



Обзор технологий  
доступа к данным:  
LINQ и  
ADO.NET Entity Framework

# Содержание

- LINQ
- ADO.NET Entity Framework
- 
- 
- Общая картина подхода Microsoft к технологиям доступа к данным

# История

- OLE DB(1.0 – 1992) – низкоуровневый API
- OLE DB - COM API
- ADO – объектная модель - ***Recordset***
- ADO.NET – объектная модель .NET - ***Dataset***
  
- MCAD (Windows DAC) -1996
  - Microsoft Jet Database Engine
  - DB-Library
  - Data Access Objects (DAO)
  - Remote Data Objects (RDO)

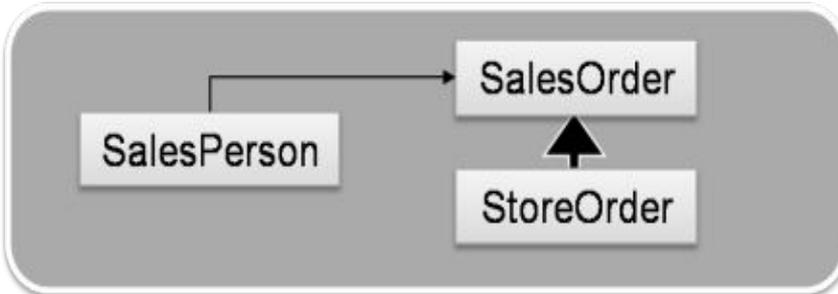
# Доступ к данным

```
DataSet DS=new DataSet();
XQueryNavigatorCollection oXQ = new XQueryNavigatorCollection();
string strXML = "";
string fileName1="c:\\Test\\T1.xml";
string alias1 = "MyDataTest.xml";
oXQ.AddNavigator( fileName1, alias1 );
string strQuery = "<NewDataSet> { " +
" let $bb := document(\"MyDataTest.xml\")/*/* " +
" let $cc := document(\"MyDatattt.xml\")/*/* " +
" for $c in $cc " +
" for $b in $bb " +
" where $c/kod = $b/kod " +
" return <Table> { $b/nazv,$b/dat,$c/naim } </Table> " +
" }</NewDataSet> ";
XQueryExpression xExpression = new XQueryExpression( strQuery );
strXML = xExpression.Execute();
StringReader strReader = new StringReader( strXML );
XmlTextReader reader = new XmlTextReader( strReader );
DS.ReadXml(reader);
DataGrid1.DataSource = DS.Tables[0];
DataGrid1.DataBind();
```

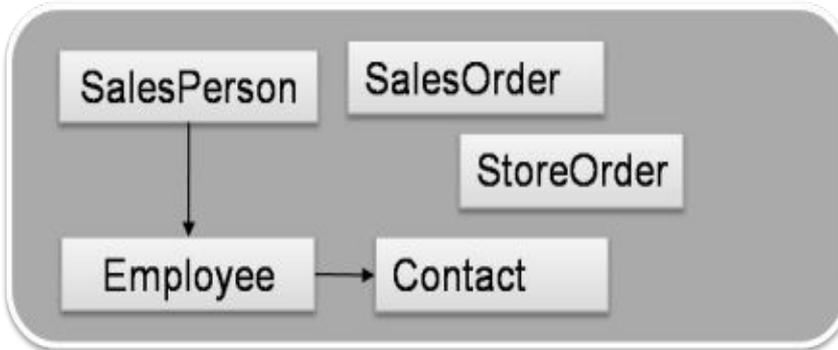
```
Dim xmlDoc As New System.Xml.XPath.XPathDocument("c:\books.xml")
Dim nav As System.Xml.XPath.XPathNavigator = xmlDoc.CreateNavigator()
Dim expr As System.Xml.XPath.XPathExpression = nav.Compile( "//Publisher[ = 'MSPress']/parent::node()/Title" )
```

# Типовые уровни при работе с данными

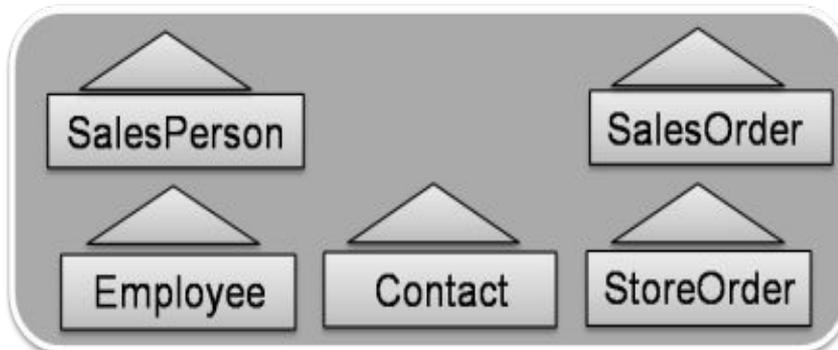
Conceptual



Logical



Physical



## ■ Концептуальный

- Бизнес сущности и логика
- Объекты и объектные коллекции

## ■ Логический

- Реляционное представление
- Таблицы, отношения
- junction tables

## ■ Физический

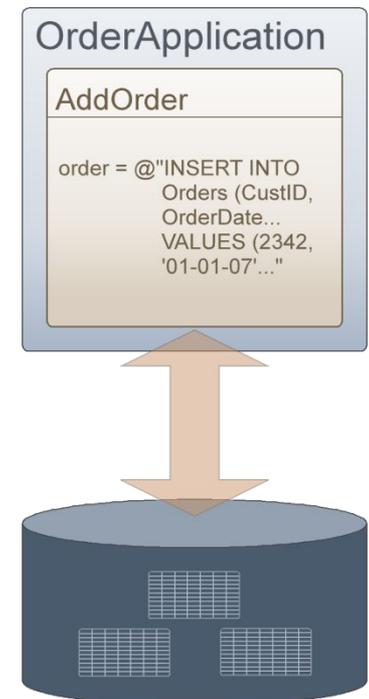
- физическое хранение и манипулирование
- страницы, экстенды, индексы ограничения (constraints) ...

# Сложности в работе с данными

Проблема:  
Data != Objects

## Нетипизированные острова

- разные для каждого источника
  - специфические знания для каждого источника
  - отсутствие проверок на этапе компиляции
  - сложность в поддержке ...
  - ... и даже нет IntelliSense 😊
- 
- Dataset – реляционное представление в объектном виде
  - Скрытые знания о структуре хранения и объектно-реляционном маппинге (ORM) в виде строк
  - Data Access Layer – спасение?



# Решение – основные идеи

- Типизированные запросы
- Типизированные результаты

***LINQ***

- Объектное представление схемы хранения
- Общее решение для целого ряда продуктов
- Использование преимуществ ADO.NET

- Создание концептуальной объектной модели
- Явное декларативное представление схемы объектно-реляционного маппинга (ORM) между концептуальной моделью и моделью хранения

***Entity Framework***

# Language Integrated Query

- **Единообразные** типизированные **запросы** к любому источнику данных
  - Методы расширения интерфейса IEnumerable (Select, OrderBy, GroupBy, Join. Where)
  - Возвращаемый результат как объектная коллекция (IEnumerable <T>)
- Intellisense – проверка на этапе компиляции
- Специальный синтаксис выражений запросов

# C# 3.0

```
var contacts =  
    from c in customers  
    where c.State == "WA"  
    select new { c.Name, c.Phone };
```

Query  
expressions

Local variable  
type inference

Lambda  
expressions

```
var contacts =  
    customers  
    .Where(c => c.State == "WA")  
    .Select(c => new { c.Name, c.Phone });
```

Extension  
methods

Anonymous  
types

Object  
initializers

# VBA 9

```
Dim contacts =  
  From c In customers _  
  Where c.State = "WA" _  
  Select c.Name, c.Phone
```

Query expressions

Local variable type inference

Lambda expressions

```
Dim contacts = _  
  customers _  
  .Where(Function (c) c.State = "WA")_  
  .Select(Function(c) New With { c.Name, c.Phone })
```

Extension methods

Anonymous types

Object initializers

# Проект LINQ

C# 3.0

VB 9.0

Другие

.NET Language Integrated Query

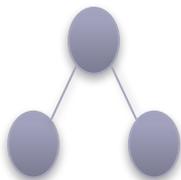
LINQ to  
Objects

LINQ to  
DataSets

LINQ to  
SQL

LINQ to  
Entities

LINQ to  
XML



Объекты



Реляционные  
источники

```
<book>  
<title/>  
<author/>  
<year/>  
<price/>  
</book>
```

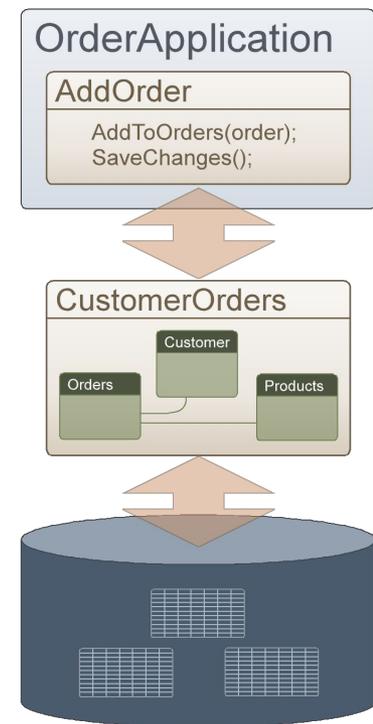
XML

# LINQ

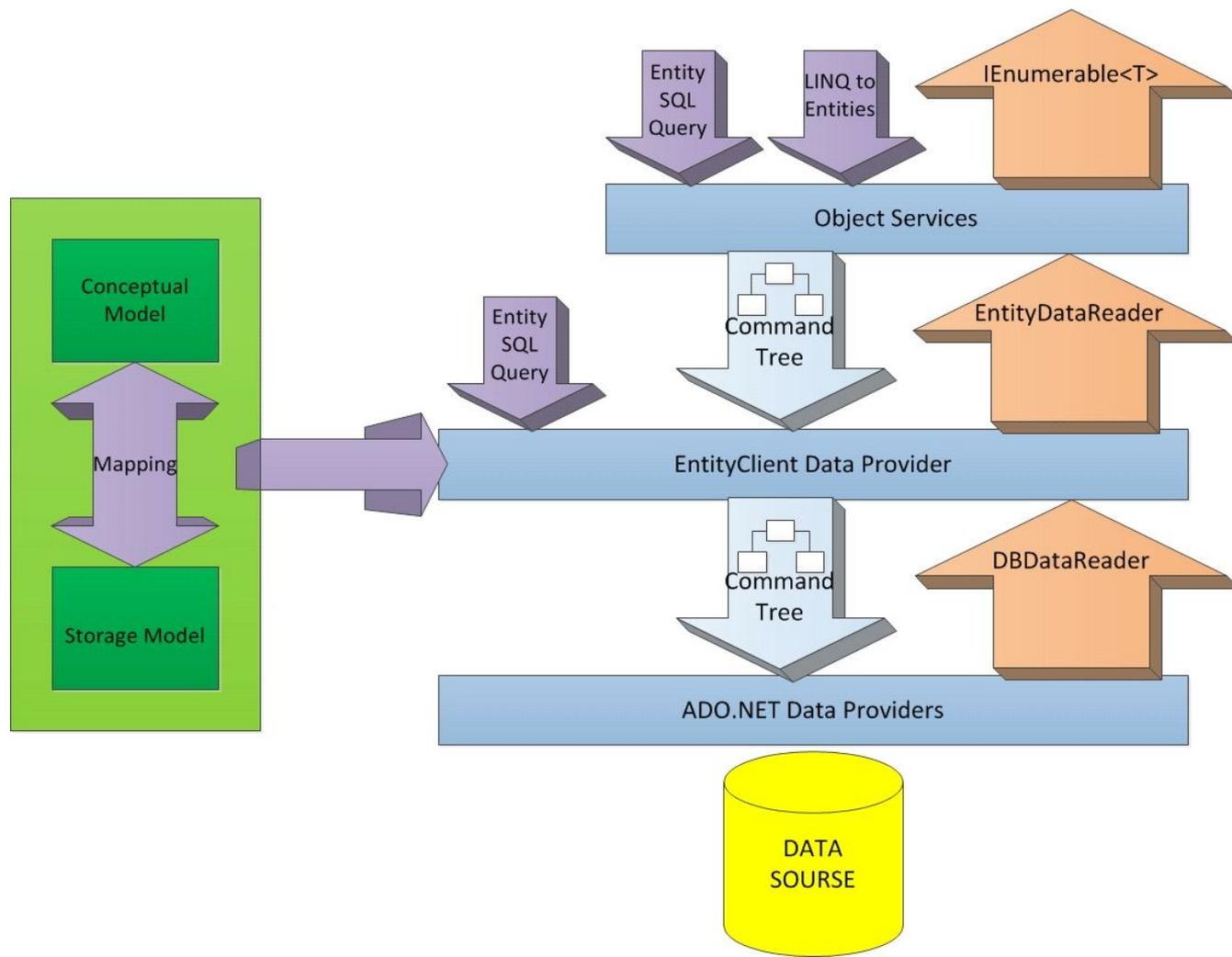
- Поддерживает модель провайдеров, существуют реализации:
  - LINQ to NHibernate
  - LINQ to Amazon
  - SharePoint, ....
- Развивается в направлении параллелизма (manycore) –
  - PLINQ

# Entity Framework

- Инфраструктура формирования концептуального представления данных с помощью сущностей (Entities)
- Реализация классических задач ORM
- Абстрагирование от схемы хранения
- Гибкий маппинг



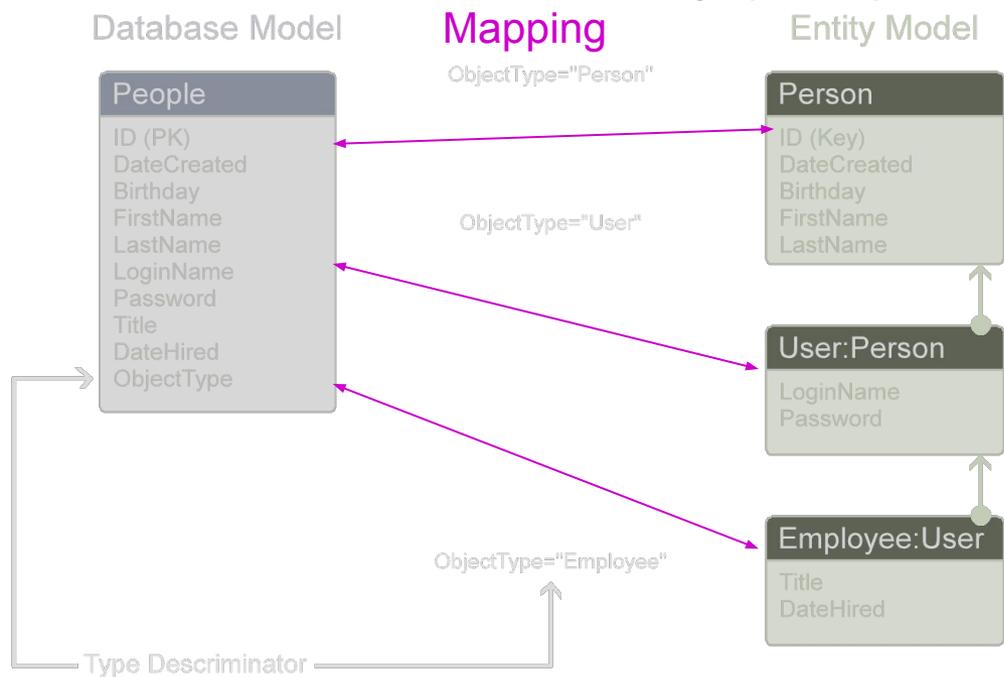
# Архитектура Entity Framework



# Гибкий маппинг

## Таблица на иерархию

### Table Per Hierarchy (TPH)



# Гибкий маппинг.

## Таблица на тип

### Table Per Type (TPT)

Database Model



Mapping

Entity Model



Inherited Members

# Гибкий маппинг.

## Таблица на класс

### Table Per Class (TPC)

Database Model



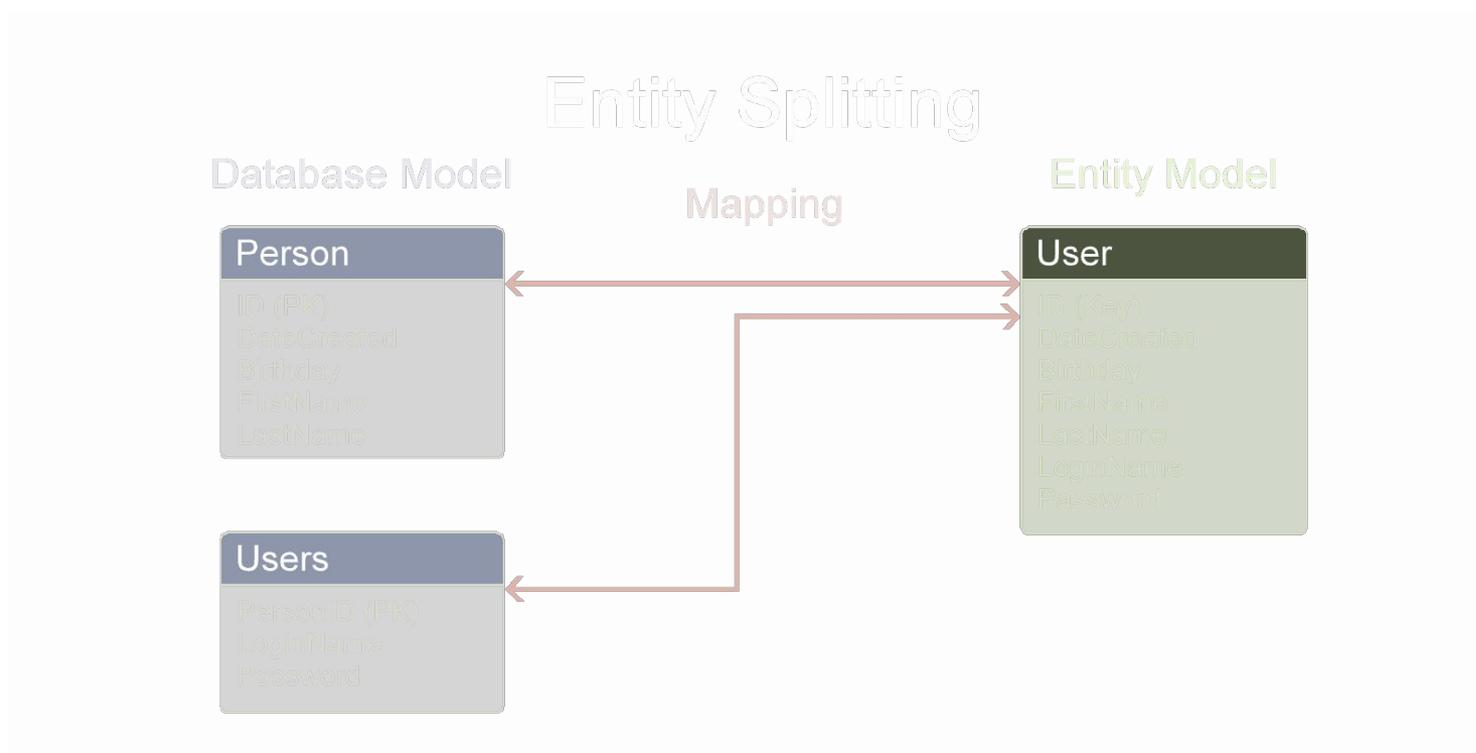
Mapping

Entity Model



# Гибкий маппинг.

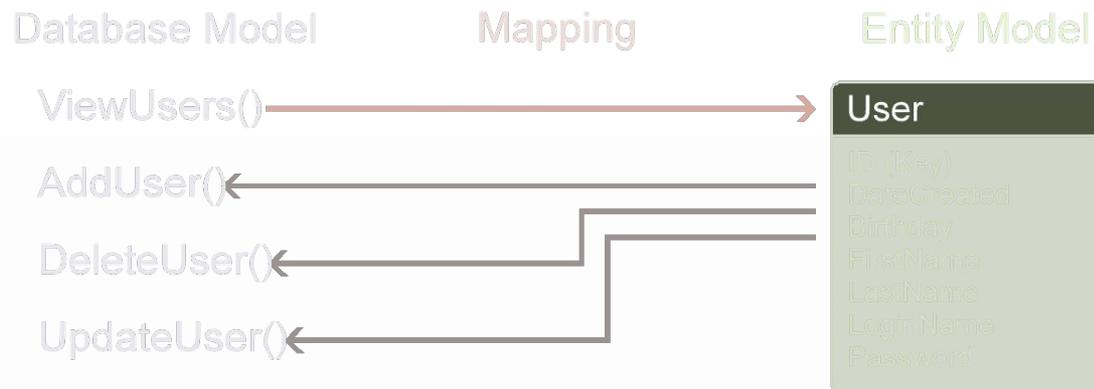
## *Разделение сущности*



# Гибкий маппинг.

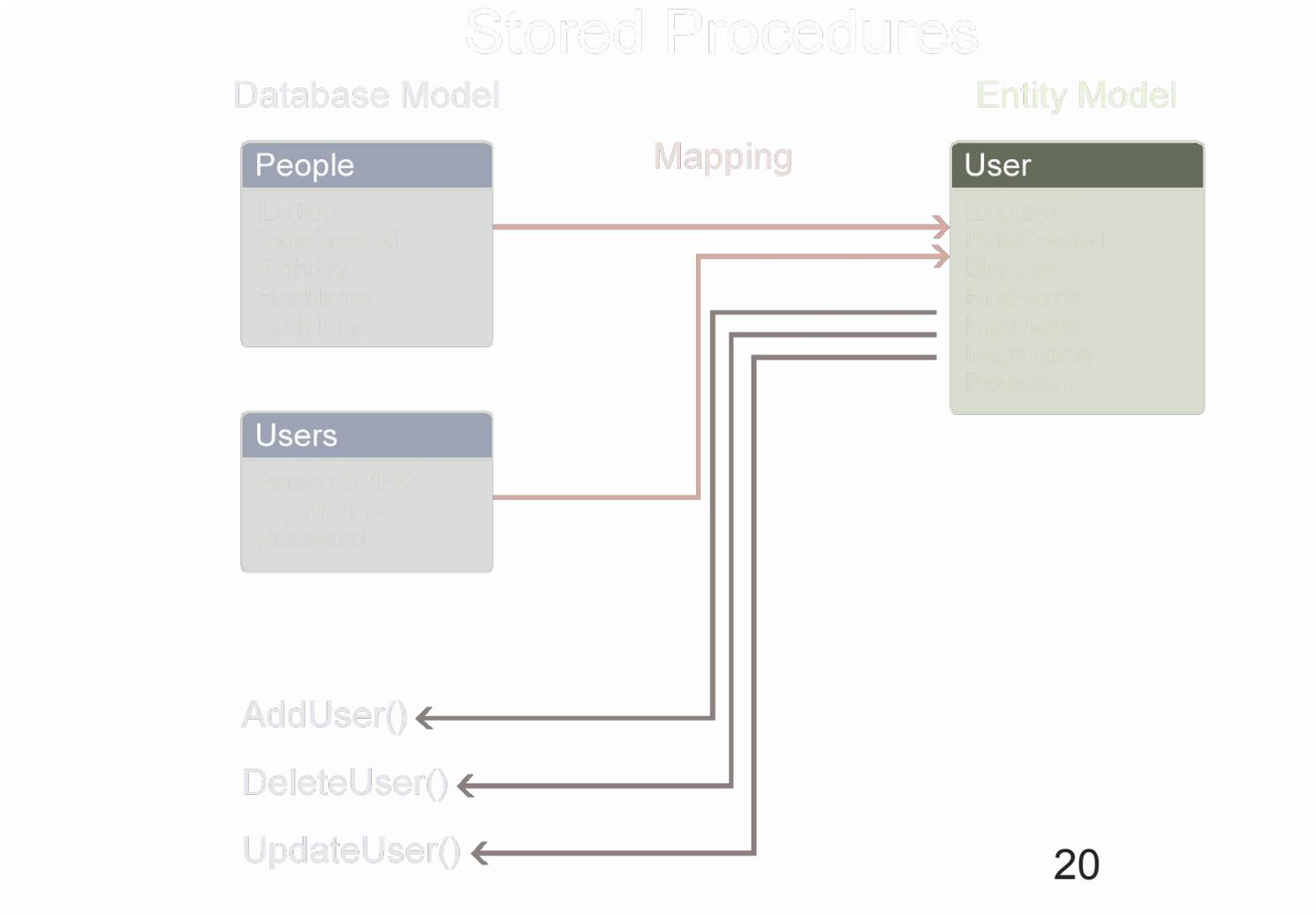
## *Хранимые процедуры*

### Stored Procedures



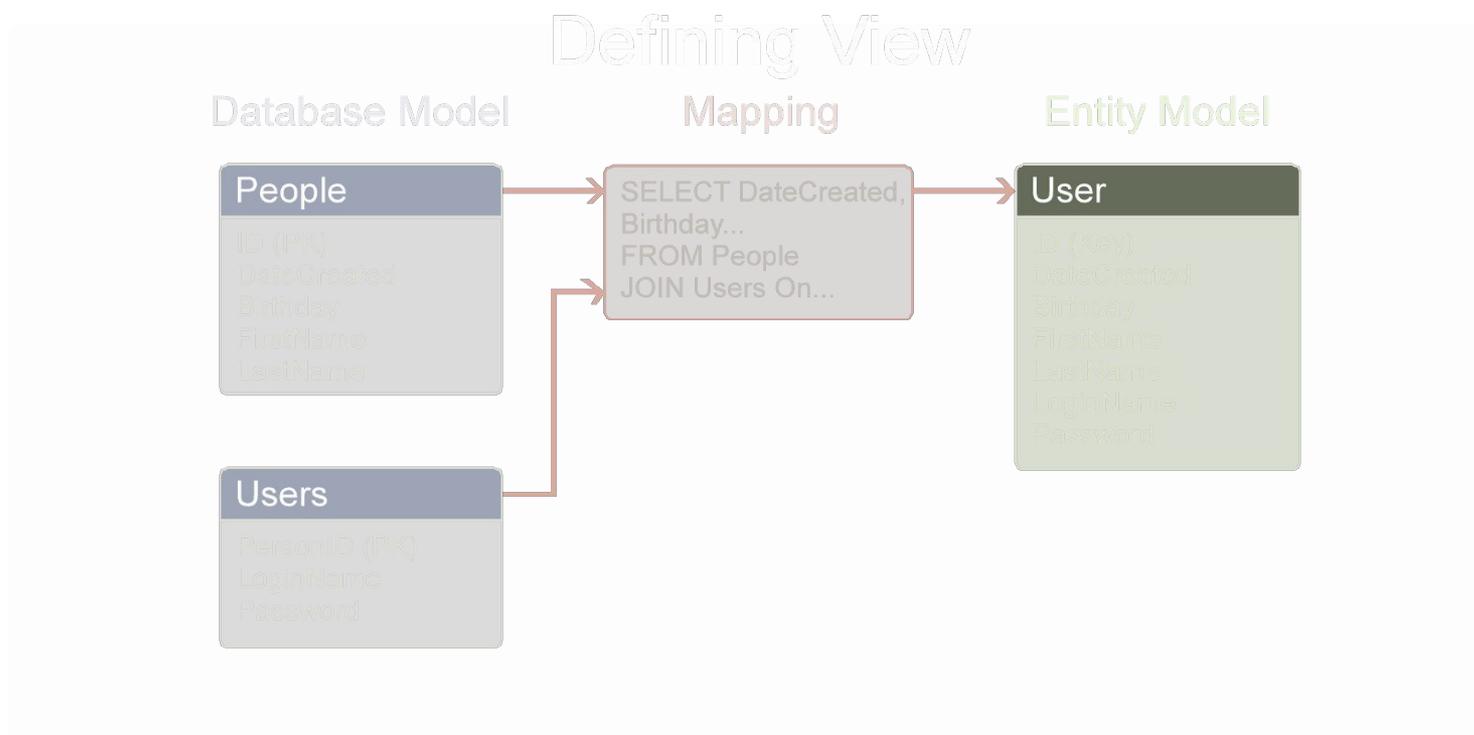
# Гибкий маппинг.

## *Хранимые процедуры*



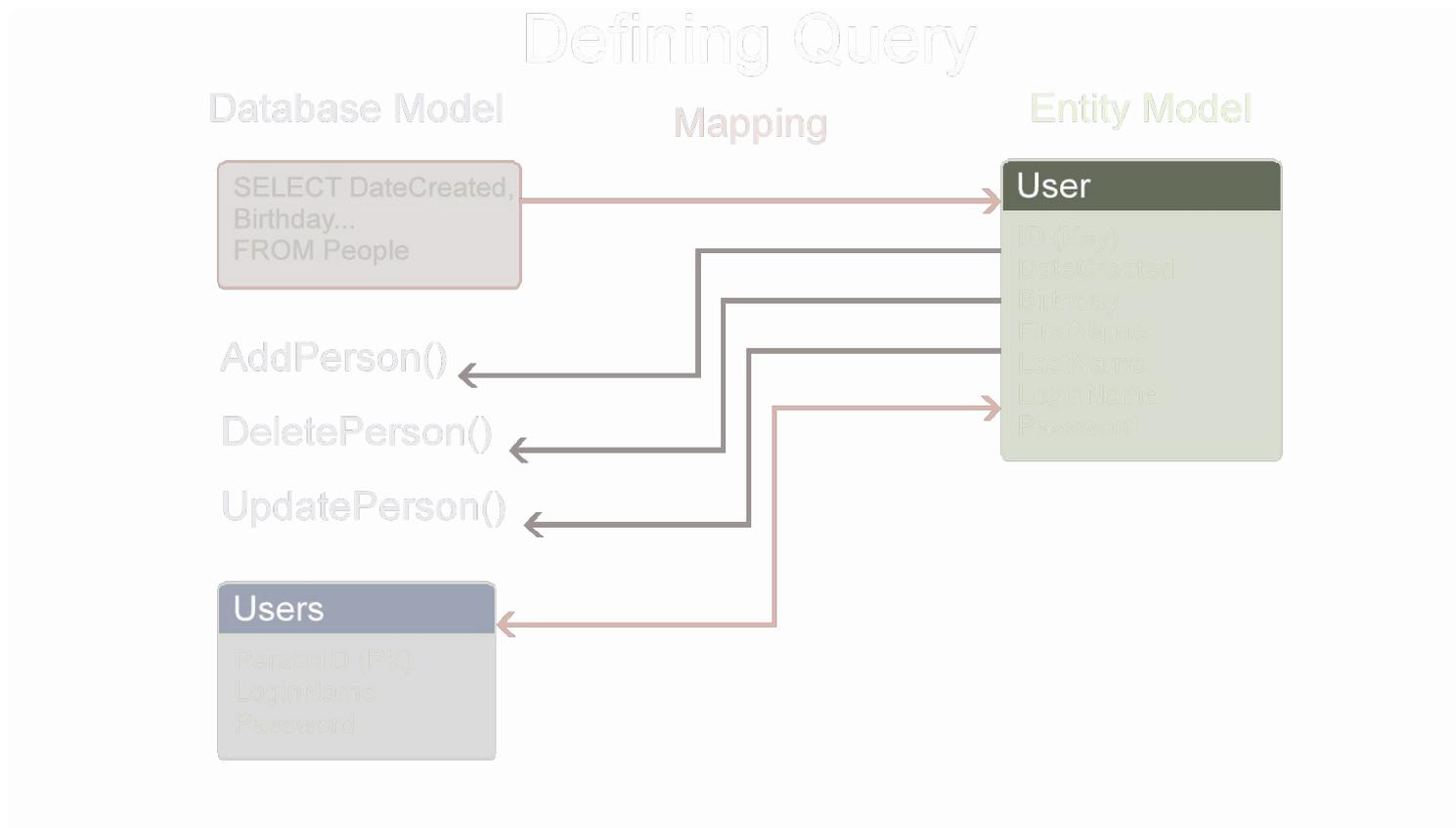
# Гибкий маппинг.

## Создание представлений (QueryView)



# Гибкий маппинг.

## Создание представлений (*Defining Query*)

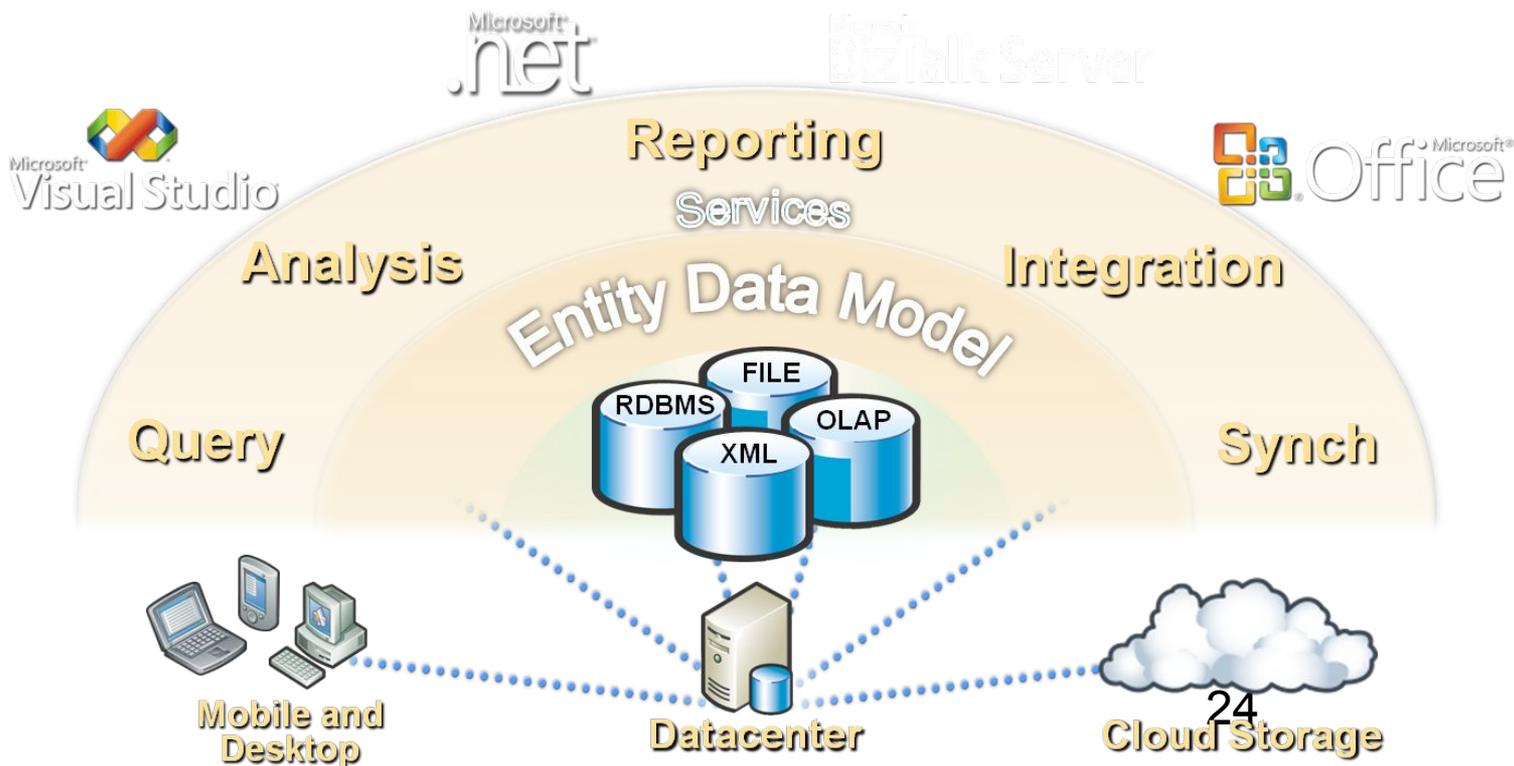


# Entity Framework - текущее состояние

- Доступен в Beta 3
- RTM вышел во второй половине 2008
- Провайдеры третьих производителей для различных баз данных
  - В течение 3 месяцев после выхода (RTM)
    - Oracle, IBM DB2, MySQL, PostgreSQL, Informix, Ingres, Sybase, SQLite, SQL Anywhere, Progress ...
    - Firebird

# Роль Entity Framework

- Единый движок концептуального представления и ORM для ряда ключевых продуктов Microsoft



# Сравнение LINQ и Entity Framework

## LINQ

## Entity Framework

### *Единообразные запросы*

- Единообразные типизированные запросы к любому источнику
- Типизированные результаты

### *Гибкий маппинг*

- Создание концептуального объектного уровня, абстрагированного от логики хранения

### *Основное назначение*

- LINQ to SQL - типизированное представление *логической* схемы данных 1:1 (ORM)
- LINQ to SQL – возможности переименования объектов данных и поддержка TPH

- Типизированное представление *концептуальной* схемы с гибким маппингом - полноценный ORM

### *Возможности маппинга*

# LINQ + Entity Framework

C# 3.0

VB 9.0

Другие

## .NET Language Integrated Query

LINQ to  
Objects

LINQ to  
DataSets

LINQ to  
SQL

LINQ to  
Entities

LINQ to  
XML

Entity Framework

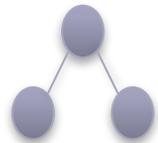
Концептуальная  
схема (CSDL)

МAPPING (MSL)

Схема  
хранения (CSDL)

```
<book>  
<title/>  
<author/>  
<year/>  
<price/>  
</book>
```

XML



Объекты

ADO.NET

Microsoft SQL Server, Oracle, DB2, MySQL...

Реляционные источники



# ССЫЛКИ

## ■ LINQ

<http://msdn2.microsoft.com/en-us/netframework/aa904594.aspx>

## ■ ADO.NET Entity Framework

□ <http://msdn.microsoft.com/data>

□ <http://blogs.msdn.com/adonet/>

□ <http://blogs.msdn.com/data/>