

# Введение в Entity Framework



# Объектно-реляционное отображение

(Object-relational mapping - ORM) технология программирования, предназначенная для автоматического сопоставления и преобразования данных между таблицами реляционных СУБД и объектами классов.



**ORM - системы автоматически генерируют SQL запросы для выполнения операций над данными при выполнении вызовов.**

Функционал ORM систем:

- Создание объектной модели по БД
- Создание схемы БД по объектной модели
- Выполнение запросов к БД с помощью LINQ

## Схема БД

## Сущности (Классы)



**Entity Framework (EF)** - это объектно-реляционный модуль сопоставления (соответствия), позволяющий разработчикам .NET работать с реляционными данными с помощью специализированных объектов.

### **Entity Framework (EF):**

- объектно-ориентированная технология доступа к данным;
- объектно-реляционное отображение (ORM) для .NET Framework от Microsoft.

**Entity Framework** - представляет собой «обёртку» для ADO.NET по взаимодействию с базой данных, базирующуюся на ORM

## История:

- Первая версия **EntityFramework 1.0** вышла в 2008 году и представляла ограниченную функциональность, базовую поддержку **ORM** и один подход к взаимодействию с БД – **«DatabaseFirst»**.
- В 2010 году вышла **EntityFramework 4.0** версия и с этого времени EntityFramework **стал рекомендуемой технологией** для доступа к данным, а в Framework были введены новые возможности взаимодействия с БД - подходы **«ModelFirst»** и **«CodeFirst»**.
- Дополнительные улучшения функционала последовали с выходом версии **EntityFramework 5.0** в 2012 году.
- В 2013 году был выпущен **EntityFramework 6.0**, обладающий возможностью **асинхронного доступа** к данным.

# История:



Visual Studio  
2008 SP 1



EF Version 1



Visual Studio  
2010



EF Version 4



EF 4.1 - 4.3



Visual Studio  
2012

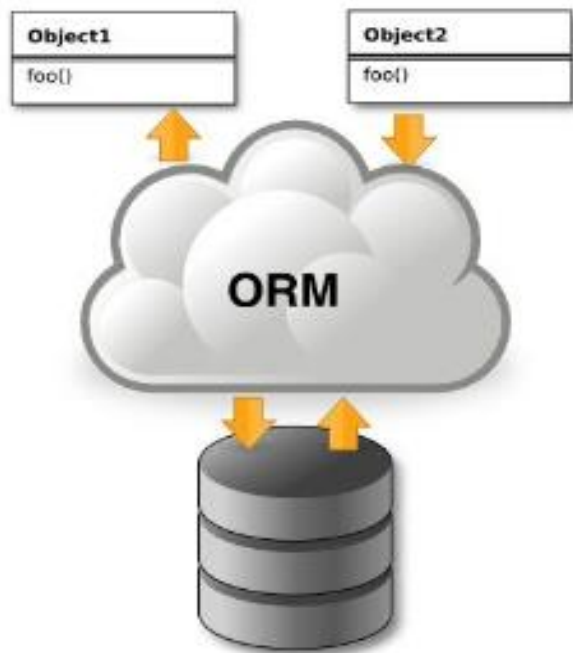


EF Version 5



EF Version 6

# Достоинства применения Framework



## Основное достоинство:

- устраняет необходимость в написании большей части кода для доступа к данным, который обычно требуется разработчикам.

- Производительность (быстрота разработки)

*Microsoft любит того, у кого хорошее железо*

- Дизайн приложения
- Повторное использование
- Сопровождаемость

## Основные возможности Entity Framework

- Сопоставление **таблиц базы данных** и связей между ними **в объекты .NET** и отношения между ними;
- Выполнение **запросов к базе данных** через работу с .NET объектами (**linq to entities**);
- Create / Read / Update / Delete
- Создание, изменение, удаление **схемы базы данных**.



# Архитектура Entity Framework

**КЛИЕНТ**

**LINQ to Entities**

**Entity SQL**

**Службы объектов**  
(Object services)

**Клиентский провайдер данных Entity**  
(Entity Client data provider)

**Провайдер данных ADO.NET**  
(ADO.NET data provider)

**База данных**

## Архитектура Entity Framework

**Служба объектов** управляет сущностями клиентской стороны при работе с ними в коде.

### ***Служба объектов:***

- отслеживают изменения, внесенные в сущность, управляет отношениями между сущностями в базе данных и сохраняет состояния сущности с помощью сериализации.

**Служба объектов** управляет любым классом за счет наследования от базового класса

**EntityObject.**

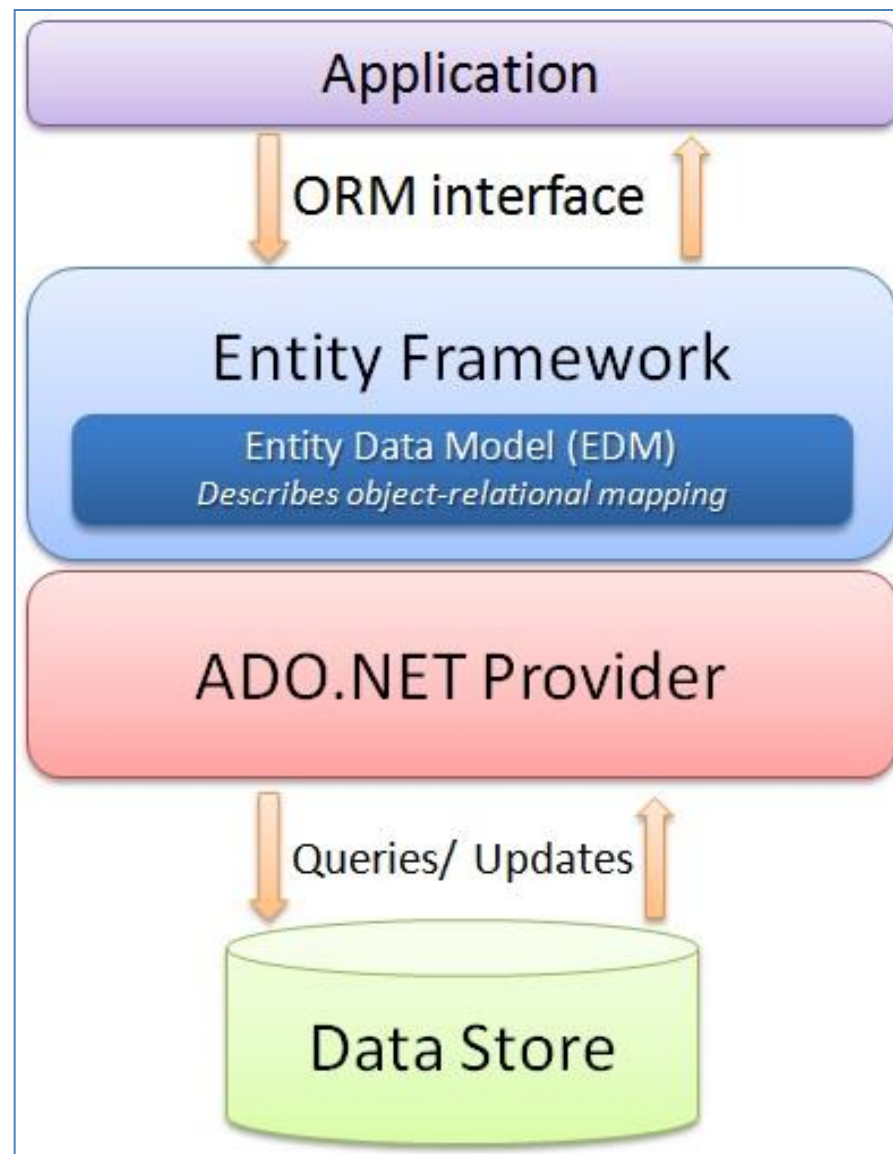
## Архитектура Entity Framework

### Клиентский провайдер данных Entity:

- организует работу с **поставщиком данных ADO.NET** для установки соединений с базой данных,
- генерирует необходимые SQL-операторы на основе состояния сущностей и запросов LINQ,
- отображает извлеченные данные на сущностях,
- другие внутренние операции.

**System. Data. EntityClient** содержит классы **EntityCommand** и **EntityConnection**, **EntityDataReader**.

# Архитектура Entity Framework



## Entity Data Model

**Entity Data Model (EDM)** представляет концептуальную модель между объектной моделью (сущностями) и БД, согласно которой определяются правила соответствия объектов элементам базы данных

Строго типизированные классы, используемые для сопоставления базам данных, называются **сущностями**.

# Entity Data Model

Для корректного отображения данных сущностных классов на данные таблиц используется EDM.

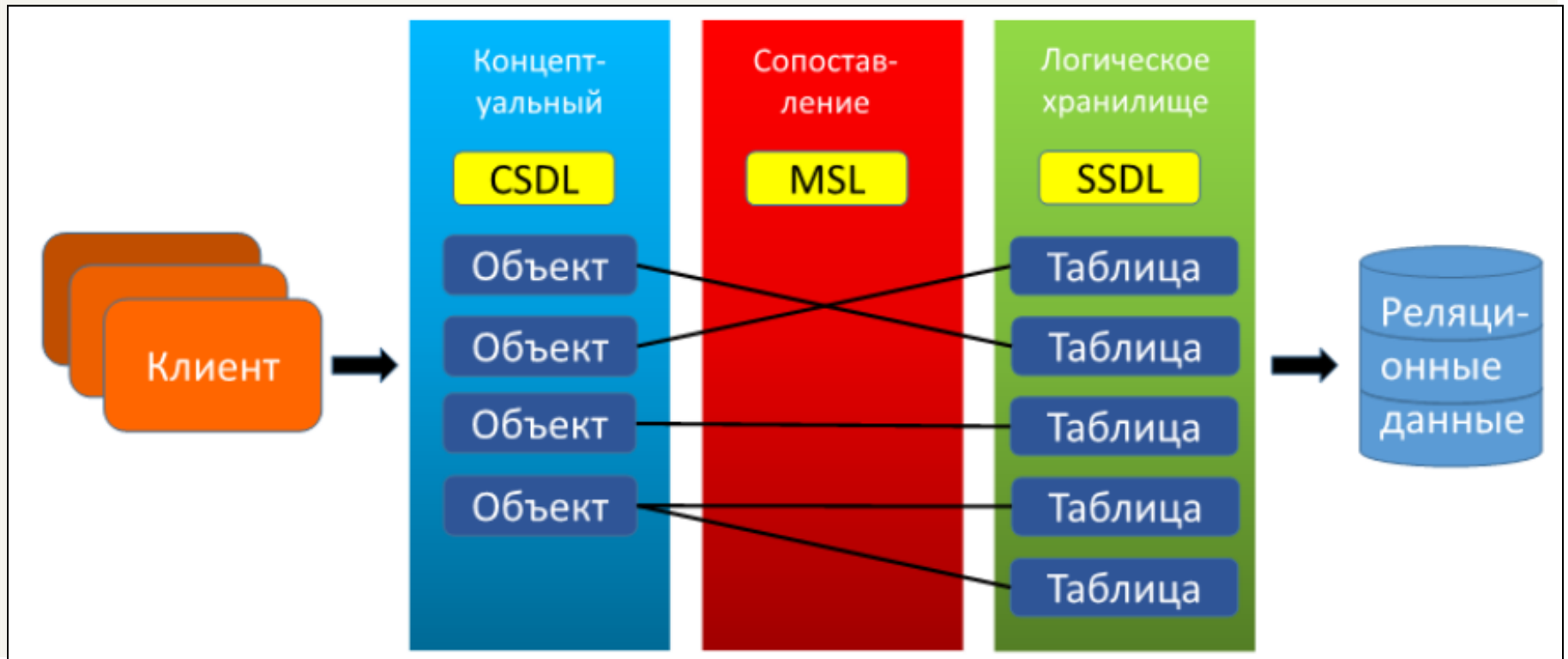
- **Концептуальная модель** определяет сущности отношения между ними (при их наличии).
- **Отображение** описывает сопоставление между концептуальной моделью и моделью хранения приложения Entity Framework
- **Модель хранилища** описывает модель хранения в приложениях Entity Framework.



# Архитектура модели EDM

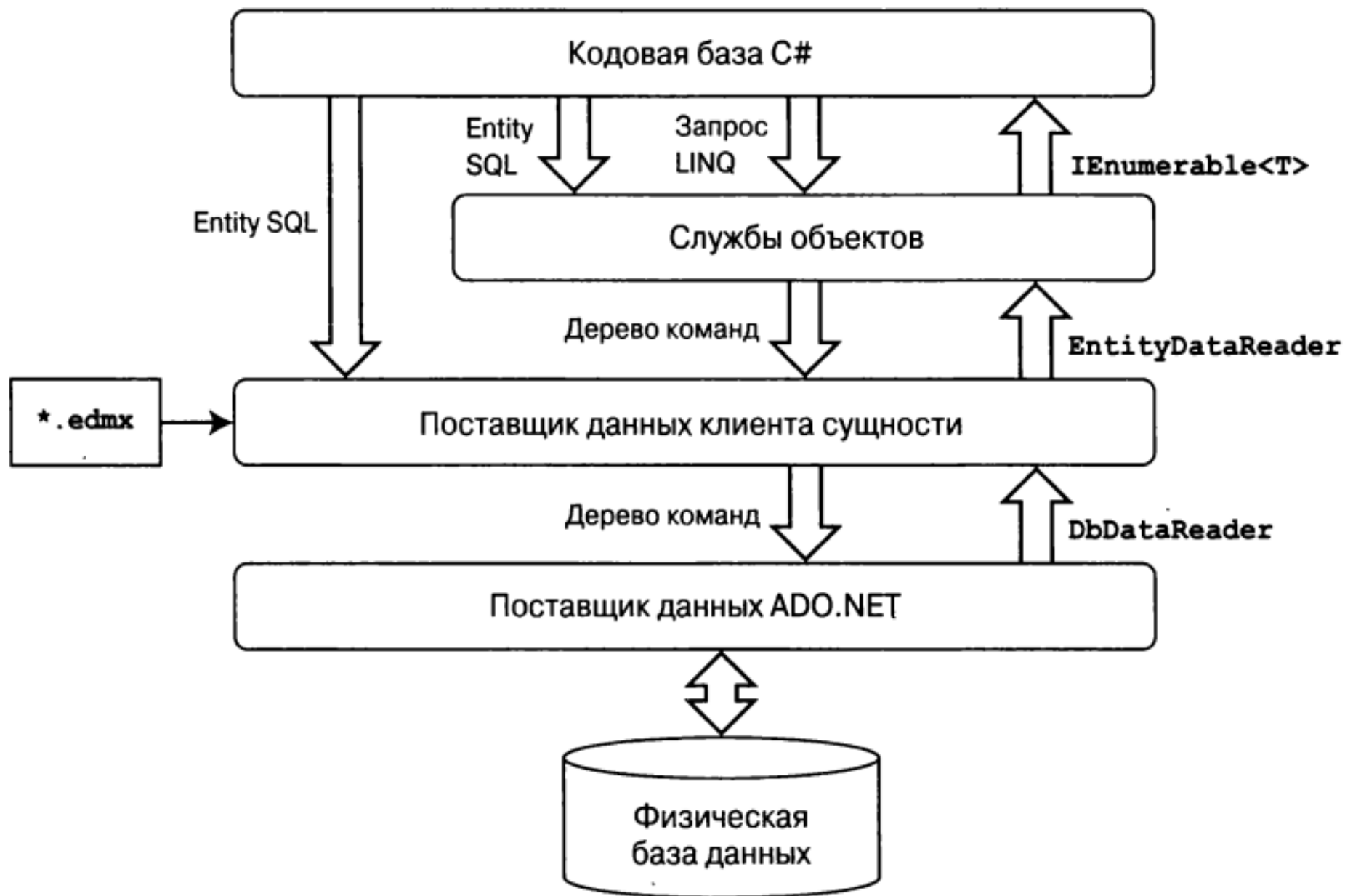
Языки на основе XML, которые описывают концептуальную модель, модель хранения и сопоставление между этими моделями:

- язык определения концептуальной схемы (**CSDL**)
- язык определения схемы хранения (**SSDL**)
- язык определения сопоставлений (**MSL**)

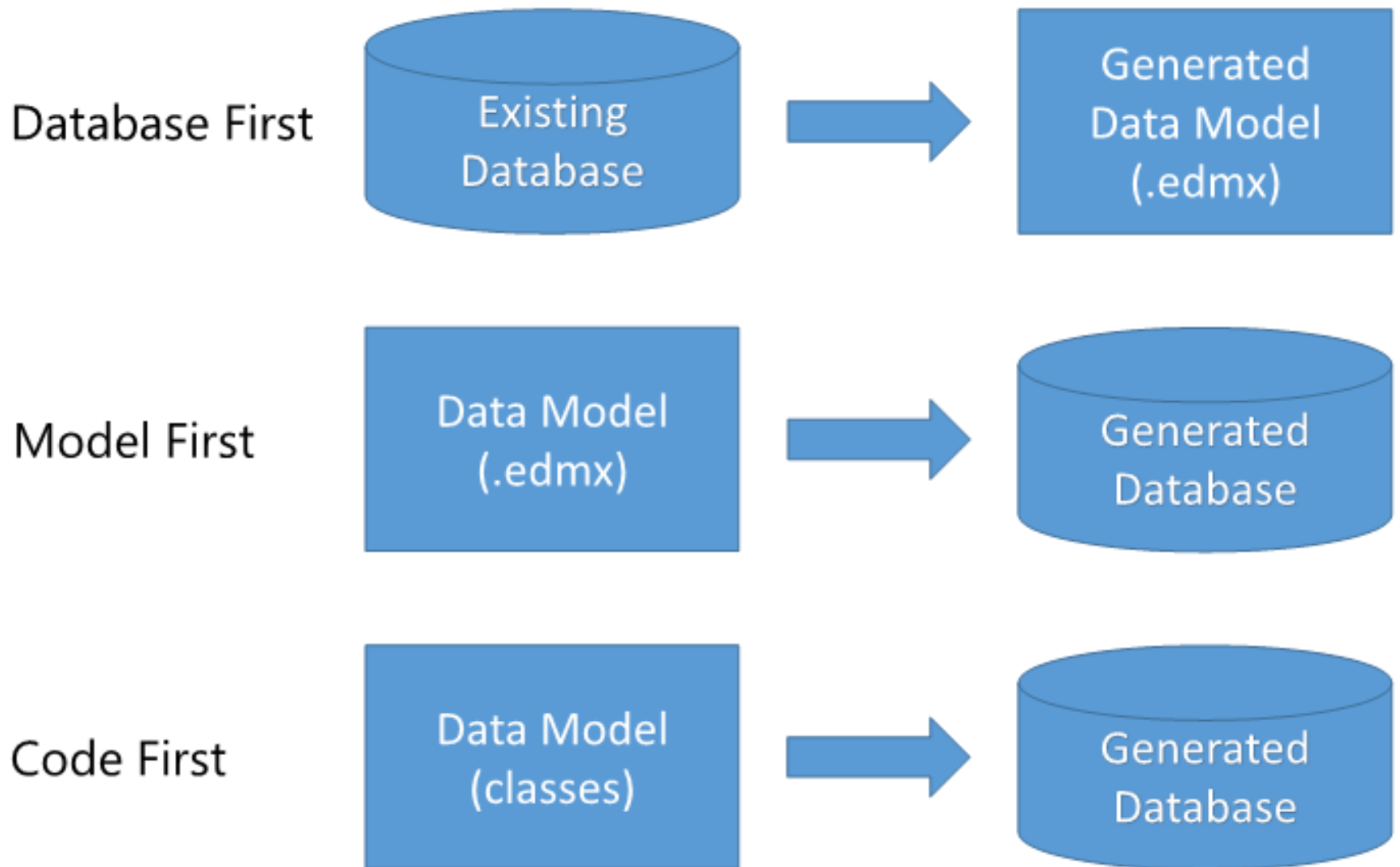




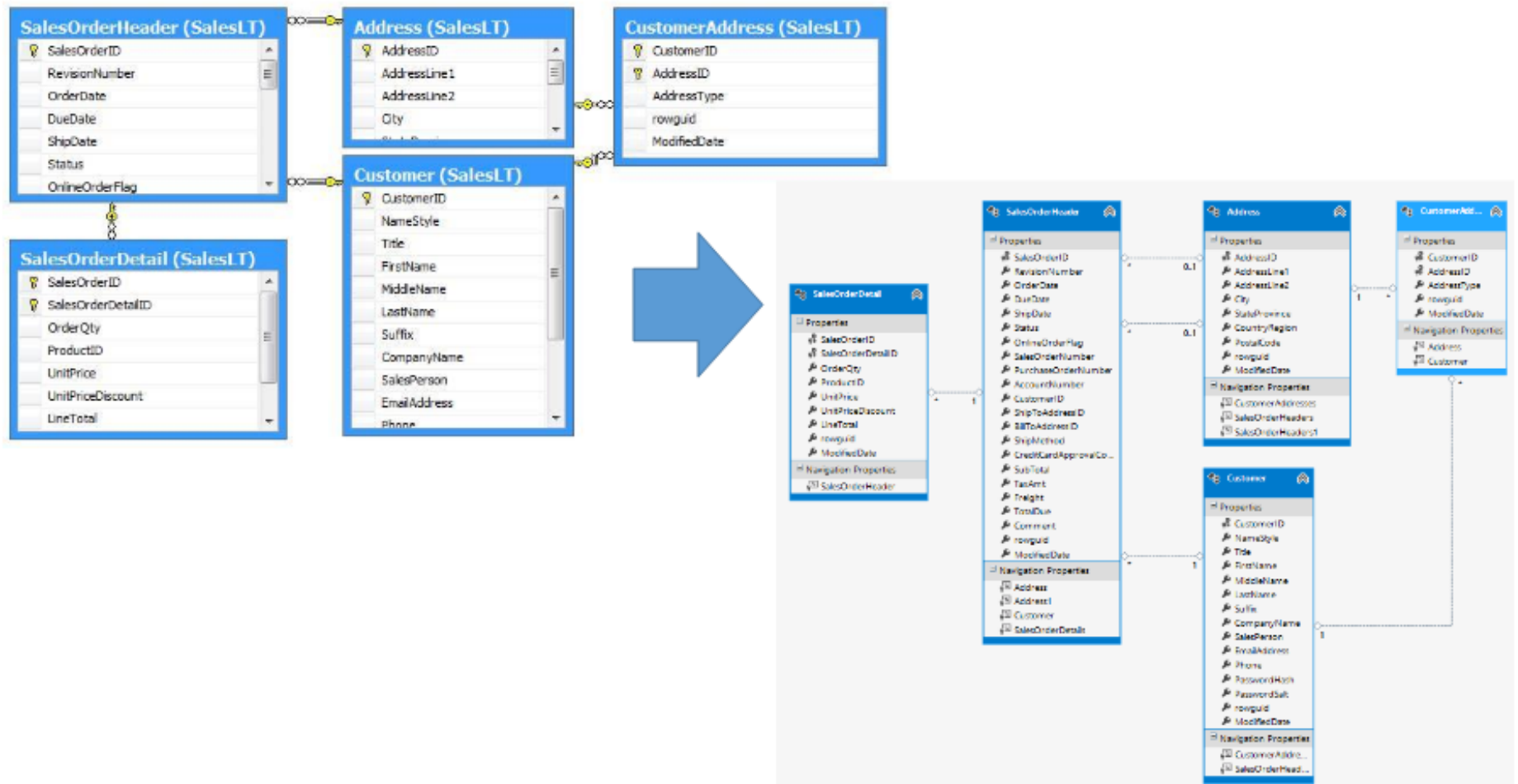




# Основные способы создания моделей



# Database First



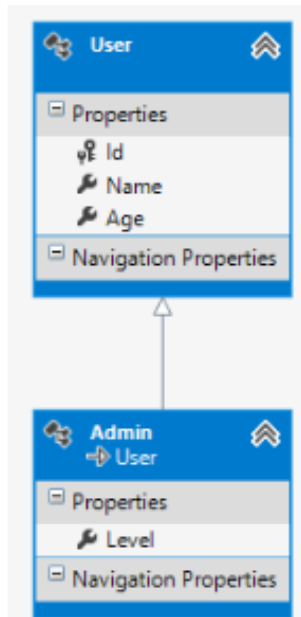
# Model First

Model First

Data Model  
(.edmx)

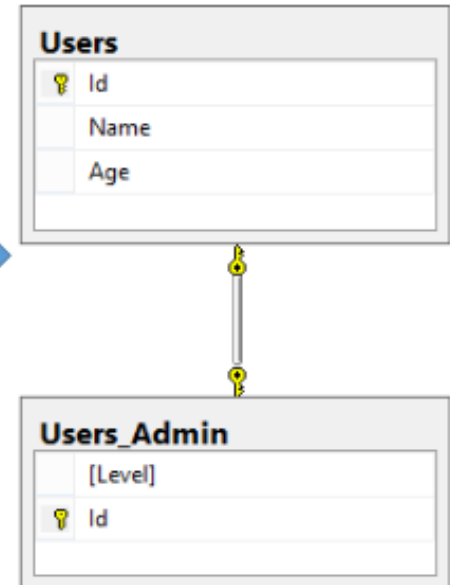


Generated  
Database



```
public partial class User
{
    public int Id { get; set; }
    public string Name { get; set; }
    public DateTime? Age { get; set; }
}

public partial class Admin : User
{
    public string Level { get; set; }
}
```



# Демонстрация

- DataBase First
- Model First

## Основные операции для работы с данными

Создание, чтение, обновление и удаление данных.

- Create
- Read
- Update
- Delete

Entity Framework позволяет легко манипулировать данными благодаря следующим операциям: