

Проектирование ПО

Тема 6. Проектирование БД

Механизмы хранения

Многие системы содержат **устойчивые** объекты, то есть такие, которые можно сохранять на постоянных носителях и впоследствии извлекать при необходимости. Для этого чаще всего используют:

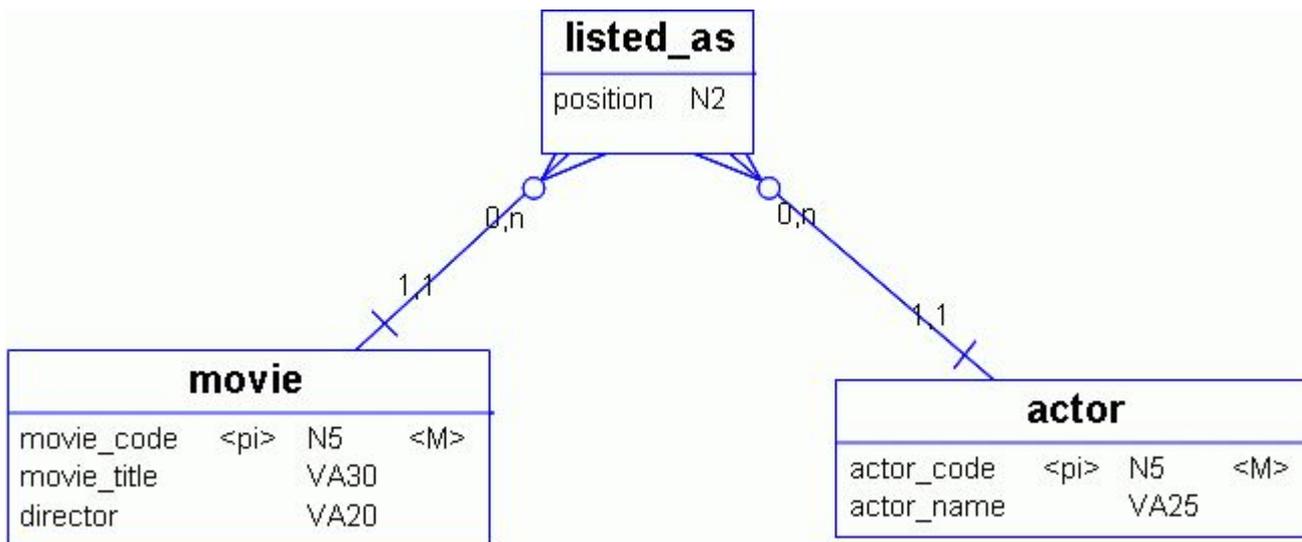
- **реляционные БД,**
- **объектно-ориентированные БД,**
- гибридные **объектно-реляционные БД,**
- **иерархические БД,**
- текстовые файлы, XML-файлы и пр.

Проектирование данных сводится к построению **моделей данных.**

UML позволяет создавать:

- **концептуальные модели,**
- **логические схемы БД и**
- **физические модели БД.**

Концептуальная модель



Information Engineering (IE), воронья лапа (Crow's Foot) - стандарт;

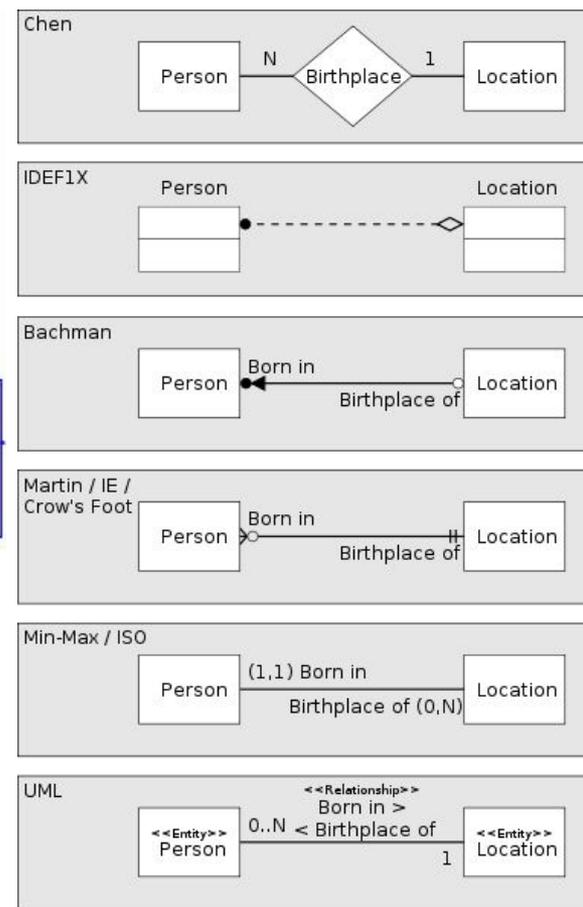
IDEF1X - стандарт;

Entity/Relationship (ER) – производная от Information Engineering; используется в CASE-средствах Sybase.

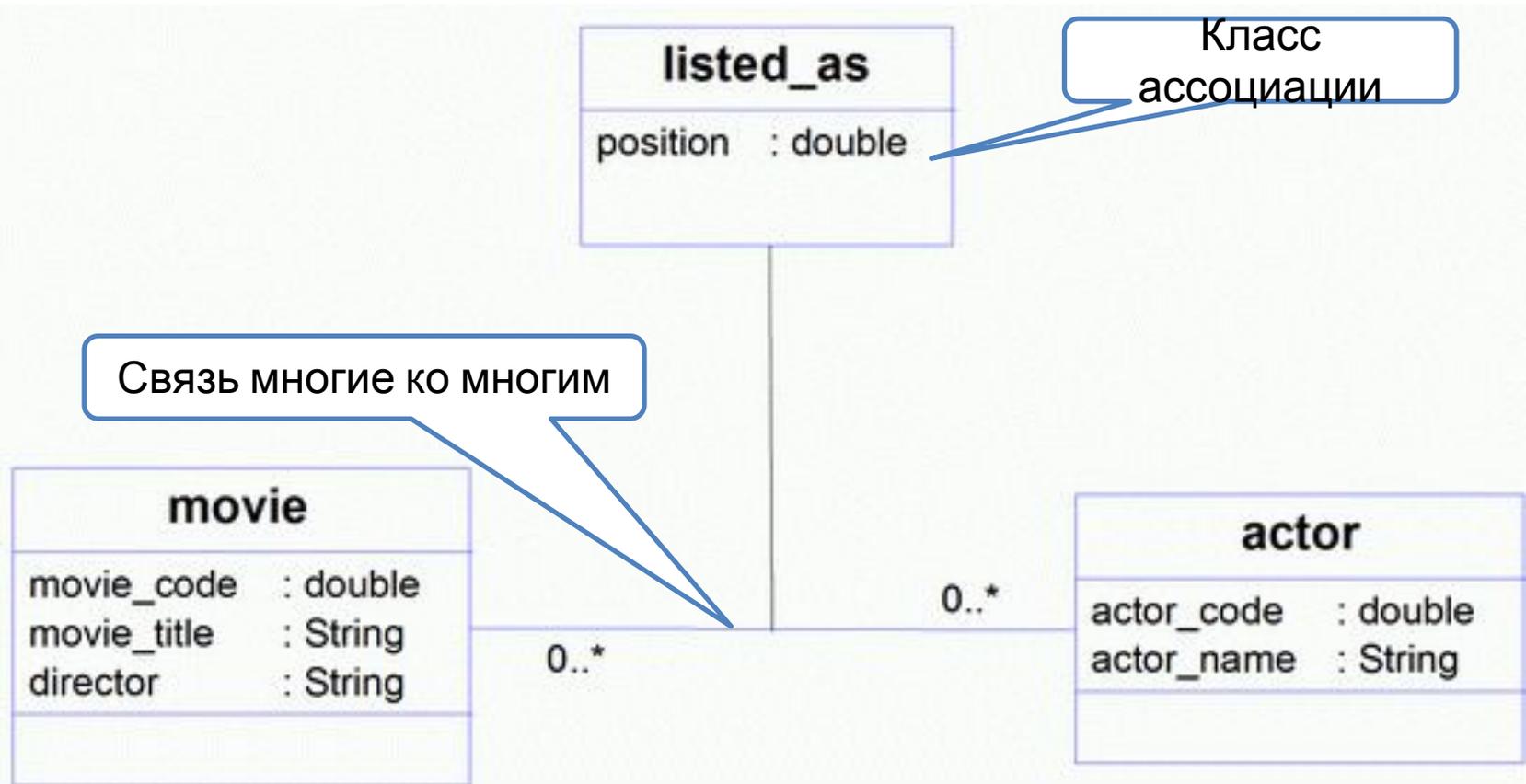
Нотация Баркера - стандарт; разработал Ричард Баркер; используется в CASE-средствах Oracle;

Merise - использует ассоциации вместо связей;

Нотация Чена - стандарт; разработал Питер Чен; отношения отображаются вершинами.

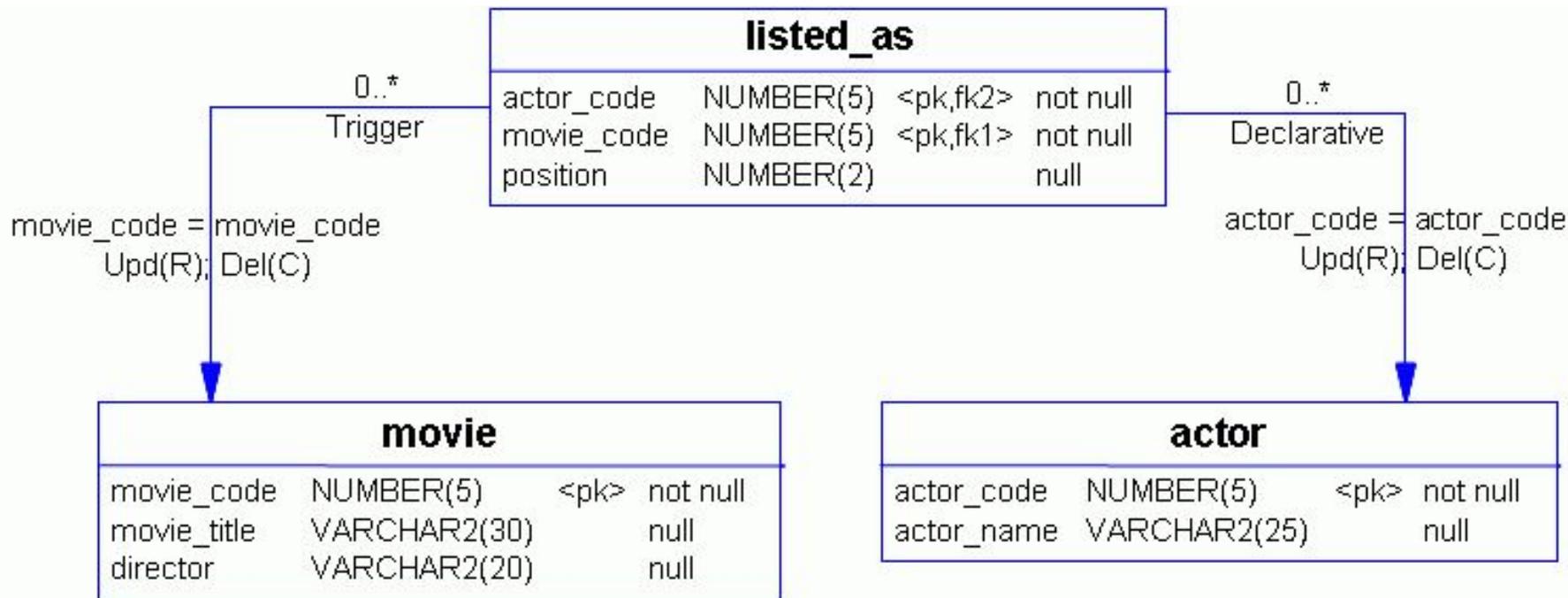


Концептуальная UML-модель классов



Класс ассоциации — класс, который представляет отношение ассоциации между другими классами.

Логическая модель данных



Логические модели данных предлагают независимые от конкретной СУБД реляцион-ные структуры, которые могут использоваться разработчиками для оптимизации и понимания метаданных. Логические модели данных можно разрабатывать отдельно или генерировать из концептуальных моделей данных.

Ссылочная целостность в правом отношении декларативная, а в левом отношении она является процедурной (триггером).

Поддержка целостности

БД представляет собой собрание записей, связанных ссылочной целостностью.

Стрелки показывают направление ссылки внешнего ключа на первичный ключ.

Эти отношения между записями должны соответствовать **бизнес-правилам**. Все операции по модификации: **insert**, **update** и **delete** проверяются на соответствие бизнес-правилам.

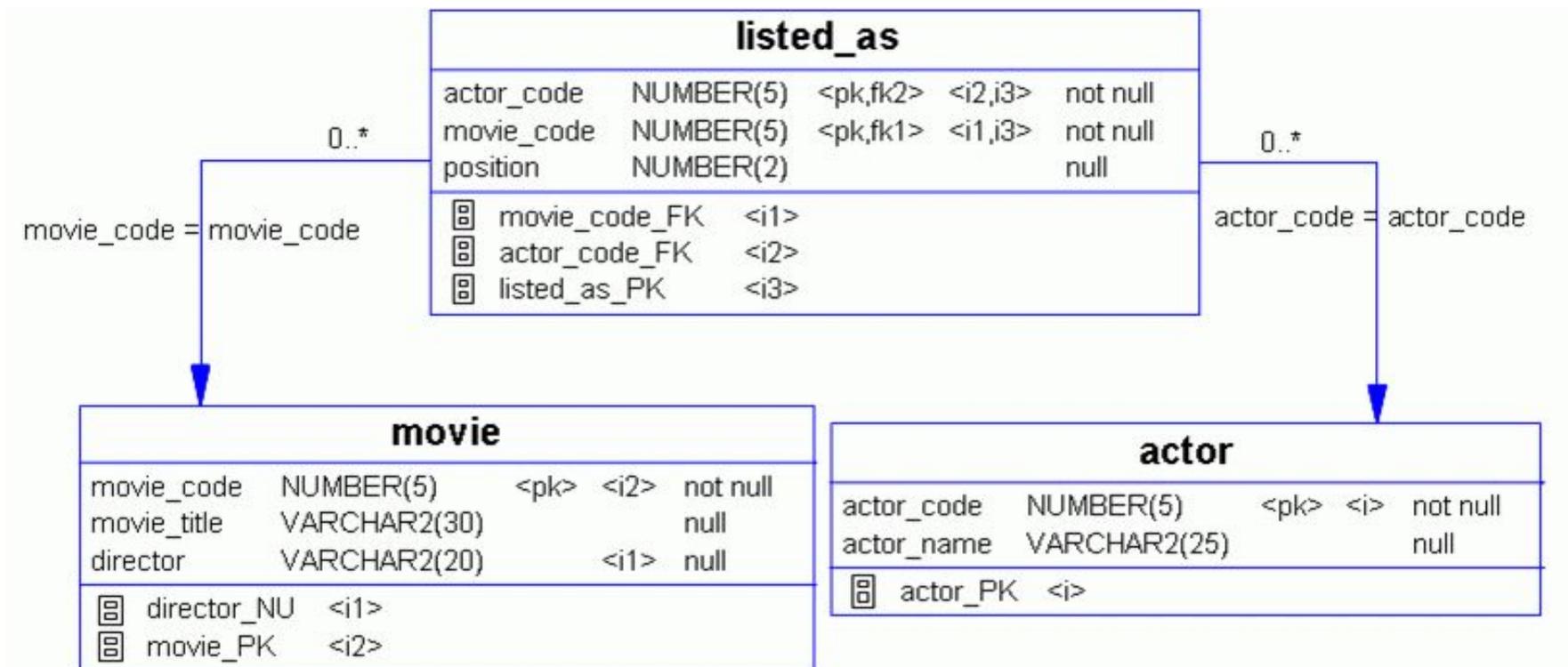
Декларативная ссылочная целостность реализуется как простое объявление в SQL-операторе `create table` (`alter table`), которое указывает, что делать, когда запись в этой таблице должна быть удалена, изменена или добавлена.

Процедурная ссылочная целостность использует язык программирования БД. Программы, написанные на таком языке, называются **триггерами**. Отдельный триггер может быть написан для каждой из трех операций модификации каждой таблицы.

Поддержка операций удаления и корректировки для декларативной ссылочной целостности позволяет использовать следующие варианты:

- **cascade** (каскадный),
- **restrict** (ограниченный) ,
- **nullify** (задание пустого указателя),
- **default** (значение по умолчанию).

Физическая модель данных



Типы атрибутов соответствуют типам конкретной СУБД и добавляются индексы.

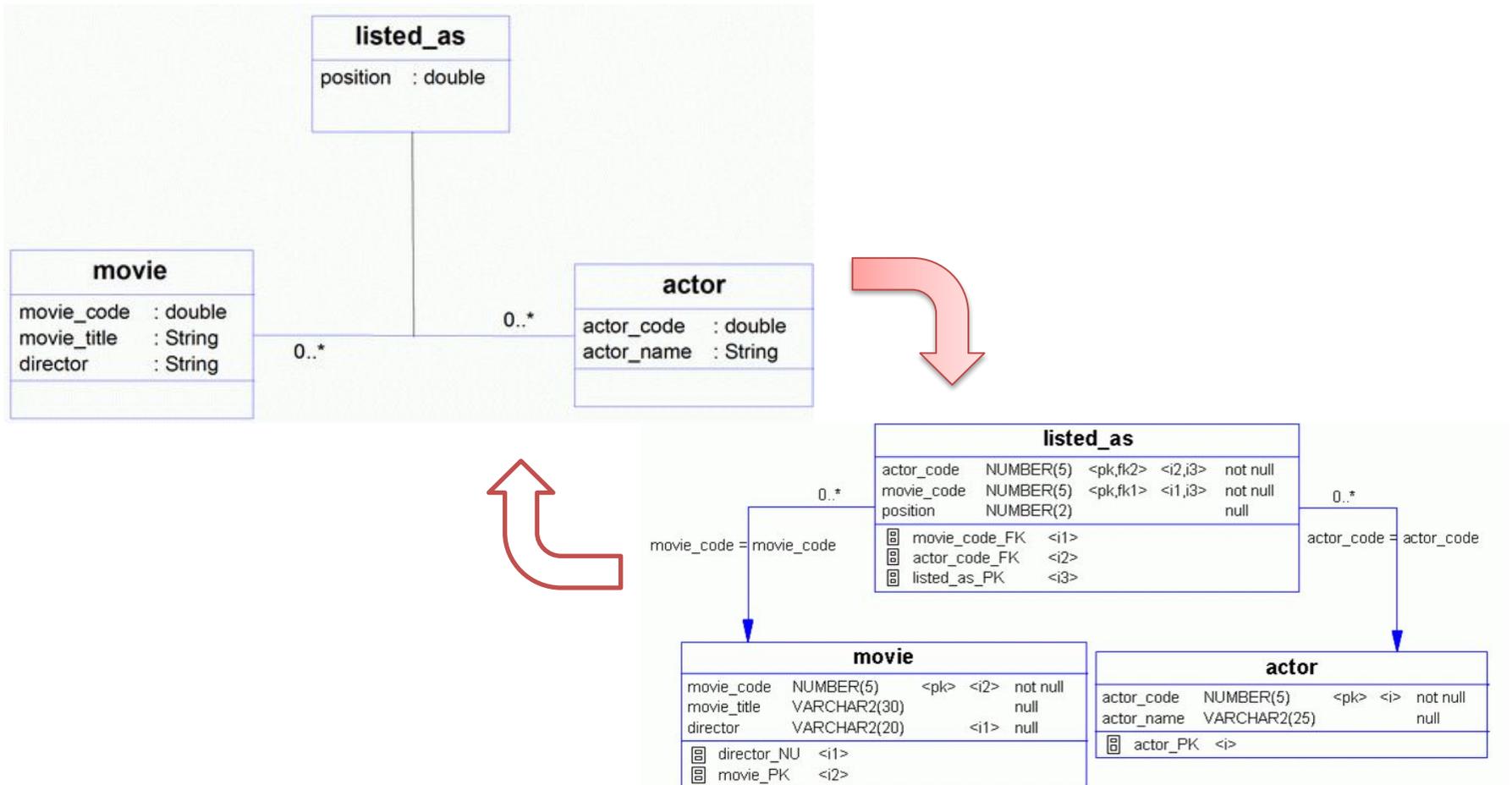
Скрипт для генерации БД

```
create table actor (  
actor_code    NUMBER(5)    not null,  
actor_name    VARCHAR2(25) )  
create unique index actor_PK on actor ( actor_code ASC )  
/  
create table movie (  
movie_code    NUMBER(5)    not null,  
movie_title    VARCHAR2(30),  
director      VARCHAR2(20),  
constraint PK MOVIE primary key (movie_code) )  
create index directorJSJU on movie ( director ASC )  
/  
create table listed_as (  
actor code    NUMBER(5)    not null,  
movie_code    NUMBER(5)    not null,  
position      NUMBER(2),  
constraint PK_LISTED_AS primary key (actor_code,movie_code),  
constraint FK_LISTED_A_REFERENCE_ACTOR foreign key  
(actor_code) references actor (actor code) on delete cascade )  
create index movie_code_FK on listed_as ( movie_code ASC )  
create index actor_code_FK on listed_as ( actor_code ASC )  
/  
Проектирование ПО. Проектирование БД
```

Если физическая модель БД определена с помощью CASE-средства, то оно может автоматически создать скрипт для генерации БД.

Объектно-реляционное отображение

Объектно-реляционным отображением называется отображение концептуальных классов в таблицы



Отображение объектов

Шаблон Представление объектов в виде таблиц (Representing Objects as Tables)

Для каждого класса объектов, подлежащих постоянному хранению, определить отдельную таблицу, а атрибуты объекта, содержащие данные простых типов хранить в отдельных столбцах.

Шаблон Идентификатор объектов (Object Identifier)

Каждой записи и объекту присваивается уникальный, автоматически формируемый, неизменный идентификатор объекта (Object Identifier — OID). Большинство CASE-средств позволяет определять идентификаторы, вводя их как дополнительные свойства атрибутов класса.

Каждая таблица в качестве первичного ключа использует идентификатор объекта, прямо или косвенно связанный с каждым объектом.

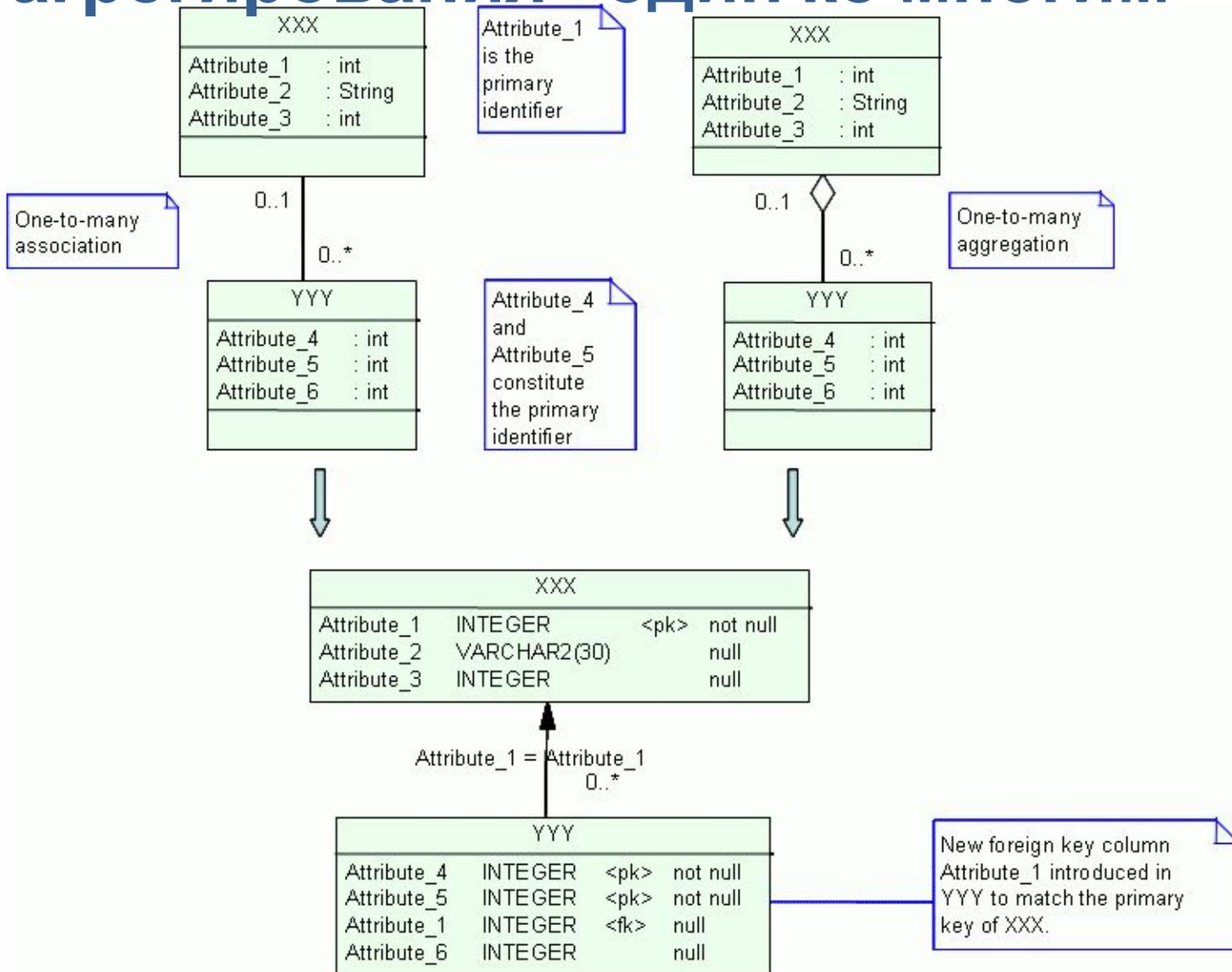
UML не имеет никакого визуального способа показать идентификаторы классов, кроме вспомогательных стереотипов или примечаний. В UML есть понятие *профиля* (UML profile), под которым понимается набор взаимосвязанных стереотипов, тегированных значений и ограничений, объединенных общим предназначением. Так для моделей данных можно использовать профиль, расширяющий семантику диаграммы классов необходимыми средствами.

Отображение ассоциаций

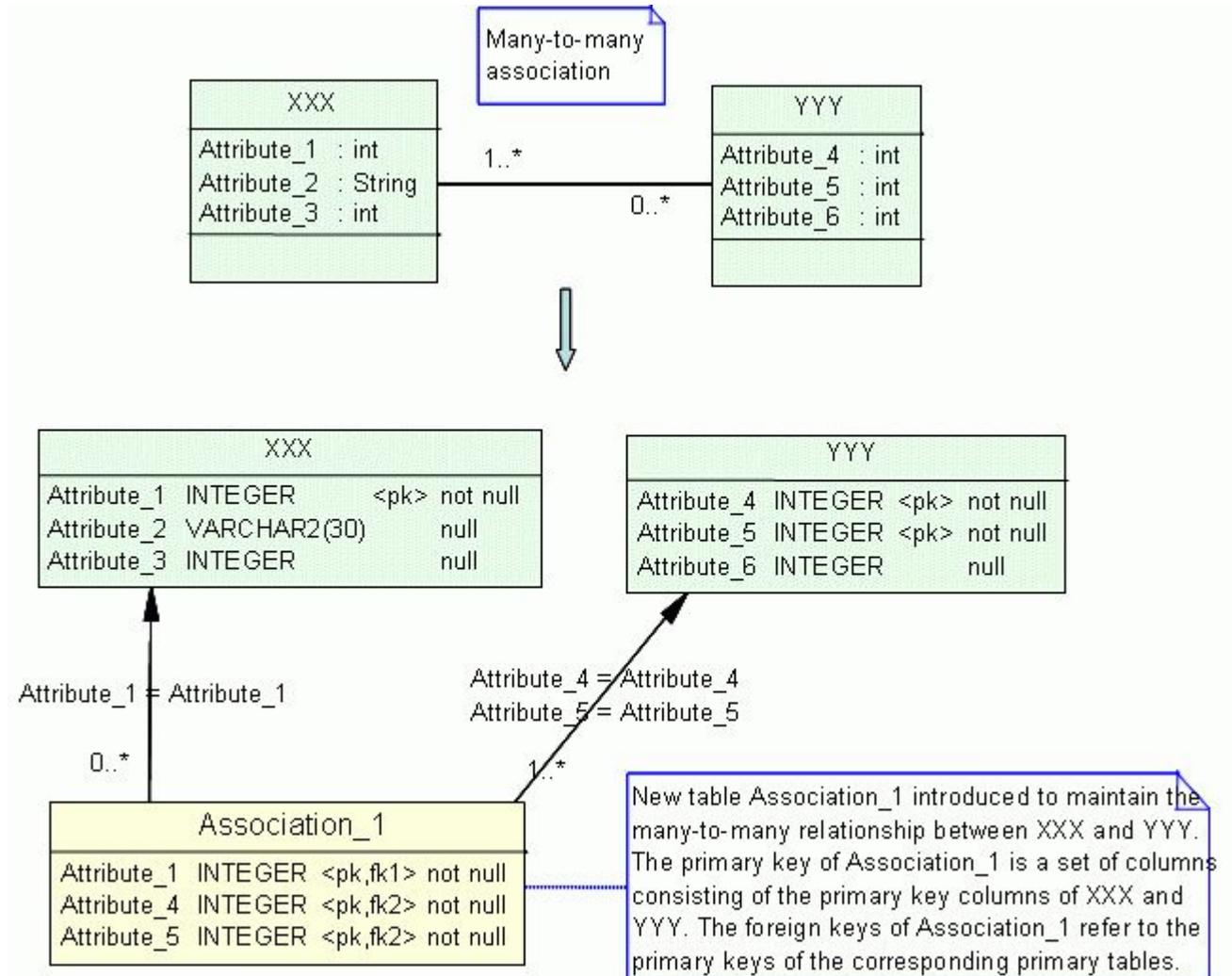
Концептуальные классы — просто контейнеры данных. Операции обычно не рассматриваются в концептуальных классах. Это делает отображение в таблицы более простым.

Главная проблема заключается в **отображении отношений** между классами. Рассматриваемые отношения включают ассоциации с различными множественностями, агрегированиями и обобщениями.

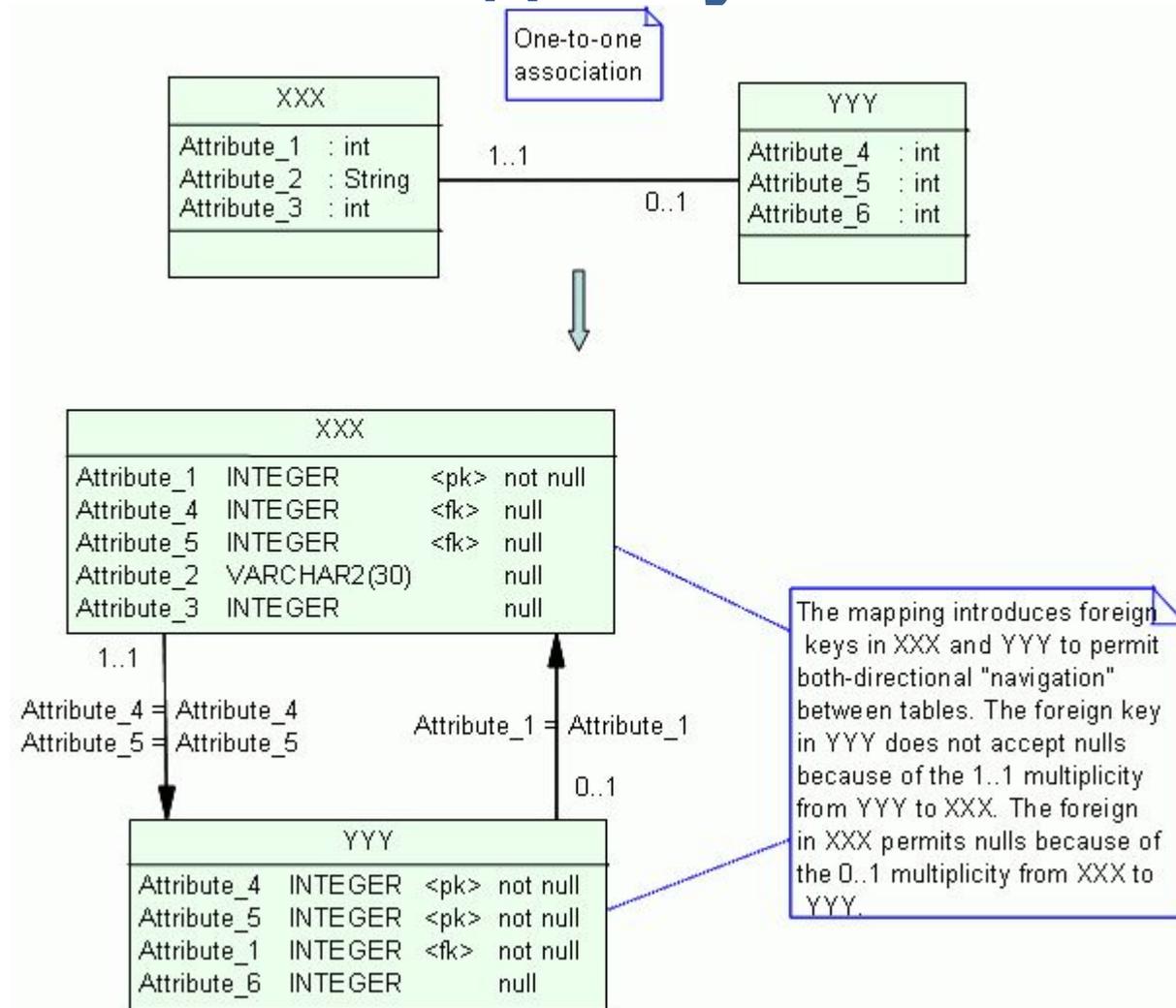
Отображение ассоциации и агрегирования «ОДИН КО МНОГИМ»



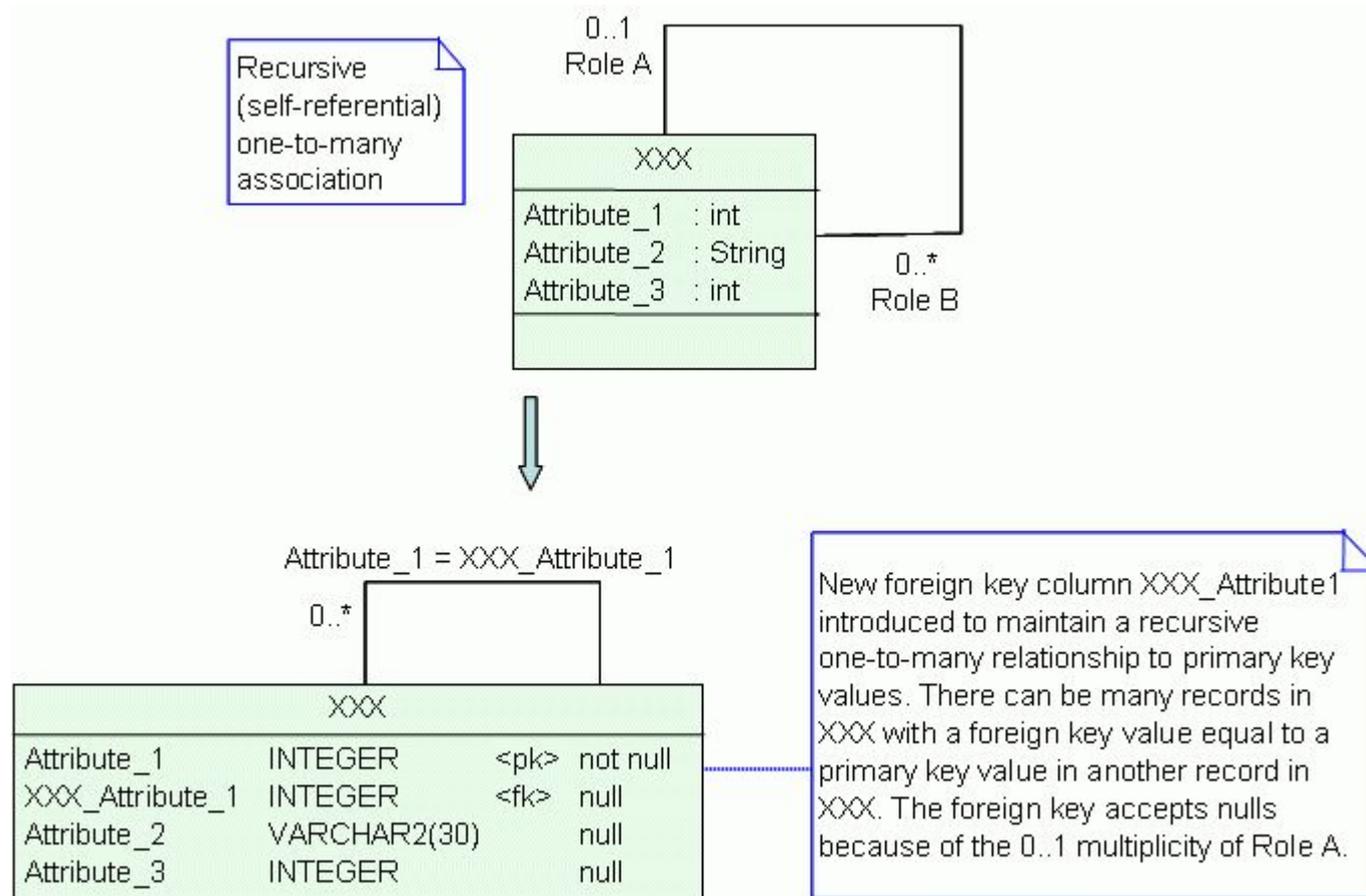
Отображение ассоциации «МНОГИЕ КО МНОГИМ»



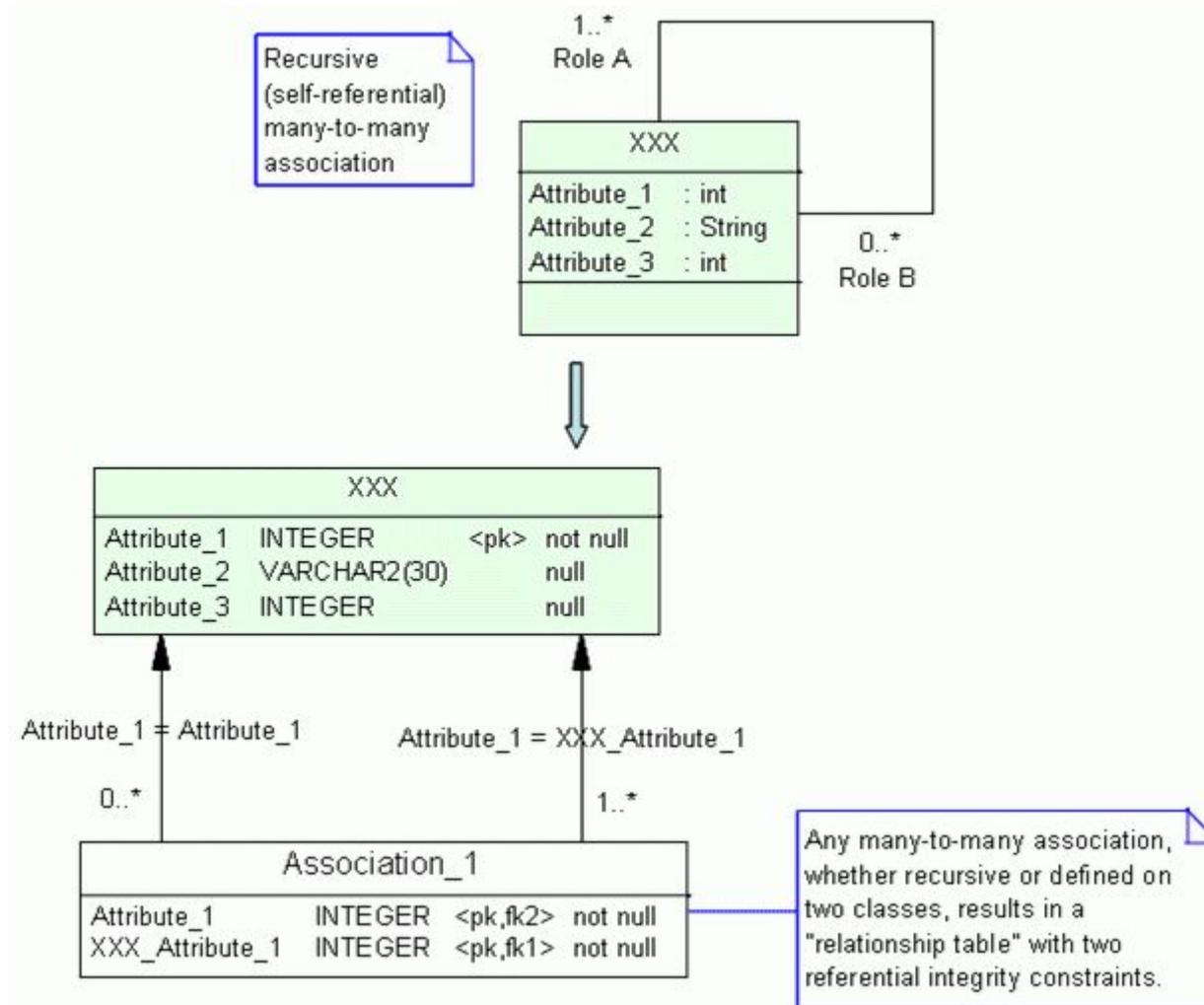
Отображение ассоциации «один к одному»



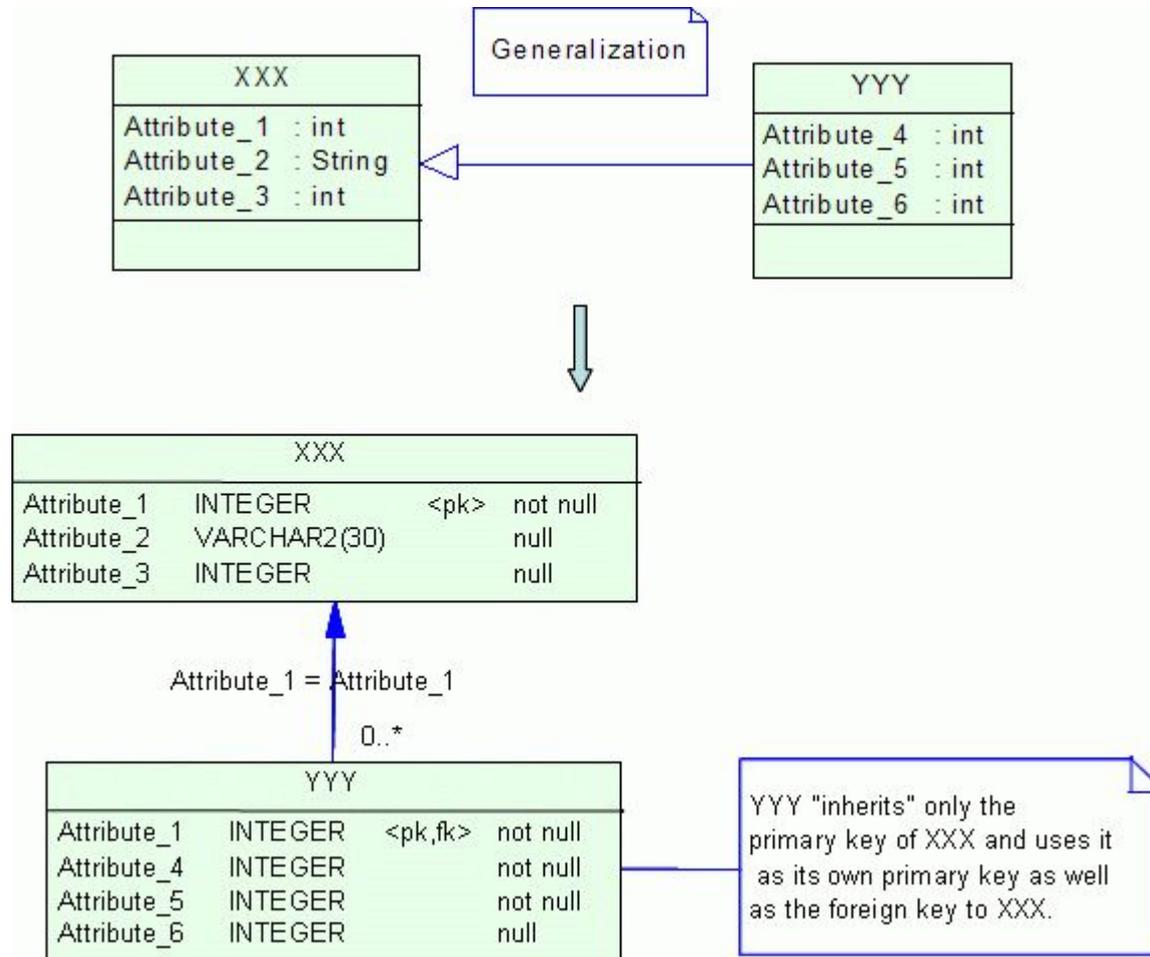
Отображение рекурсивной ассоциации «ОДИН КО МНОГИМ»



Отображение рекурсивной ассоциации «МНОГИЕ КО МНОГИМ»



Отображение обобщения



Моделирование

Проектирование и создание БД для управления электронной почтой

Итерация 1 из учебного примера EM предполагает, что БД существует заранее и загружена данными относительно **служащих, деловых партнеров и исходящих сообщений**.

Прикладная программа EM извлекает эту информацию из БД, готовит и посылает электронные сообщения и отмечает в БД какие исходящие сообщения уже были посланы.

Следующие итерации позволят помещать в БД исходящие сообщения и связанную с ними информацию из прикладной программы.

Диаграмма концептуальных классов для ЕМ

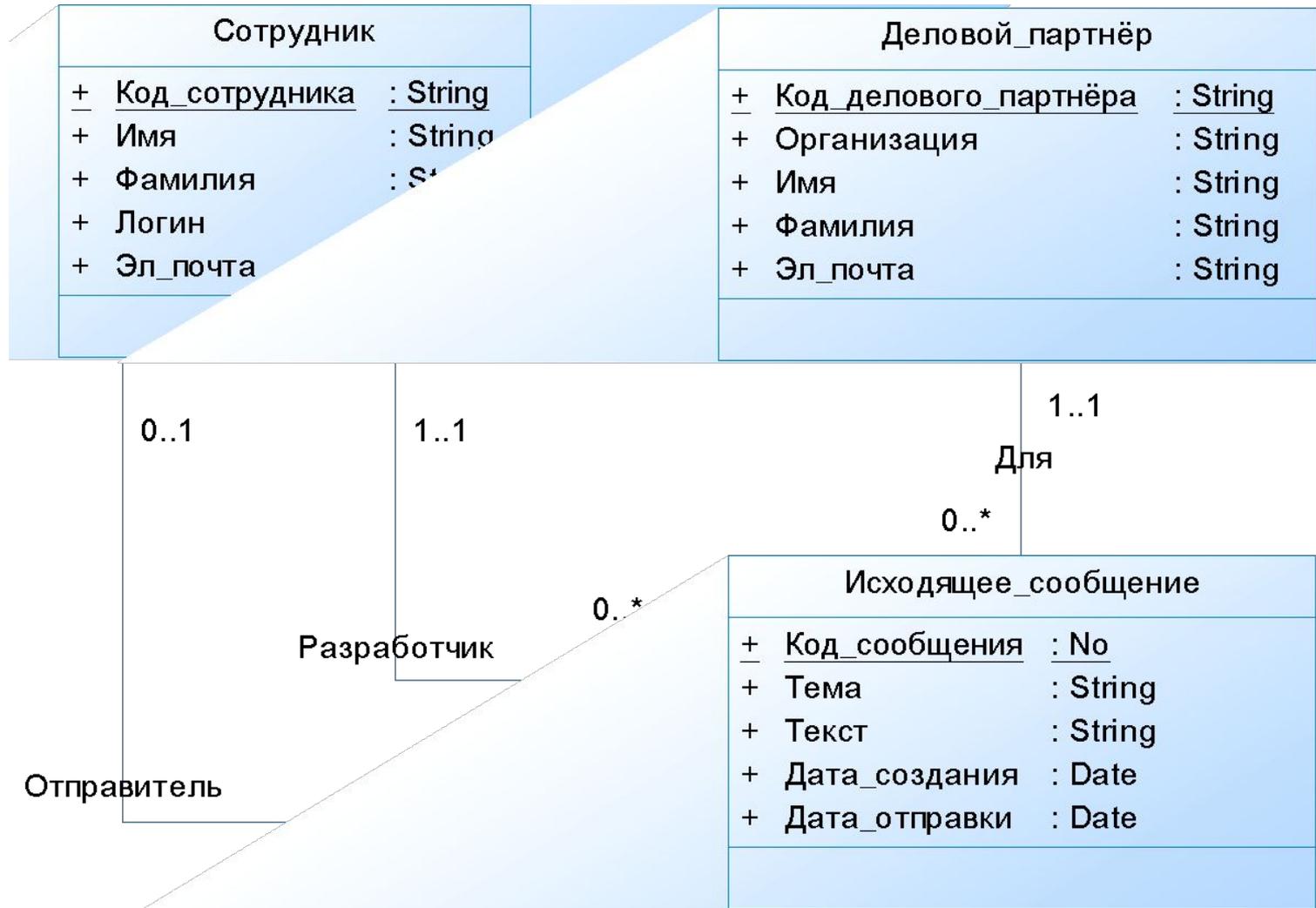
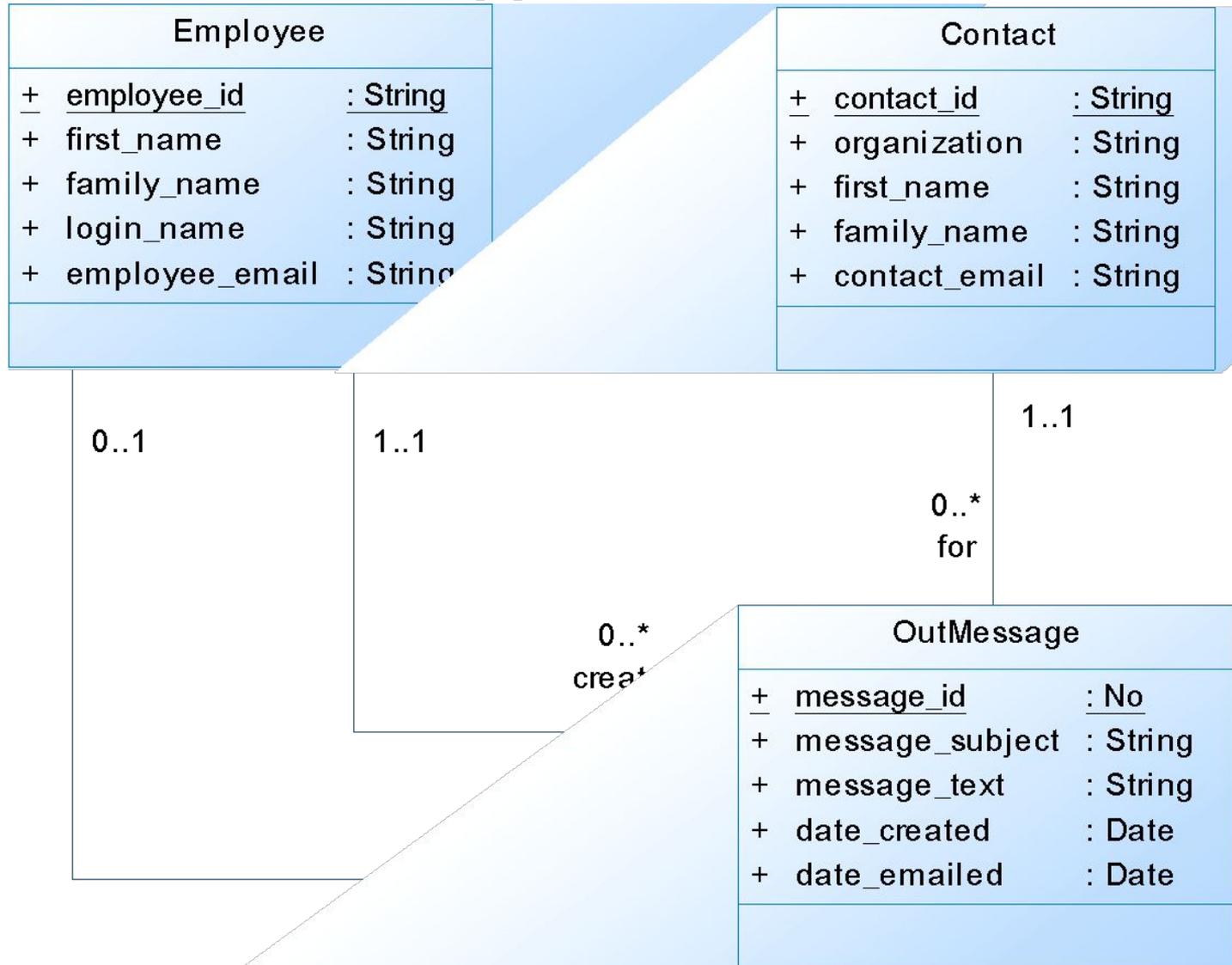
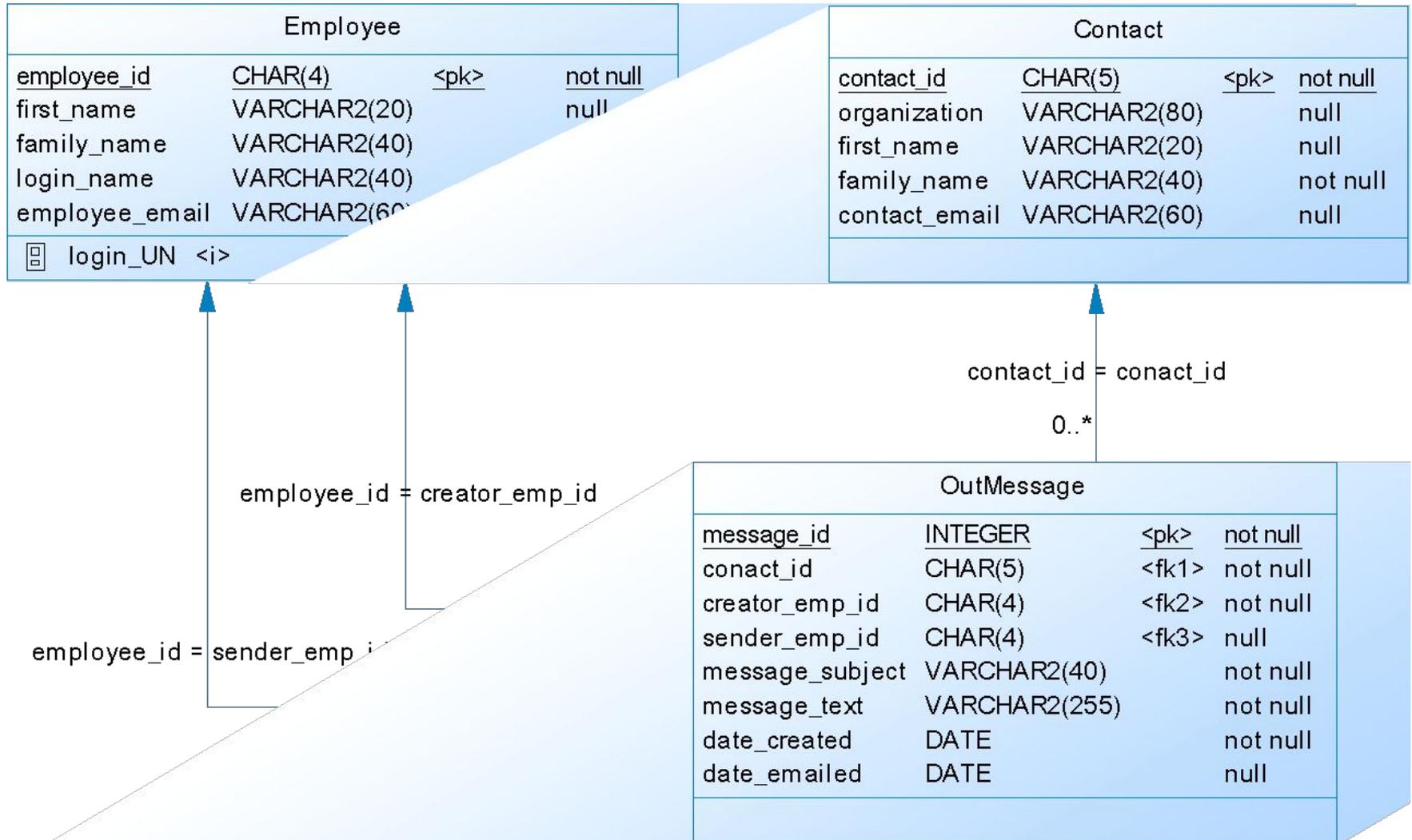


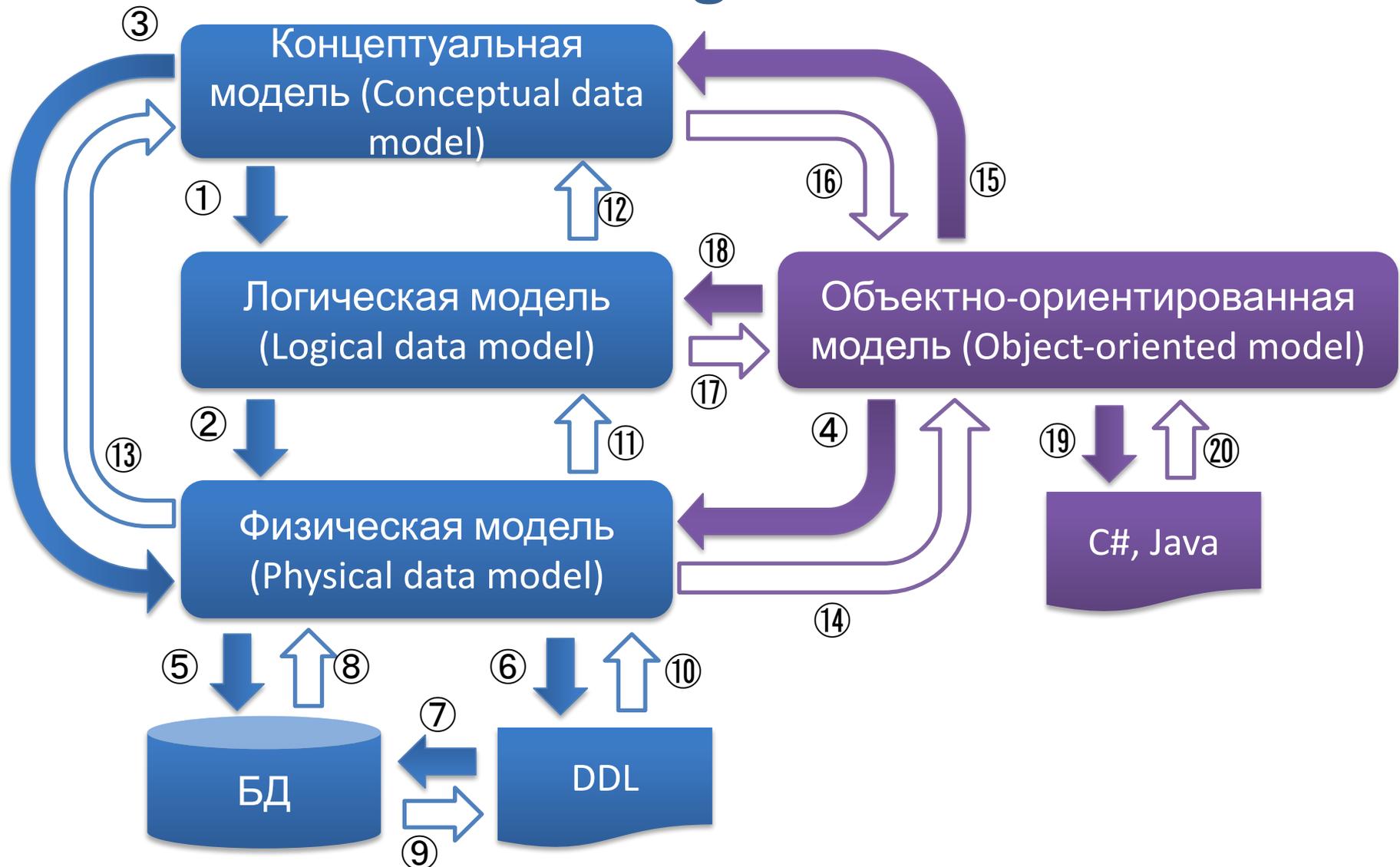
Диаграмма концептуальных классов для ЕМ



Физическая модель БД для ЕМ



Связи моделей данных в Power Designer



Проектирование ПО. Проектирование БД