table Data Gateway

*An object that acts as a Gateway (466) to a database table.  
One instance handles all the rows in the table.*

Person Gateway
<pre>find (id) : RecordSet findWithLastName(String) : RecordSet update (id, lastname, firstname, numberOfDependents) insert (lastname, firstname, numberOfDependents) delete (id)</pre>

# Table Data Gateway. Как он работает?

Обычно имеет простой интерфейс, который содержит несколько методов для получения доступа к данным из БД, а также методы для обновления, вставки и удаления данных

Table Data Gateway обычно не имеет состояния (stateless)

Интересной задачей является возврат результатов поиска в таблице. Варианты:

- Простая структура данных (например, map)
- Data Transfer Object
- Полный Record Set
- Domain object в случае Domain Model (рассмотрим позднее)

# Table Data Gateway. Когда использовать?

Первое, что нужно понять – будет ли использоваться Gateway вообще

Очень хорошо себя показывает, если для слоя бизнес-логики был выбран Table Module (рассмотрим позднее)

Также может быть использован в качестве прослойки между Data Mapper и базой данных

# Table Data Gateway. Пример без Data Sets

```
1  using System.Data;
2  using System.Data.SQLite;
3
4  namespace PoEAA_TableDataGateway.ReaderVersion
5  {
6      class PersonGateway
7      {
8          public IDataReader FindAll()
9          {
10              string sql = "select * from person";
11              var conn = DbManager.CreateConnection();
12              conn.Open();
13              return new SQLiteCommand(sql, conn).ExecuteReader();
14          }
15      }
16  }
```

# Table Data Gateway. Пример без Data Sets

```
16         public IDataReader FindWithLastName(string lastName)
17     {
18         string sql = "select * from person where lastname = $lastname";
19         var conn = DbManager.CreateConnection();
20         conn.Open();
21         IDbCommand comm = new SQLiteCommand(sql, conn);
22         comm.Parameters.Add(new SQLiteParameter("$lastname", lastName));
23
24         return comm.ExecuteReader();
25     }
26
27     public IDataReader FindWhere(string whereClause)
28     {
29         string sql = $"select * from person where {whereClause}";
30         var conn = DbManager.CreateConnection();
31         conn.Open();
32         return new SQLiteCommand(sql, conn).ExecuteReader();
33     }
```



# Table Data Gateway. Пример без Data Sets

```
35         public object[] FindRow(long key)
36     {
37         string sql = "select * from person where id = $id";
38         using var conn = DbManager.CreateConnection();
39         conn.Open();
40         using IDbCommand comm = new SQLiteCommand(sql, conn);
41         comm.Parameters.Add(new SQLiteParameter("$id", key));
42         using IDataReader reader = comm.ExecuteReader();
43         reader.Read();
44         object[] result = new object[reader.FieldCount];
45         reader.GetValues(result);
46         return result;
47     }
```

# Table Data Gateway. Пример без Data Sets

```
49     public void Update(long key, string lastName, string firstName, int numberOfDependents)
50     {
51         string sql =
52             @"Update person SET lastname = $lastname, firstname = $firstname, numberOfDependents = $numberOfDependents
53                 WHERE id = $id";
54         using var conn = DbManager.CreateConnection();
55         conn.Open();
56         using IDbCommand comm = new SQLiteCommand(sql, conn);
57         comm.Parameters.Add(new SQLiteParameter("$lastname", lastName));
58         comm.Parameters.Add(new SQLiteParameter("$firstname", firstName));
59         comm.Parameters.Add(new SQLiteParameter("$numberOfDependents", numberOfDependents));
60         comm.Parameters.Add(new SQLiteParameter("$id", key));
61         comm.ExecuteNonQuery();
62     }
```

# Table Data Gateway. Пример без Data Sets

```
64     public long Insert(string lastName, string firstName, int numberOfDependents)
65     {
66         string sql =
67             @"INSERT INTO person VALUES ($id, $lastname, $firstname, $numberOfDependents)";
68         long key = GetNextId();
69         using var conn = DbManager.CreateConnection();
70         conn.Open();
71         using IDbCommand comm = new SQLiteCommand(sql, conn);
72         comm.Parameters.Add(new SQLiteParameter("$id", key));
73         comm.Parameters.Add(new SQLiteParameter("$lastname", lastName));
74         comm.Parameters.Add(new SQLiteParameter("$firstname", firstName));
75         comm.Parameters.Add(new SQLiteParameter("$numberOfDependents", numberOfDependents));
76         comm.ExecuteNonQuery();
77         return key;
78     }
```

# Table Data Gateway. Пример без Data Sets

```
80         public void Delete(long key)
81     {
82         string sql = "DELETE FROM person WHERE id = $id";
83         using var conn = DbManager.CreateConnection();
84         conn.Open();
85         IDbCommand comm = new SQLiteCommand(sql, conn);
86         comm.Parameters.Add(new SQLiteParameter("$id", key));
87         comm.ExecuteNonQuery();
88     }
```

# Table Data Gateway. Пример без Data Sets

```
90         private long GetNextId()
91     {
92         string sql = "SELECT max(id) as curId from person";
93         using var conn = DbManager.CreateConnection();
94         conn.Open();
95         using IDbCommand comm = new SQLiteCommand(sql, conn);
96         using IDataReader reader = comm.ExecuteReader();
97         bool hasResult = reader.Read();
98         if (hasResult)
99         {
100             return ((long)reader["curId"] + 1);
101         }
102         else
103         {
104             return 1;
105         }
106     }
```

# Table Data Gateway. Пример с Data Sets

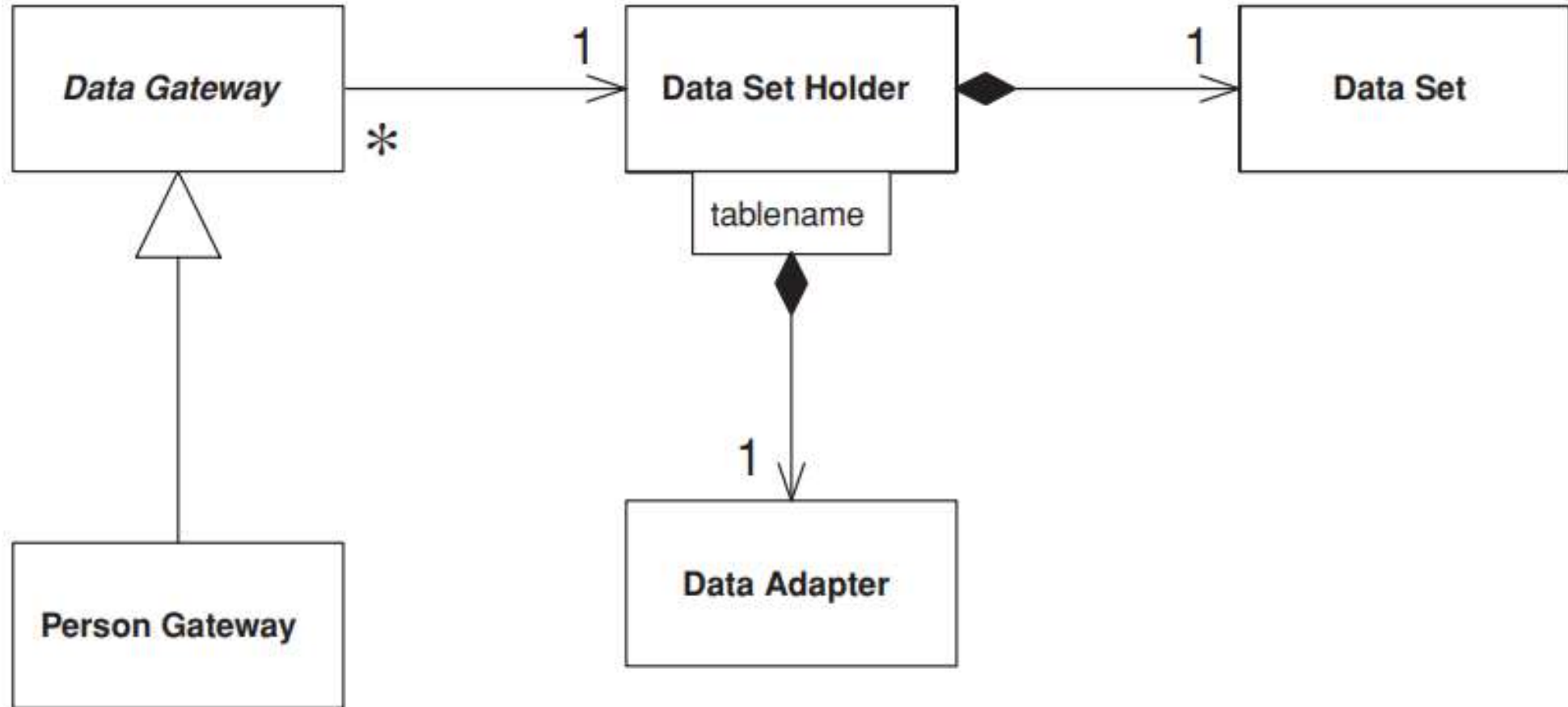


Figure 10.1 Class diagram of data-set-oriented gateway and the supporting data holder.

# Table Data Gateway. Пример с Data Sets

```
1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Data.SQLite;
5
6  namespace PoEAA_TableDataGateway.DataTableVersion
7  {
8      class DataSetHolder
9      {
10         public DataSet Data = new DataSet();
11         private readonly Dictionary<string, SQLiteDataAdapter> _dataAdapters = new Dictionary<string, SQLiteDataAdapter>();
```

# Table Data Gateway. Пример с Data Sets

```
13      public void FillData(string query, string tableName)
14      {
15          if (!_dataAdapters.ContainsKey(tableName))
16          {
17              throw new MultipleLoadException();
18          }
19
20          var conn = DbManager.CreateConnection();
21          conn.Open();
22          SQLiteDataAdapter da = new SQLiteDataAdapter(query, conn);
23          SQLiteCommandBuilder builder = new SQLiteCommandBuilder(da);
24          da.Fill(Data, tableName);
25          _dataAdapters.Add(tableName, da);
26      }
```



# Table Data Gateway. Пример с Data Sets

```
28         public void Update()
29         {
30             foreach (string table in _dataAdapters.Keys)
31             {
32                 (_dataAdapters[table]).Update(Data, table);
33             }
34         }
35
36         public DataTable this[string tableName] => Data.Tables[tableName];
37     }
38
39     public class MultipleLoadException : Exception
40     {
41     }
42 }
```

# Table Data Gateway. Пример с Data Sets

```
1  using System.Data;
2
3  namespace PoEAA_TableDataGateway.DataTableVersion
4  {
5      abstract class DataGateway
6      {
7          public abstract string TableName { get; }
8          public DataSetHolder Holder;
9
10         public DataSet Data => Holder.Data;
11
12         public abstract DataTable Table { get; }
```

# Table Data Gateway. Пример с Data Sets

```
14         protected DataGateway()  
15     {  
16         Holder = new DataSetHolder();  
17     }  
18  
19     protected DataGateway(DataSetHolder holder)  
20     {  
21         this.Holder = holder;  
22     }
```

# Table Data Gateway. Пример с Data Sets

```
24     public void LoadAll()
25     {
26         string commandString = $"select * from {TableName}";
27         Holder.FillData(commandString, TableName);
28     }
29
30     public void LoadWhere(string whereClause)
31     {
32         string commandString = $"select * from {TableName} where {whereClause}";
33         Holder.FillData(commandString, TableName);
34     }
```

# Table Data Gateway. Пример с Data Sets

```
1  using System.Data;
2
3  namespace PoEAA_TableDataGateway.DataTableVersion
4  {
5      class PersonGateway : DataGateway
6      {
7          public override string TableName => "person";
8
9          public override DataTable Table => Data.Tables[TableName];
10
11         public PersonGateway() : base()
12         {
13
14         }
15
16         public PersonGateway(DataSetHolder holder) : base(holder)
17         {
18
19         }
```

# Table Data Gateway. Пример с Data Sets

```
21     public DataRow this[long key]
22     {
23         get
24         {
25             string filter = $"id = {key}";
26             return Table.Select(filter)[0];
27         }
28     }
29
30     public long Insert(string lastName, string firstName, int numberOfDependents)
31     {
32         long key = GetNextId();
33         DataRow newRow = Table.NewRow();
34         newRow["id"] = key;
35         newRow["lastname"] = lastName;
36         newRow["firstname"] = firstName;
37         newRow["numberOfDependents"] = numberOfDependents;
38         Table.Rows.Add(newRow);
39
40         return key;
41     }
```

# Table Data Gateway. Пример с Data Sets

```
43     private long GetNextId()
44     {
45         var result = Table.Compute("max([id])", string.Empty);
46         if (result != System.DBNull.Value)
47         {
48             return ((int)result + 1);
49         }
50         else
51         {
52             return 1;
53         }
54     }
```

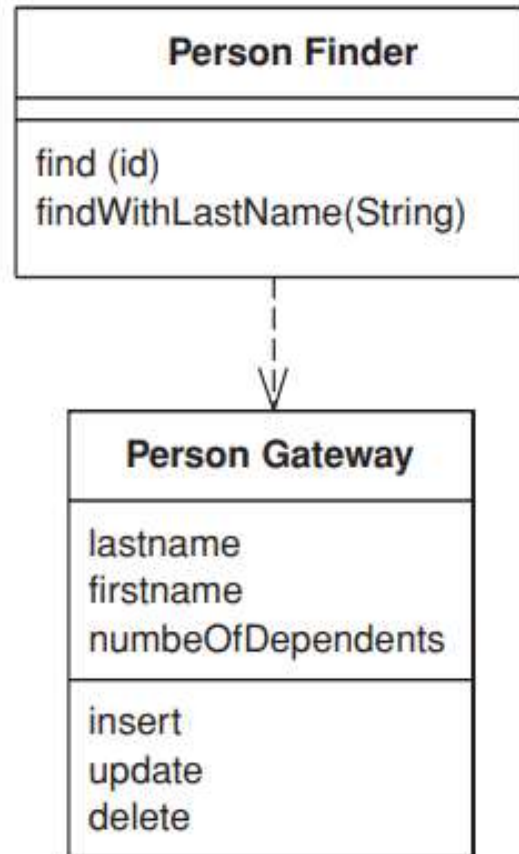
# Row Data Gateway

Класс, который является Gateway для одной строки в таблице базы данных. Один экземпляр работает со одной строкой в этой таблице



# Row Data Gateway

*An object that acts as a Gateway (466) to a single record in a data source. There is one instance per row.*



# Row Data Gateway. Как он работает?

Каждый столбец в таблице становится полем (свойством) соответствующего класса

Класс в том числе занимается конвертацией типов данных

Где должны располагаться методы, которые создадут экземпляры такого класса?

- Статические методы (плохо, не работает полиморфизм)
- Отдельные Finder классы

Неплохой выбор для автоматической генерации кода, вручную писать обычно сложно (много столбцов)

# Row Data Gateway. Как он работает?

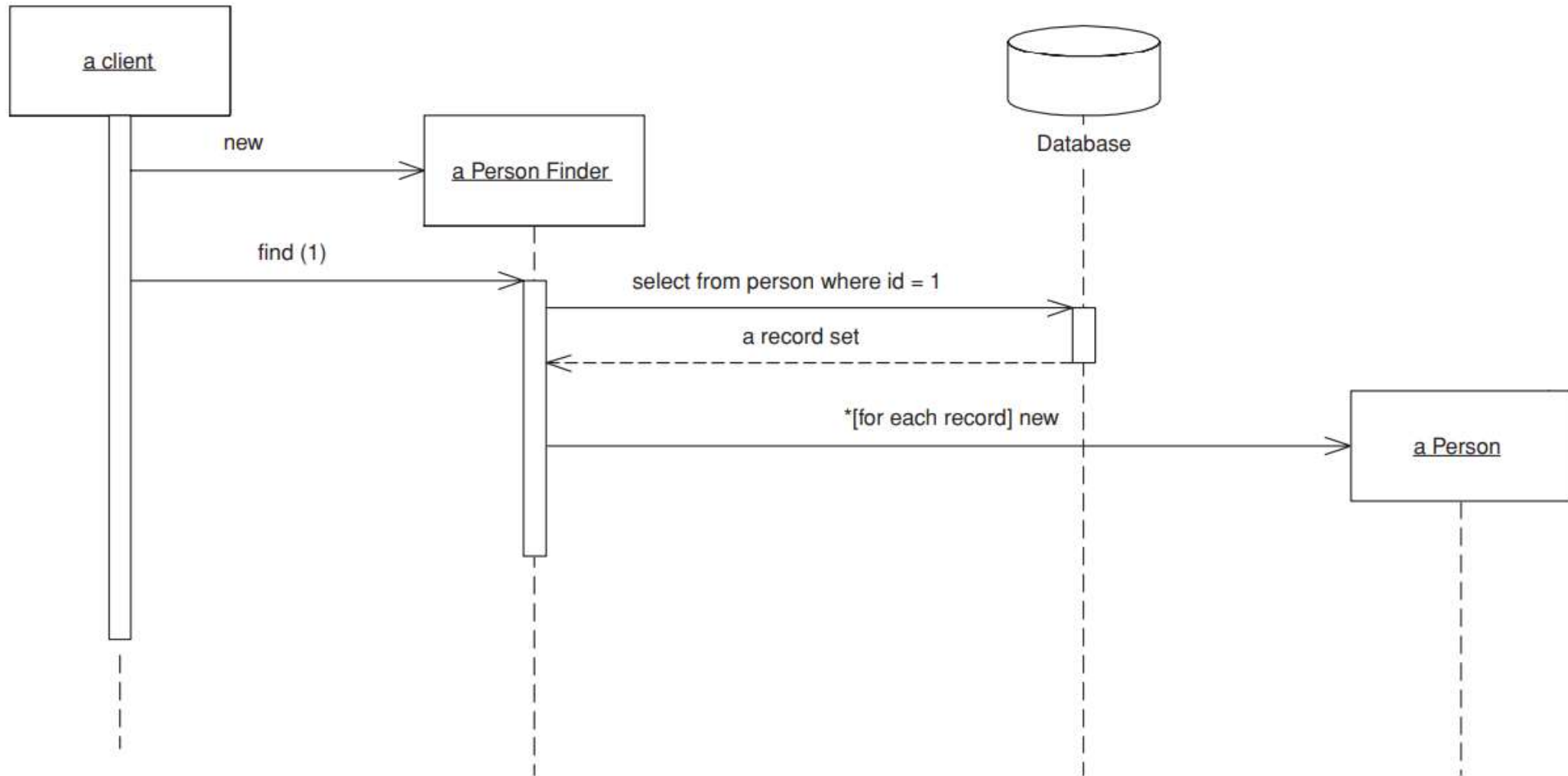


Figure 10.2 Interactions for a `find` with a row-based Row Data Gateway.

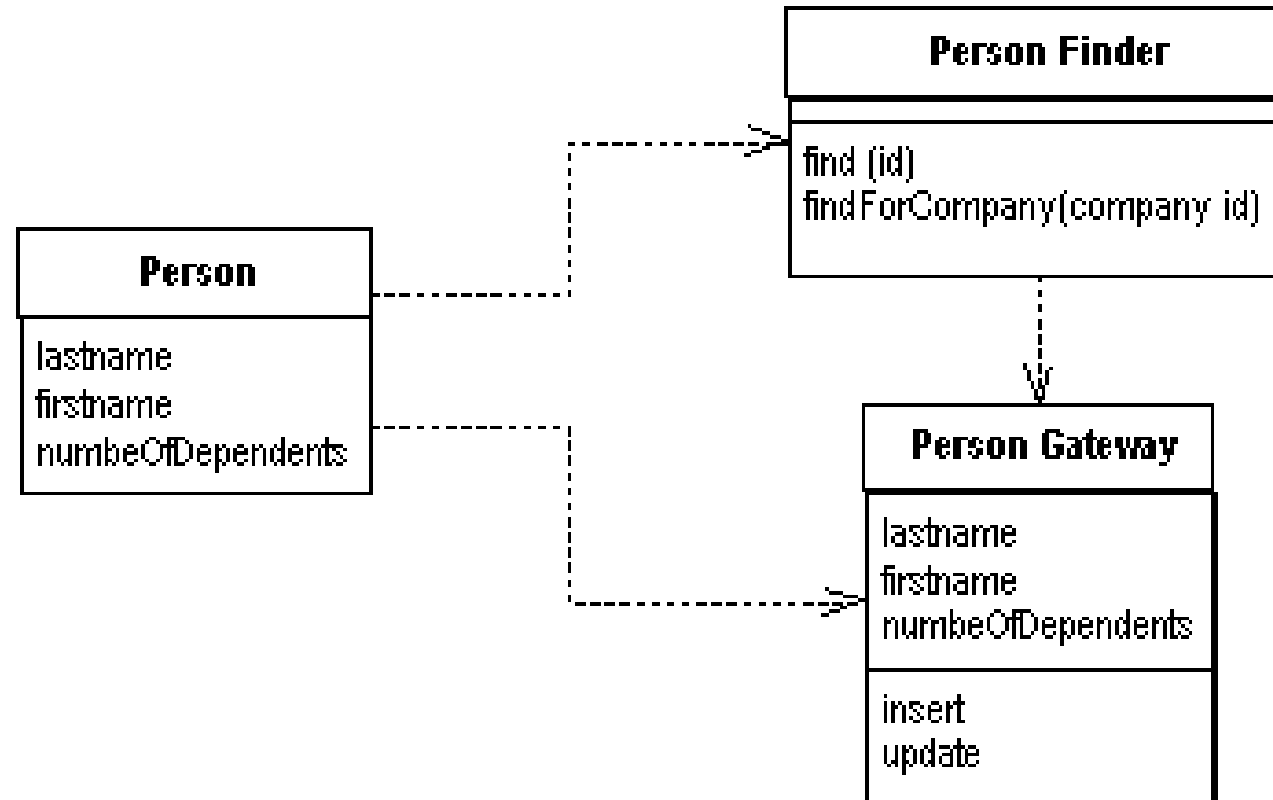
# Row Data Gateway. Когда использовать?

Обычно используется вместе с Transaction Script (рассмотрим позднее)

С Domain Model обычно не используется

Хоть это и сложнее, может быть использован вместе с Data Mapper

# Row Data Gateway. Пример



# Row Data Gateway. Пример

```
internal class Registry
{
    private static readonly Registry Instance = new Registry();
    private readonly Dictionary<int, PersonGateway> _personsMap = new Dictionary<int,
PersonGateway>();

    private Registry()
    {
    }

    public static void AddPerson(PersonGateway personGateway)
    {
        Instance._personsMap.Add(personGateway.Id, personGateway);
    }

    public static PersonGateway GetPerson(int id)
    {
        if (Instance._personsMap.ContainsKey(id))
        {
            return Instance._personsMap[id];
        }

        return null;
    }
}
```

# Row Data Gateway. Пример

```
internal class BaseGateway
{
    public int Id { get; set; }

    public BaseGateway()
    {
    }
}
```

# Row Data Gateway. Пример

```
class PersonGateway : BaseGateway
{
    public PersonGateway(int id, string lastName, string firstName, int numberOfDependents)
    {
        Id = id;
        LastName = lastName;
        FirstName = firstName;
        NumberOfDependents = numberOfDependents;
    }

    private const string UpdateStatementString =
        @"UPDATE person
           SET lastname = $lastname, firstname = $firstname,
numberOfDependents = $numberOfDependents
       where id = $id";

    private const string InsertStatementString =
        @"INSERT INTO person
           VALUES ($id, $lastname, $firstname, $numberOfDependents)";

    public string LastName { get; set; }
    public string FirstName { get; set; }
    public int NumberOfDependents { get; set; }
```



# Row Data Gateway. Пример

```
public static PersonGateway Load(IDataReader reader)
{
    object[] resultSet = new object[reader.FieldCount];
    reader.GetValues(resultSet);

    int id = (int) resultSet[0];
    PersonGateway result = Registry.GetPerson(id);
    if (result != null)
    {
        return result;
    }

    string lastName = resultSet[1].ToString();
    string firstName = resultSet[2].ToString();
    int numberOfDependents = (int)resultSet[3];
    result = new PersonGateway(id, lastName, firstName, numberOfDependents);
    Registry.AddPerson(result);
    return result;
}
```

# Row Data Gateway. Пример

```
public void Update()
{
    try
    {
        using var conn = DbManager.CreateConnection();
        conn.Open();
        using IDbCommand comm = new SQLiteCommand(UpdateStatementString, conn);
        comm.Parameters.Add(new SQLiteParameter("$lastname", LastName));
        comm.Parameters.Add(new SQLiteParameter("$firstname", FirstName));
        comm.Parameters.Add(new SQLiteParameter("$numberOfDependents",
NumberOfDependents));
        comm.Parameters.Add(new SQLiteParameter("$id", Id));
        comm.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        throw new ApplicationException(ex.Message);
    }
}
```

# Row Data Gateway. Пример

```
public int Insert()
{
    try
    {
        using var conn = DbManager.CreateConnection();
        conn.Open();
        using IDbCommand comm = new SQLiteCommand(InsertStatementString, conn);
        Id = FindNextDatabaseId();
        comm.Parameters.Add(new SQLiteParameter("$id", Id));
        comm.Parameters.Add(new SQLiteParameter("$lastname", LastName));
        comm.Parameters.Add(new SQLiteParameter("$firstname", FirstName));
        comm.Parameters.Add(new SQLiteParameter("$numberOfDependents",
NumberOfDependents));

        comm.ExecuteNonQuery();
        Registry.AddPerson(this);

        return Id;
    }
    catch (Exception ex)
    {
        throw new ApplicationException(ex.Message);
    }
}
```

# Row Data Gateway. Пример

```
private int FindNextDatabaseId()
{
    string sql = "SELECT max(id) as curId from person";
    using var conn = DbManager.CreateConnection();
    conn.Open();
    using IDbCommand comm = new SQLiteCommand(sql, conn);
    using IDataReader reader = comm.ExecuteReader();
    bool hasResult = reader.Read();
    if (hasResult)
    {
        return ((int)((long)reader["curId"] + 1));
    }
    else
    {
        return 1;
    }
}
```

# Row Data Gateway. Пример

```
class PersonFinder
{
    private const string FindStatementString = @"
        SELECT id, lastname, firstname, numberOfDependents
        from person
        WHERE id = $id";

    private const string FindResponsibleStatementString = @"
        SELECT id, lastname, firstname, numberOfDependents
        from person
        WHERE numberOfDependents > 0";
```

# Row Data Gateway. Пример

```
public PersonGateway Find(int id)
{
    PersonGateway result = Registry.GetPerson(id);
    if (result != null)
    {
        return result;
    }

    try
    {
        using var conn = DbManager.CreateConnection();
        conn.Open();
        using IDbCommand comm = new SQLiteCommand(FindStatementString, conn);
        comm.Parameters.Add(new SQLiteParameter("$id", id));
        using IDataReader reader = comm.ExecuteReader();
        reader.Read();
        result = PersonGateway.Load(reader);
        return result;
    }
    catch (Exception ex)
    {
        throw new ApplicationException(ex.Message);
    }
}
```

# Row Data Gateway. Пример

```
public List<PersonGateway> FindResponsibles()
{
    List<PersonGateway> result = new List<PersonGateway>();
    try
    {
        using var conn = DbManager.CreateConnection();
        conn.Open();
        using IDbCommand comm = new SQLiteCommand(FindResponsibleStatementString,
conn);

        using IDataReader reader = comm.ExecuteReader();
        while (reader.Read())
        {
            result.Add(PersonGateway.Load(reader));
        }

        return result;
    }
    catch (Exception ex)
    {
        throw new ApplicationException(ex.Message);
    }
}
```

# Active Record

Класс, который оборачивает строку в таблице базы данных, инкапсулирует доступ к базе данных и содержит бизнес-логику для этих данных

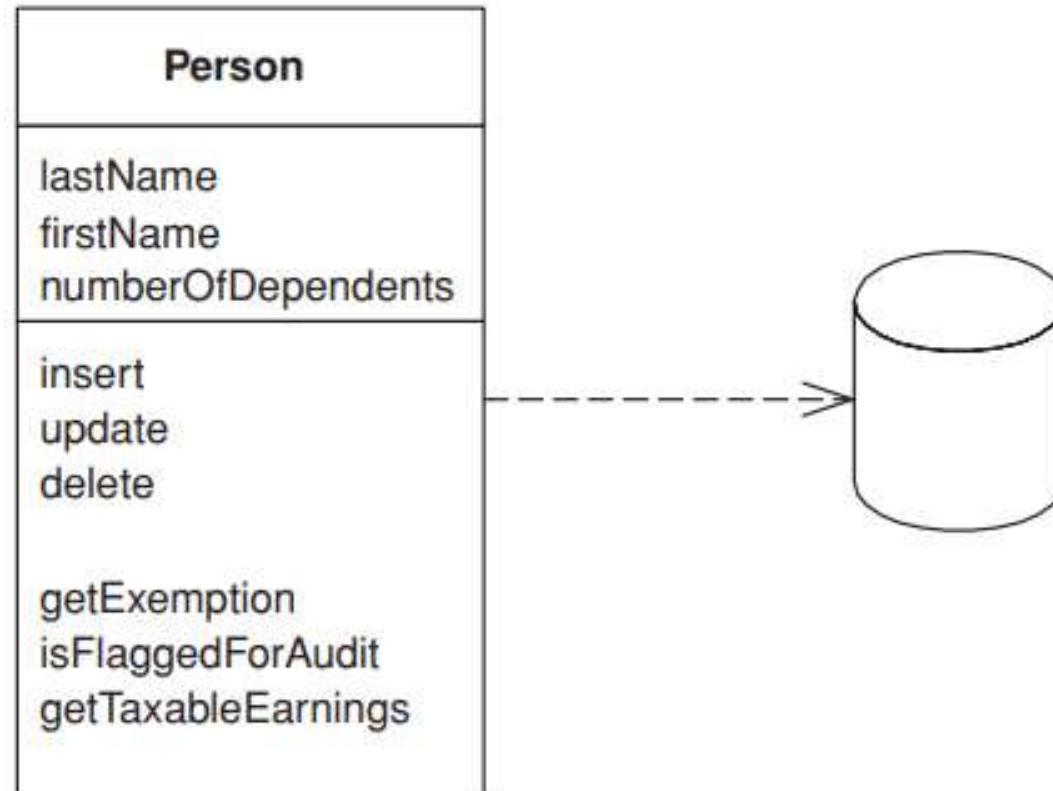
То есть такой объект обладает как данными, так и своим поведением

Очень похож на Row Data Gateway, разница в том, что Active Record содержит код бизнес-логики



# Active Record

*An object that wraps a row in a database table or view, encapsulates the database access, and adds domain logic on that data.*



# Active Record. Как он работает?

Каждый класс Active Record ответственен за сохранение и загрузку из базы данных, а также за бизнес-логику для этих (и только этих) данных

Набор данных строго соответствует базе данных – одно поле (свойство) в классе соответствует одному столбцу в таблице

# Active Record. Как он работает?

Обычно обладает следующими видами методов:

- Конструктор из строки-результата SQL запроса
- Конструктор для последующего добавления в таблицу
- Статические методы поиска, которые возвращают экземпляры Active Record
- Update и Insert методы
- Доступ к полям (свойства)
- Методы бизнес-логики

# Active Record. Как он работает?

Обычно обладает следующими видами методов:

- Конструктор из строки-результата SQL запроса
- Конструктор для последующего добавления в таблицу
- Статические методы поиска, которые возвращают экземпляры Active Record (можно выделить в отдельный класс)
- Update и Insert методы
- Доступ к полям (свойства)
- Методы бизнес-логики

# Active Record. Когда использовать?

Хороший выбор для не слишком сложной бизнес-логики

Неплохо работает с Transaction Script, также с простым Domain Model

# Active Record. Пример

```
internal class Registry
{
    private static readonly Registry Instance = new Registry();
    private readonly Dictionary<int, Person> _personsMap = new Dictionary<int, Person>();

    private Registry()
    {
    }

    public static void AddPerson(Person person)
    {
        Instance._personsMap.Add(person.Id, person);
    }

    public static Person GetPerson(int id)
    {
        if (Instance._personsMap.ContainsKey(id))
        {
            return Instance._personsMap[id];
        }

        return null;
    }
}
```

# Active Record. Пример

```
public static class DbManager
{
    public static SqlConnection CreateConnection()
    {
        return new SqlConnection("Data Source=poeaa_activerecord.db");
    }
}
```

```
internal class BaseActiveRecord
{
    public int Id { get; protected set; }
}
```

# Active Record. Пример

---

```
1  using System;
2  using System.Data;
3  using System.Data.SQLite;
4  using CodeParadise.Money;
5
6  namespace PoEAA_ActiveRecord
7  {
8      class Person : BaseActiveRecord
9      {
10         public string LastName { get; set; }
11         public string FirstName { get; set; }
12         public int NumberOfDependents { get; set; }
13         private const string FindStatementString = @"
14             SELECT id, lastname, firstname, numberOfDependents
15             FROM person
16             WHERE id = $id
17             ";
```



# Active Record. Пример

```
19     private const string UpdateStatementString =  
20         @"UPDATE person  
21             SET lastname = $lastname, firstname = $firstname, numberOfDependents = $numberOfDependents  
22             where id = $id";  
23  
24     private const string InsertStatementString =  
25         @"INSERT INTO person  
26             VALUES ($id, $lastname, $firstname, $numberOfDependents)";
```

# Active Record. Пример

```
28     public Person(int id, string lastName, string firstName, int numberOfDependents)
29     {
30         Id = id;
31         LastName = lastName;
32         FirstName = firstName;
33         NumberOfDependents = numberOfDependents;
34     }
```

# Active Record. Пример

```
36     public static Person Find(int id)
37     {
38         Person result = Registry.GetPerson(id);
39         if (result != null)
40         {
41             return result;
42         }
43
44         try
45         {
46             using var conn = DbManager.CreateConnection();
47             conn.Open();
48             using IDbCommand comm = new SQLiteCommand(FindStatementString, conn);
49             comm.Parameters.Add(new SQLiteParameter("$id", id));
50             using IDataReader reader = comm.ExecuteReader();
51             reader.Read();
52             result = Load(reader);
53             return result;
54         }
55         catch (Exception ex)
56         {
57             throw new ApplicationException(ex.Message);
58         }
59     }
```

# Active Record. Пример

```
61         public static Person Load(IDataReader reader)
62         {
63             object[] resultSet = new object[reader.FieldCount];
64             reader.GetValues(resultSet);
65
66             int id = (int)resultSet[0];
67             Person result = Registry.GetPerson(id);
68             if (result != null)
69             {
70                 return result;
71             }
72
73             string lastName = resultSet[1].ToString();
74             string firstName = resultSet[2].ToString();
75             int numberOfDependents = (int)resultSet[3];
76             result = new Person(id, lastName, firstName, numberOfDependents);
77             Registry.AddPerson(result);
78             return result;
79         }
```

# Active Record. Пример

```
81     public void Update()
82     {
83         try
84         {
85             using var conn = DbManager.CreateConnection();
86             conn.Open();
87             using IDbCommand comm = new SQLiteCommand(UpdateStatementString, conn);
88             comm.Parameters.Add(new SQLiteParameter("$lastname", LastName));
89             comm.Parameters.Add(new SQLiteParameter("$firstname", FirstName));
90             comm.Parameters.Add(new SQLiteParameter("$numberOfDependents", NumberOfDependents));
91             comm.Parameters.Add(new SQLiteParameter("$id", Id));
92             comm.ExecuteNonQuery();
93         }
94         catch (Exception ex)
95         {
96             throw new ApplicationException(ex.Message);
97         }
98     }
```

# Active Record. Пример

```
100         public int Insert()
101         {
102             try
103             {
104                 using var conn = DbManager.CreateConnection();
105                 conn.Open();
106                 using IDbCommand comm = new SQLiteCommand(InsertStatementString, conn);
107                 Id = FindNextDatabaseId();
108                 comm.Parameters.Add(new SQLiteParameter("$id", Id));
109                 comm.Parameters.Add(new SQLiteParameter("$lastname", LastName));
110                 comm.Parameters.Add(new SQLiteParameter("$firstname", FirstName));
111                 comm.Parameters.Add(new SQLiteParameter("$numberOfDependents", NumberOfDependents));
112                 comm.ExecuteNonQuery();
113                 Registry.AddPerson(this);
114
115                 return Id;
116             }
117             catch (Exception ex)
118             {
119                 throw new ApplicationException(ex.Message);
120             }
121         }
```

# Active Record. Пример

```
123     public Money GetExemption()  
124     {  
125         Money baseExemption = Money.Dollars(1500d);  
126         Money dependentExemption = Money.Dollars(750d);  
127         return baseExemption.Add(dependentExemption.Multiply((double) NumberOfDependents));  
128     }
```

# Active Record. Пример

```
130         private int FindNextDatabaseId()
131         {
132             string sql = "SELECT max(id) as curId from person";
133             using var conn = DbManager.CreateConnection();
134             conn.Open();
135             using IDbCommand comm = new SQLiteCommand(sql, conn);
136             using IDataReader reader = comm.ExecuteReader();
137             bool hasResult = reader.Read();
138             if (hasResult)
139             {
140                 return ((int)((long)reader["curId"] + 1));
141             }
142             else
143             {
144                 return 1;
145             }
146         }
```

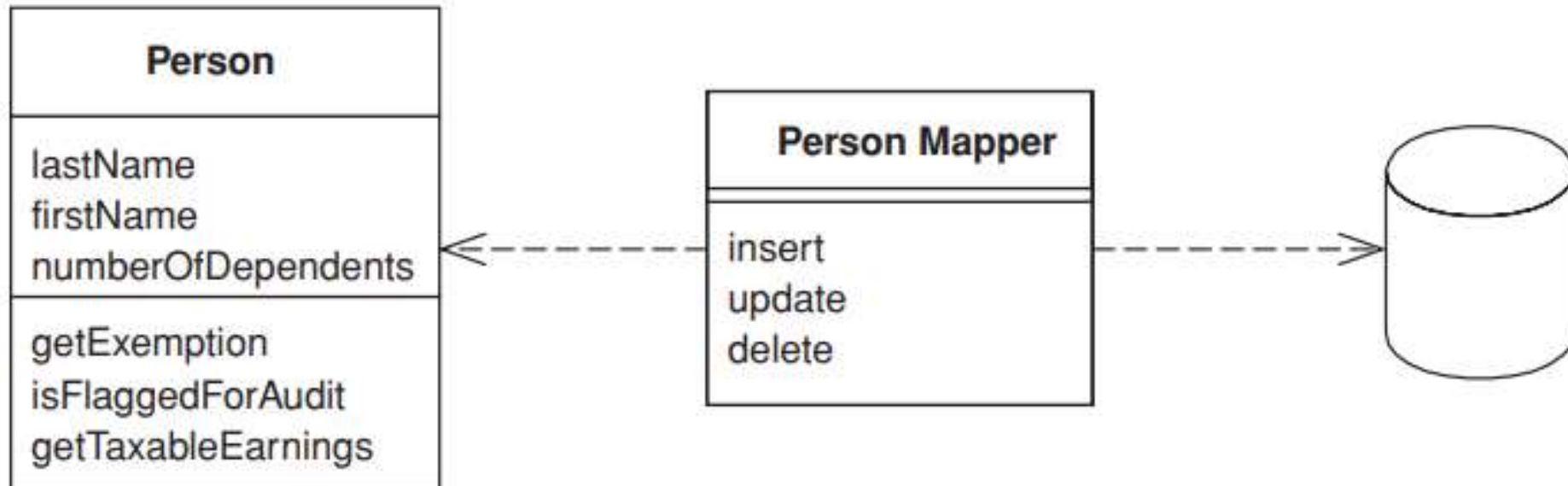


# Data Mapper

Набор классов Mapper, которые перемещают данные между объектами приложения и базой данных, не создавая зависимости между ними

# Data Mapper

*A layer of Mappers (473) that moves data between objects and a database while keeping them independent of each other and the mapper itself.*



# Data Mapper. Как он работает?

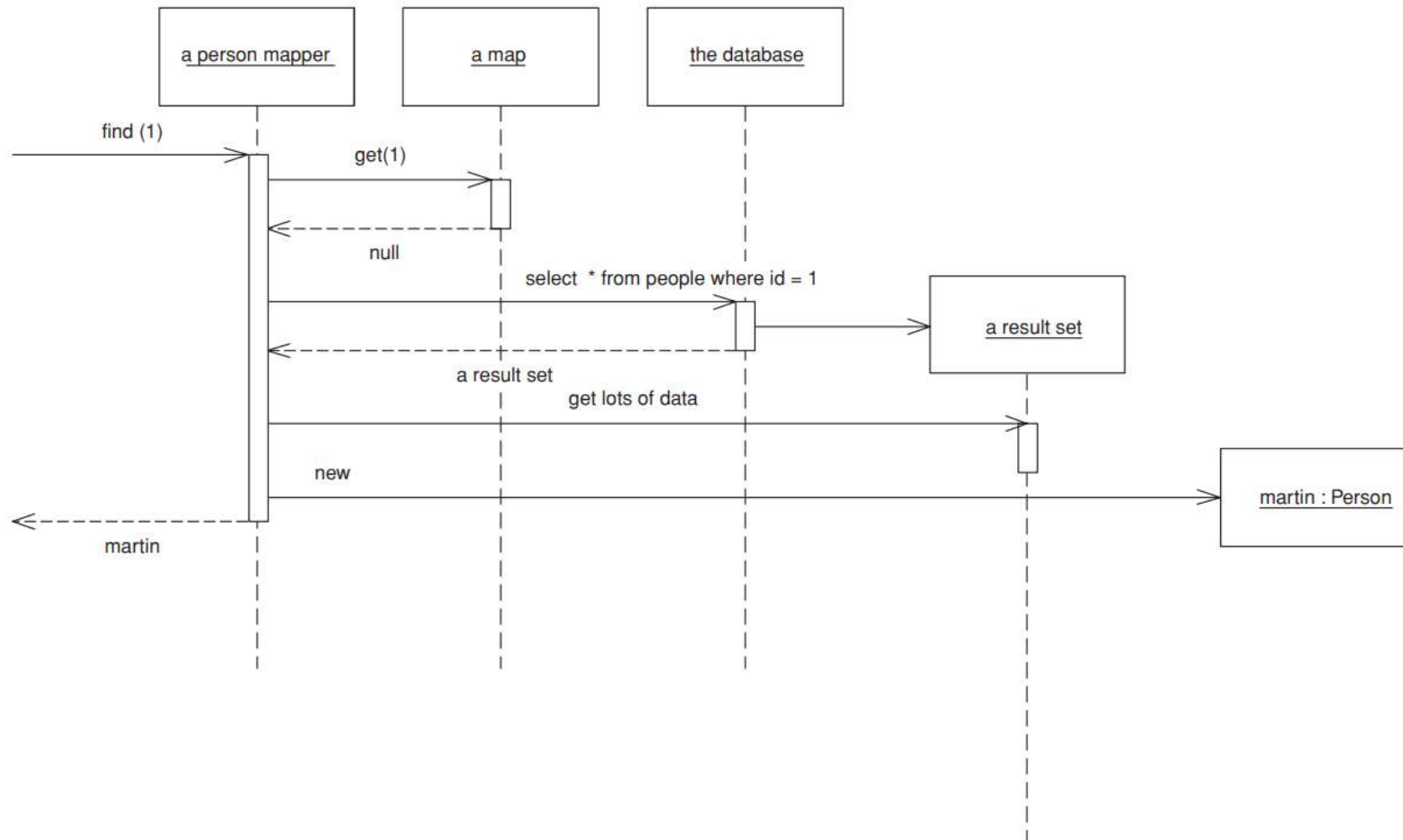


Figure 10.3 Retrieving data from a database.

# Data Mapper. Как он работает?

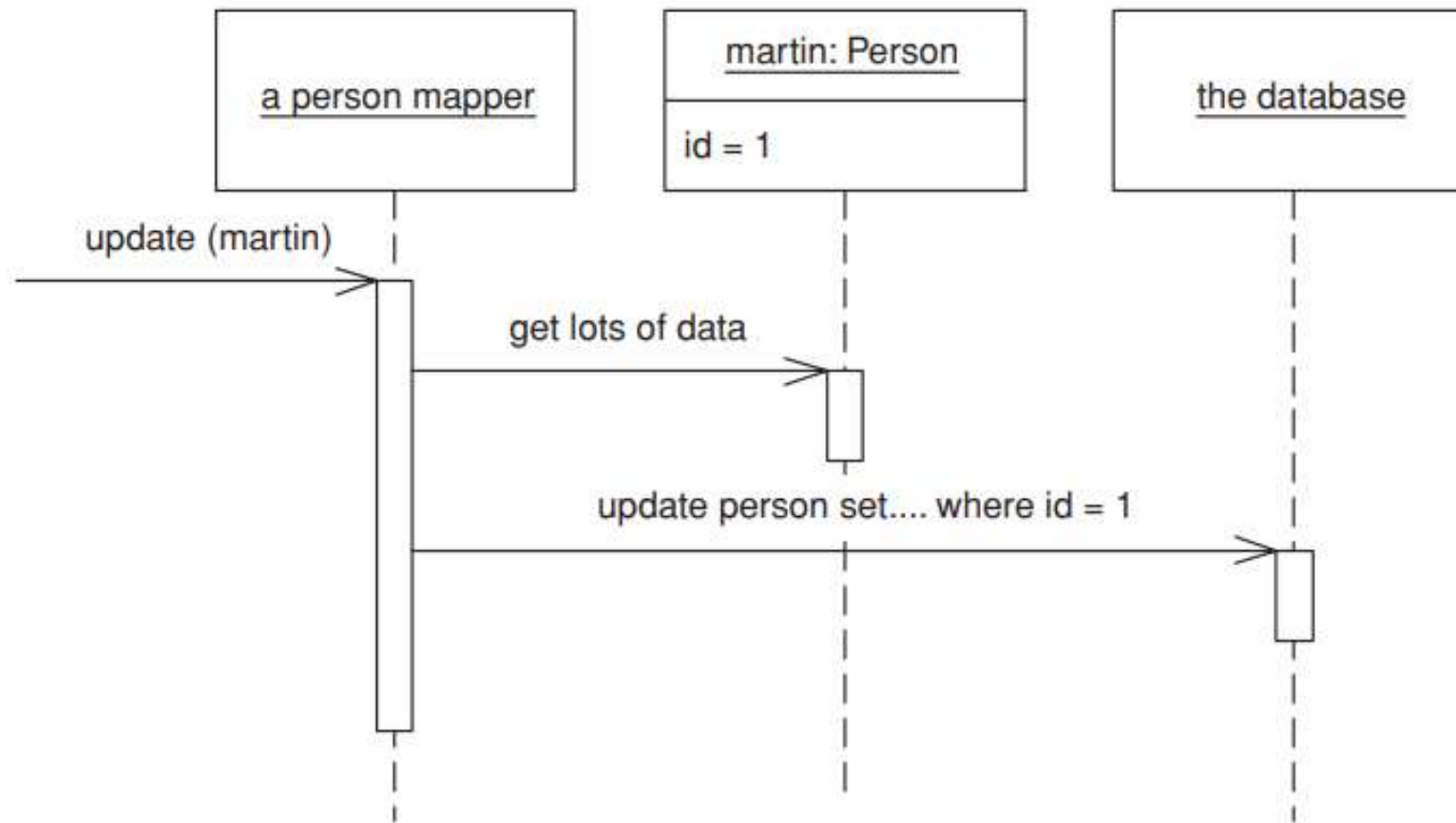


Figure 10.4 *Updating data.*

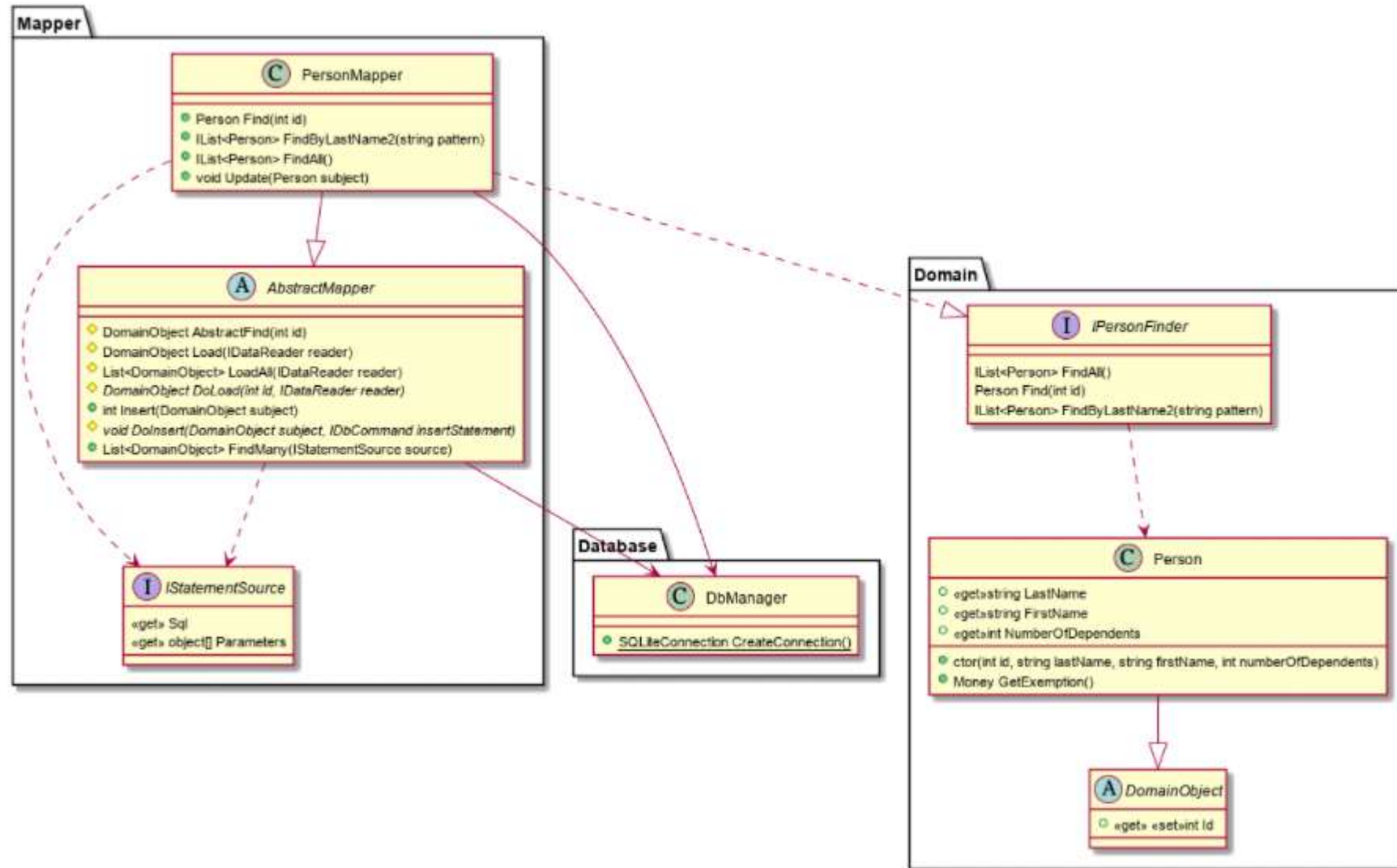
# Data Mapper. Когда использовать?

Если вы хотите, чтобы схема (структура) базы данных и ваша объектная модель не зависели друг от друга и развивались параллельно

Чаще всего такое происходит при использовании Domain Model

При этом использование оправдано, если бизнес-логика достаточно сложна, иначе можно обойтись Active Record

# Data Mapper. Пример



# Data Mapper. Пример

```
1  using System.Data.SQLite;
2
3  namespace PoEAA_DataMapper.Database
4  {
5      public static class DbManager
6      {
7          public static SQLiteConnection CreateConnection()
8          {
9              return new SQLiteConnection("Data Source=poeaa_datamapper.db");
10         }
11     }
12 }
```

# Data Mapper. Пример

```
1  using CodeParadise.Money;
2
3  namespace PoEAA_DataMapper.Domain
4  {
5      public class Person : DomainObject
6      {
7          public string LastName { get; set; }
8          public string FirstName { get; set; }
9          public int NumberOfDependents { get; set; }
10         public Person(int id, string lastName, string firstName, int numberOfDependents)
11         {
12             Id = id;
13             LastName = lastName;
14             FirstName = firstName;
15             NumberOfDependents = numberOfDependents;
16         }
17
18         public Money GetExemption()
19         {
20             Money baseExemption = Money.Dollars(1500d);
21             Money dependentExemption = Money.Dollars(750d);
22             return baseExemption.Add(dependentExemption.Multiply((double)NumberOfDependents));
23         }
24     }
25 }
```



# Data Mapper. Пример

```
1  using System.Collections.Generic;
2
3  namespace PoEAA_DataMapper.Domain
4  {
5      public interface IPersonFinder
6      {
7          IList<Person> FinAll();
8          Person Find(int id);
9          IList<Person> FindByLastName2(string pattern);
10     }
11 }
```

# Data Mapper. Пример

```
1  namespace PoEAA_DataMapper.Mapper
2  {
3      public interface IStatementSource
4      {
5          string Sql { get; }
6          object[] Parameters { get; }
7      }
8  }
```

# Data Mapper. Пример

```
1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Data.SQLite;
5  using PoEAA_DataMapper.Database;
6  using PoEAA_DataMapper.Domain;
7
8  namespace PoEAA_DataMapper.Mapper
9  {
10     public abstract class AbstractMapper
11     {
12         protected Dictionary<int, DomainObject> LoadedMap = new Dictionary<int, DomainObject>();
13         protected abstract string FindStatement();
14         protected abstract string InsertStatement();
15
16         protected abstract int FindNextDatabaseId();
```

# Data Mapper. Пример

```
18         protected DomainObject AbstractFind(int id)
19         {
20             bool findResult = LoadedMap.TryGetValue(id, out DomainObject result);
21             if (findResult)
22             {
23                 return result;
24             }
25
26             try
27             {
28                 using var conn = DbManager.CreateConnection();
29                 conn.Open();
30                 using IDbCommand comm = new SQLiteCommand(FindStatement(), conn);
31                 comm.Parameters.Add(new SQLiteParameter("$id", id));
32                 using IDataReader reader = comm.ExecuteReader();
33                 reader.Read();
34                 result = Load(reader);
35                 return result;
36             }
37             catch (Exception ex)
38             {
39                 throw new ApplicationException(ex.Message);
40             }
41         }
```

# Data Mapper. Пример

```
70         protected abstract DomainObject DoLoad(int id, IDataReader reader);

43     protected DomainObject Load(IDataReader reader)
44     {
45         object[] resultSet = new object[reader.FieldCount];
46         reader.GetValues(resultSet);
47
48         int id = (int)resultSet[0];
49         if (LoadedMap.ContainsKey(id))
50         {
51             return LoadedMap[id];
52         }
53
54         DomainObject result = DoLoad(id, reader);
55         LoadedMap.Add(id, result);
56         return result;
57     }
```

# Data Mapper. Пример

```
59         protected List<DomainObject> LoadAll(IDataReader reader)
60     {
61         List<DomainObject> result = new List<DomainObject>();
62         while (reader.Read())
63         {
64             result.Add(Load(reader));
65         }
66
67         return result;
68     }
```

# Data Mapper. Пример

```
96         protected abstract void DoInsert(DomainObject subject, IDbCommand insertStatement);  
  
72     public int Insert(DomainObject subject)  
73     {  
74         try  
75         {  
76             using var conn = DbManager.CreateConnection();  
77             conn.Open();  
78             using IDbCommand comm = new SQLiteCommand(insertStatement(), conn);  
79             subject.Id = FindNextDatabaseId();  
80             var parameter = comm.CreateParameter();  
81             parameter.DbType = DbType.Int32;  
82             parameter.Value = subject.Id;  
83             comm.Parameters.Add(parameter);  
84             DoInsert(subject, comm);  
85             comm.ExecuteNonQuery();  
86             LoadedMap.Add(subject.Id, subject);  
87             return subject.Id;  
88         }  
89         catch (Exception ex)  
90         {  
91             throw new ApplicationException(ex.Message);  
92         }  
93     }  
94 }
```

# Data Mapper. Пример

```
98         public List<DomainObject> FindMany(IStatementSource source)
99         {
100             try
101             {
102                 using var conn = DbManager.CreateConnection();
103                 conn.Open();
104                 using IDbCommand comm = new SQLiteCommand(source.Sql, conn);
105                 foreach (var p in source.Parameters)
106                 {
107                     var parameter = comm.CreateParameter();
108                     parameter.DbType = DbType.Object;
109                     parameter.Value = p;
110                     comm.Parameters.Add(parameter);
111                 }
112                 using IDataReader reader = comm.ExecuteReader();
113                 return LoadAll(reader);
114             }
115             catch (Exception ex)
116             {
117                 throw new ApplicationException(ex.Message);
118             }
119         }
120
121     }
122 }
```



# Data Mapper. Пример

---

```
1  using PoEAA_DataMapper.Database;
2  using PoEAA_DataMapper.Domain;
3  using System;
4  using System.Collections.Generic;
5  using System.Data;
6  using System.Data.SQLite;
7  using System.Linq;
8
9  namespace PoEAA_DataMapper.Mapper
10 {
11     public class PersonMapper : AbstractMapper, IPersonFinder
12     {
13         private const string Columns = " id, lastname, firstname, numberOfDependents ";
14
15         private const string UpdateStatementString =
16             "UPDATE person SET lastname = ?, firstname = ?, numberOfDependents = ? WHERE id = ?";
```

# Data Mapper. Пример

```
17         protected override string FindStatement()  
18     {  
19         return "SELECT " + Columns + " FROM person WHERE id = $id";  
20     }  
21  
22     protected override string InsertStatement()  
23     {  
24         return "INSERT INTO person VALUES (?, ?, ?, ?)";  
25     }
```

# Data Mapper. Пример

```
27         protected override int FindNextDatabaseId()
28     {
29         string sql = "SELECT max(id) as curId from person";
30         using var conn = DbManager.CreateConnection();
31         conn.Open();
32         using IDbCommand comm = new SQLiteCommand(sql, conn);
33         using IDataReader reader = comm.ExecuteReader();
34         bool hasResult = reader.Read();
35         if (hasResult)
36         {
37             return ((int)((long)reader["curId"] + 1));
38         }
39         else
40         {
41             return 1;
42         }
43     }
```

# Data Mapper. Пример

```
45         protected override DomainObject DoLoad(int id, IDataReader reader)
46     {
47         object[] resultSet = new object[reader.FieldCount];
48         reader.GetValues(resultSet);
49         string lastName = resultSet[1].ToString();
50         string firstName = resultSet[2].ToString();
51         int numberOfDependents = (int)resultSet[3];
52         return new Person(id, lastName, firstName, numberOfDependents);
53     }
```

# Data Mapper. Пример

```
55         protected override void DoInsert(DomainObject subject, IDbCommand insertStatement)
56     {
57         Person person = (Person) subject;
58         var p1 = insertStatement.CreateParameter();
59         p1.DbType = DbType.String;
60         p1.Value = person.LastName;
61
62         var p2 = insertStatement.CreateParameter();
63         p2.DbType = DbType.String;
64         p2.Value = person.FirstName;
65
66         var p3 = insertStatement.CreateParameter();
67         p3.DbType = DbType.Int32;
68         p3.Value = person.NumberOfDependents;
69         insertStatement.Parameters.Add(p1);
70         insertStatement.Parameters.Add(p2);
71         insertStatement.Parameters.Add(p3);
72     }
```

# Data Mapper. Пример

```
74     public Person Find(int id)
75     {
76         return (Person) AbstractFind(id);
77     }
78
79     public IList<Person> FindByLastName2(string pattern)
80     {
81         return FindMany(new FindByLastName(pattern))
82             .Cast<Person>().ToList();
83     }
84
85     public IList<Person> FinAll()
86     {
87         return FindMany(new FindAllStatement())
88             .Cast<Person>().ToList();
89     }
```

# Data Mapper. Пример

```

91     public void Update(Person subject)
92     {
93         try
94         {
95             using var conn = DbManager.CreateConnection();
96             conn.Open();
97             using IDbCommand comm = new SQLiteCommand(UpdateStatementString, conn);
98             var p1 = comm.CreateParameter();
99             p1.DbType = DbType.String;
100            p1.Value = subject.LastName;
101
102            var p2 = comm.CreateParameter();
103            p2.DbType = DbType.String;
104            p2.Value = subject.FirstName;
105
106            var p3 = comm.CreateParameter();
107            p3.DbType = DbType.Int32;
108            p3.Value = subject.NumberOfDependents;
109
110            var p4 = comm.CreateParameter();
111            p4.DbType = DbType.Int32;
112            p4.Value = subject.Id;
113
114            comm.Parameters.Add(p1);
115            comm.Parameters.Add(p2);
116            comm.Parameters.Add(p3);
117            comm.Parameters.Add(p4);
118
119            comm.ExecuteNonQuery();
120        }
121        catch (Exception ex)
122        {
123            throw new ApplicationException(ex.Message);
124        }
125    }

```

# Data Mapper. Пример

```
127     private class FindByLastName : IStatementSource
128     {
129         private readonly string _lastName;
130
131         public string Sql { get; } =
132             "SELECT " + Columns + " FROM person WHERE UPPER(lastname) like UPPER(?) ORDER BY lastName";
133
134         public object[] Parameters
135         {
136             get
137             {
138                 return new object[] { _lastName };
139             }
140         }
141
142         public FindByLastName(string lastName)
143         {
144             _lastName = lastName;
145         }
146     }
```



# Data Mapper. Пример

```
148         private class FindAllStatement : IStatementSource
149         {
150             private readonly string _lastName;
151
152             public string Sql { get; } =
153                 "SELECT * FROM person";
154
155             public object[] Parameters
156             {
157                 get
158                 {
159                     return new object[] {};
160                 }
161             }
162         }
163     }
164 }
```

# Data Mapper. Пример

---

```
1  using System;
2  using System.Collections.Generic;
3  using PoEAA_DataMapper.Database;
4  using PoEAA_DataMapper.Domain;
5  using PoEAA_DataMapper.Mapper;
6
7  namespace PoEAA_DataMapper
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             InitializeData();
```

# Data Mapper. Пример

```
15         Console.WriteLine("Get persons");
16         PersonMapper mapper = new PersonMapper();
17         // get all persons
18         var people = mapper.FinAll();
19         PrintPerson(people);
20
21         Console.WriteLine("Insert a new person");
22         mapper.Insert(new Person(0, "Rose", "Jackson", 60));
23         people = mapper.FinAll();
24         PrintPerson(people);
25
26         Console.WriteLine("Update a person's first name");
27         var firstPerson = mapper.Find(1);
28         firstPerson.FirstName = "Jack";
29         mapper.Update(firstPerson);
```

# Data Mapper. Пример

```
31         Console.WriteLine("Update a person's number of dependents");
32         var secondPerson = mapper.Find(2);
33         secondPerson.NumberOfDependents = 0;
34         mapper.Update(secondPerson);
35
36
37         Console.WriteLine("Get persons again");
38         people = mapper.FinAll();
39         PrintPerson(people);
40
41         Console.WriteLine("Get persons with lastname containing n");
42         people = mapper.FindByLastName2("%n%");
43         PrintPerson(people);
44     }
```

Спасибо за внимание!