

Базы данных, часть II

- Концептуальное проектирование баз данных
- Язык баз данных SQL
- Постреляционные модели данных
- Зачетная контрольная работа

(вопросы к зачету: Questions.doc в telegram
<https://t.me/+O2C1a9ctMy0zNjEy>)

Лектор: Морозов Сергей Вячеславович

Концептуальное проектирование баз данных

Виды проектирования БД



Концептуальное проектирование БД – процесс создания модели информации, не зависящий от любых физических аспектов ее представления

Логическое проектирование БД – процесс создания модели информации с учетом выбранной модели организации данных, но независимо от типа целевой СУБД и других физических аспектов реализации

Физическое проектирование БД – процесс описания реализации БД на ЗУ с указанием структур хранения и методов доступа, используемых для эффективной обработки

Концептуальное проектирование баз данных

Основные определения

Предметная область – часть реального мира, рассматриваемая в определенном смысле как единое целое, сведения о которой представляют собой информационные ресурсы в аспектах создания и использования БД.

Информационная модель – способ представления понятий или объектов предметной области, описывающий существенные для данного представления совокупности объектов, их параметры, поведение и отношения между ними.

CASE-система (Computer-Aided Software Engineering) – система поддержки технологий автоматизированного проектирования, реализации и сопровождения сложных программных систем на всех этапах их жизненного цикла.

Недостатки логического проектирования реляционных БД

1. Неудобство для проектировщиков:

- а) На ранних стадиях проектирования, как правило, требуется участие специалистов, хорошо знающих предметную область, но не владеющих теорией БД
- б) Во многих предметных областях трудно осуществлять моделирование информации на основе плоских таблиц

2. Отсутствие наглядности:

- а) Реляционная модель не предлагает какого-либо механизма для разделения объектов предметной области и связей между ними
- б) Реляционная модель не обеспечивает достаточных средств для представления семантики данных

3. Невозможность автоматизации процесса проектирования:

Реляционная модель не предоставляет какие-либо формализованные средства для представления функциональных и других зависимостей, на основе которых осуществляется процесс проектирования

Достоинства информационного моделирования

- Построение наглядной концептуальной схемы БД позволяет более полно оценить специфику моделируемой предметной области и избежать возможных ошибок на ранних стадиях проектирования.
- Информационная модель является важной документацией, полезной не только на стадии проектирования БД, но и при ее дальнейшей эксплуатации, сопровождении и развитии.
- На рынке представлены CASE-системы, обеспечивающие автоматизированное преобразование концептуальных схем (диаграмм) в реляционные (язык SQL).

Информационная модель Entity-Relationship

Альтернативные названия: ER-модель, модель «Сущность-Связь», диаграммы Чена, реляционная информационная модель

Предложена: 1976 г., Питер Чен

Назначение: описание моделей предметных областей с целью последующего проектирования БД

Стандарт: отсутствует

Применяемая нотация: графическая (диаграммы), множество альтернативных нотаций

Изучаемая нотация: применяется в CASE-системе Oracle

Основные понятия ER-модели

Сущность (тип сущности) – реальный или представляемый объект, информация о котором должна сохраняться и быть доступной

Атрибут сущности – любая деталь, которая служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности

Атрибут в реляционной модели: <имя атрибута, имя типа данных>
В ER-модели указание типа атрибута не является обязательным

ЧЕЛОВЕК

фио, например, Иванов И.И.

дата рождения, например,
30.06.1980

пол, например, М или Ж

Основные понятия ER-модели

Связь – графически изображаемая ассоциация, устанавливаемая между двумя сущностями.

Графическое отображение: ненаправленная линия



В ER-модели допускаются только бинарные связи, то есть, соединяющие две сущности или сущность саму с собою (рекурсивная связь)

Конец связи называют ролью связи в данной сущности

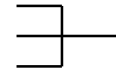
Характеристики роли:

- Имя роли (указывается над линией связи вблизи соответствующего конца)
- Степень роли (допустимое количество экземпляров соответствующей сущности в данной связи):

- Один экземпляр – одноточечный вход



- Много экземпляров – трехточечный вход

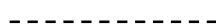


- Обязательность роли:

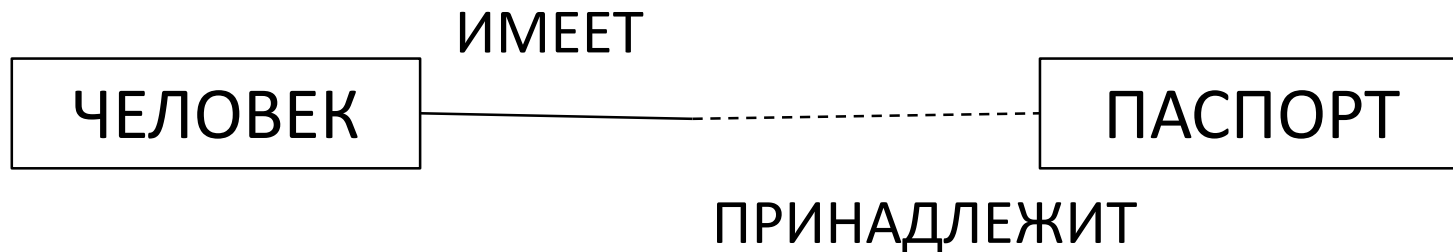
- Обязательная (каждый экземпляр данной сущности ДОЛЖЕН участвовать в связи)



- Необязательная (каждый экземпляр данной сущности МОЖЕТ участвовать в связи)



Связь «Один-к-Одному»



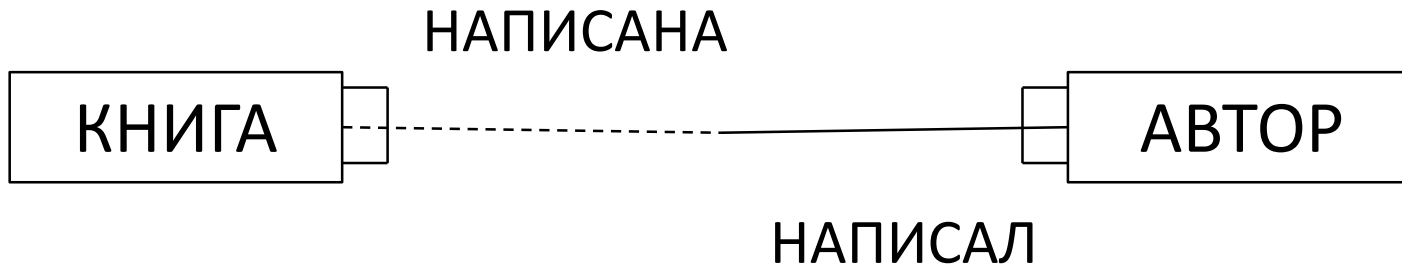
- Каждый человек имеет один и только один паспорт
- Каждый паспорт может принадлежать только одному человеку

Связь «Один-ко-многим»



- Каждый мужчина является сыном одного и только одного мужчины
- Каждый мужчина может являться отцом одного или более мужчин

Связь «Многие-ко-Многим»



- Каждая книга может быть написана одним или более авторами
- Каждый автор принимал участие в написании одной или более книг

Наследование в ER-модели

Тип сущности (A) может быть расщеплен на несколько взаимно исключающих подтипов ($B_1, B_2, \dots, B_n$)

Тип сущности, на основе которого определяются подтипы, называется супертипом

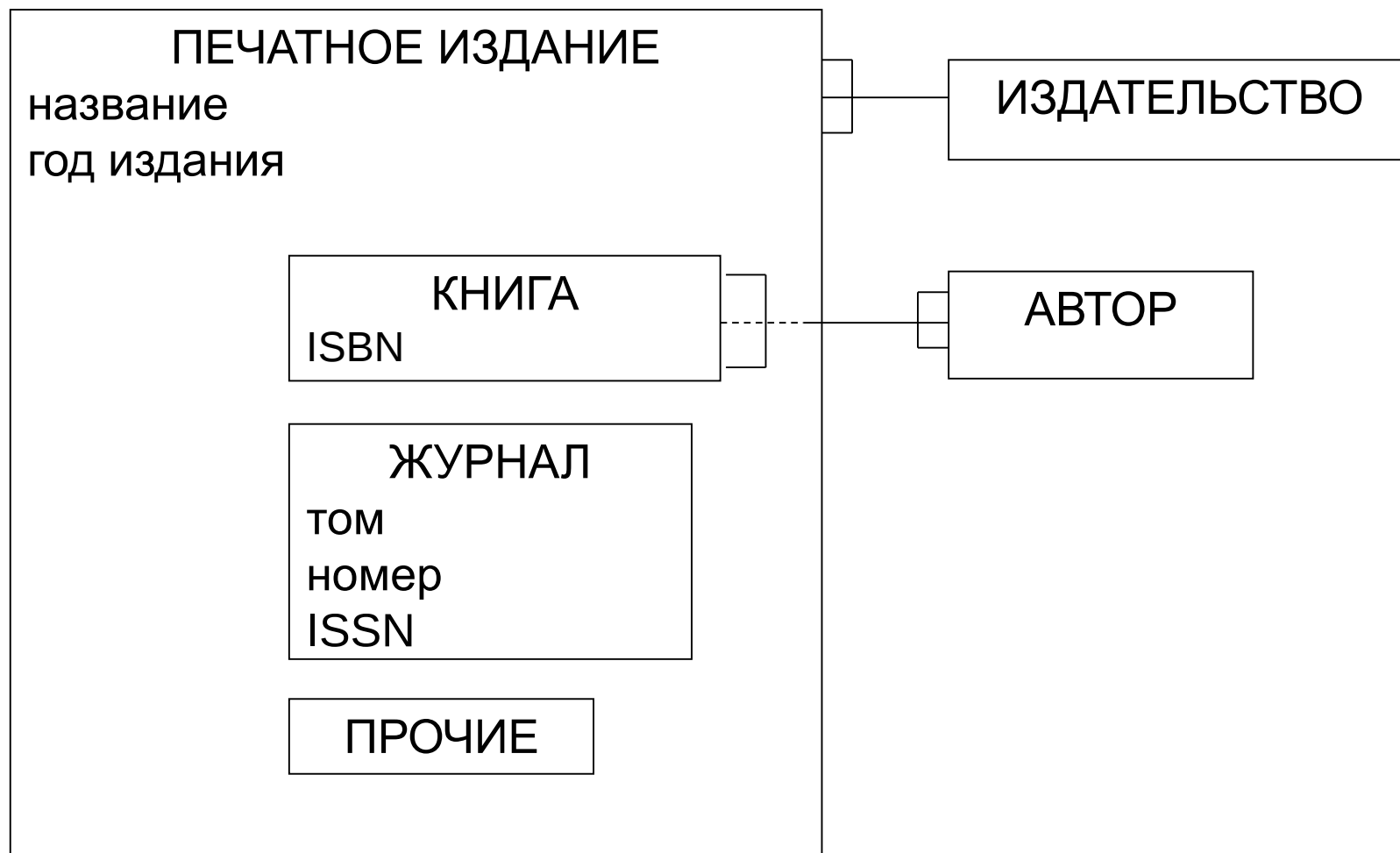
Подтипы наследуют атрибуты и связи супертипа и могут определять собственные атрибуты и/или связи

Простым типом сущности называется тип, не являющийся подтипом и не имеющий подтипов

Правила наследования в ER-модели:

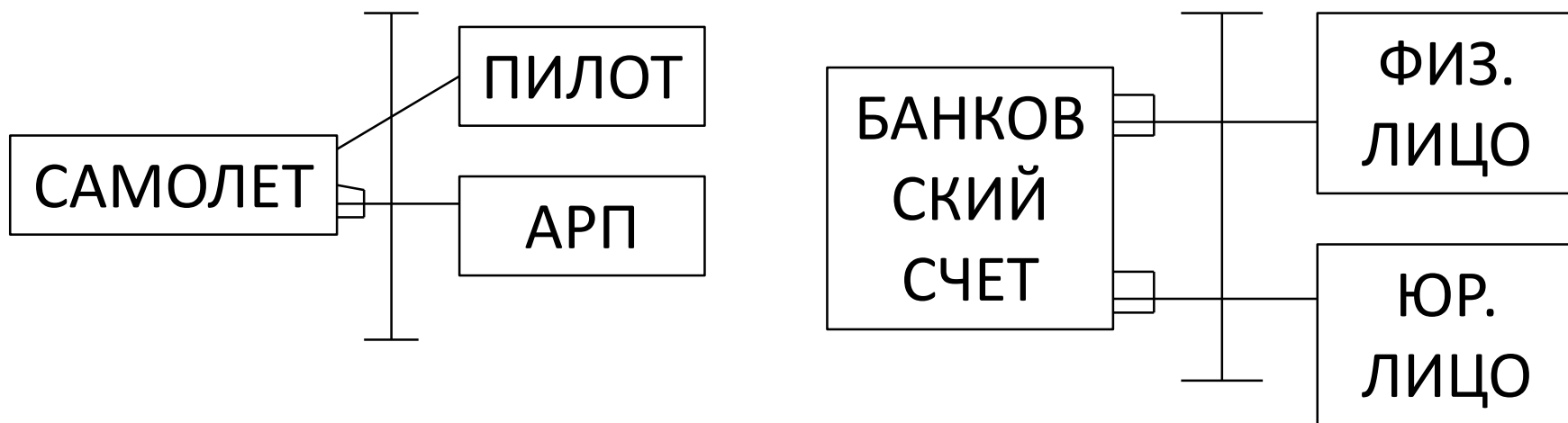
1. Включение ($\forall b \in B_i \rightarrow b \in A, i = 1, \dots, n$)
2. Отсутствие собственных экземпляров у супертипа ($\forall a \in A \rightarrow a \in B_i, i = 1, \dots, n$)
3. Разъединенность подтипов ($\forall b \in B_i \rightarrow b \notin B_j, i \neq j$)

Пример ER-модели с наследованием



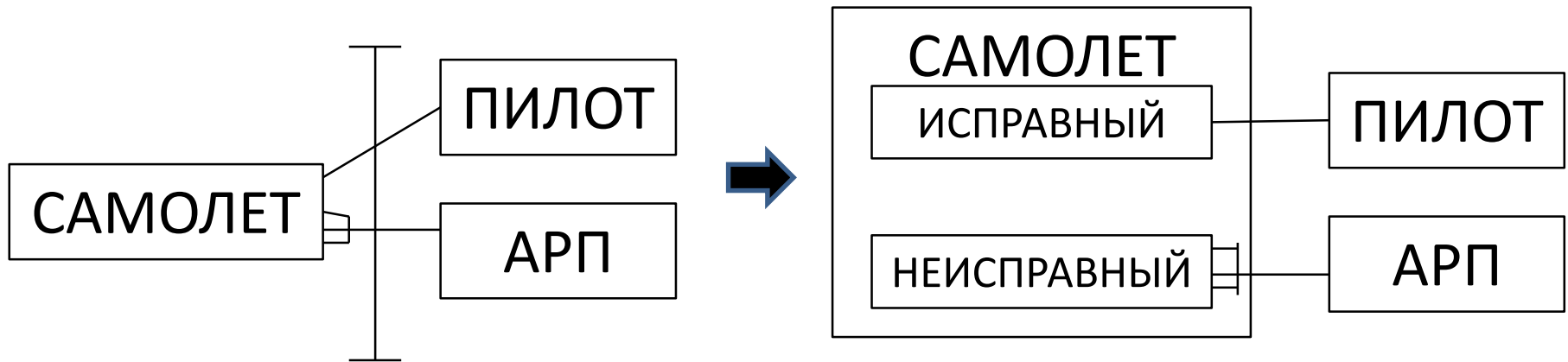
Взаимно исключающие связи

Взаимно исключающими связями называется такой набор связей одной сущности с другими, что для каждого экземпляра сущности может или должен существовать экземпляр только одной связи из данного набора

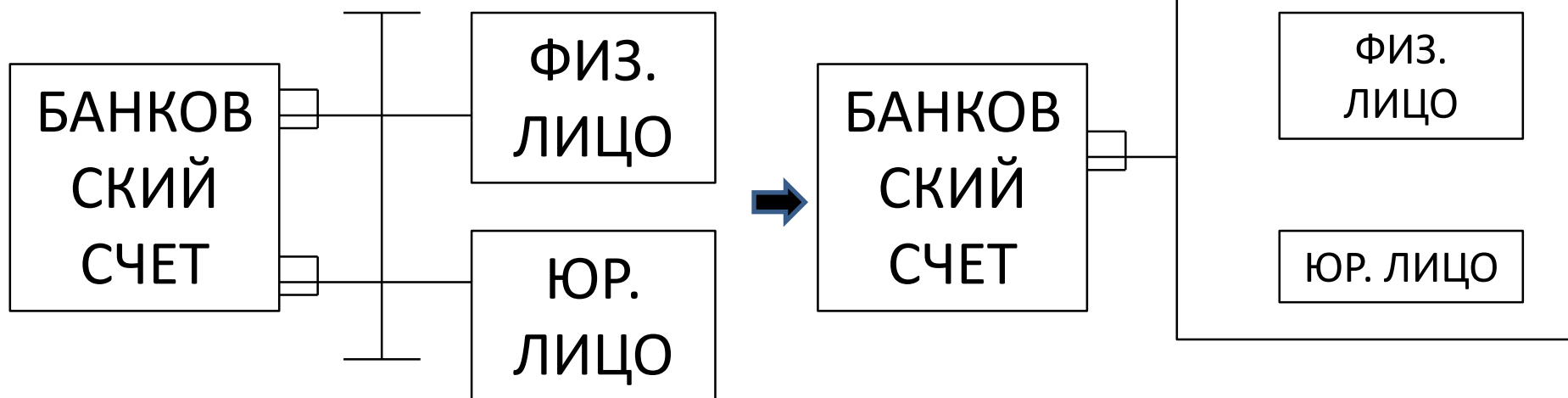


Преобразования диаграмм со взаимно исключающими связями

1. Введение подтипов



2. Введение общего супертипа



Уникальные идентификаторы экземпляров сущностей

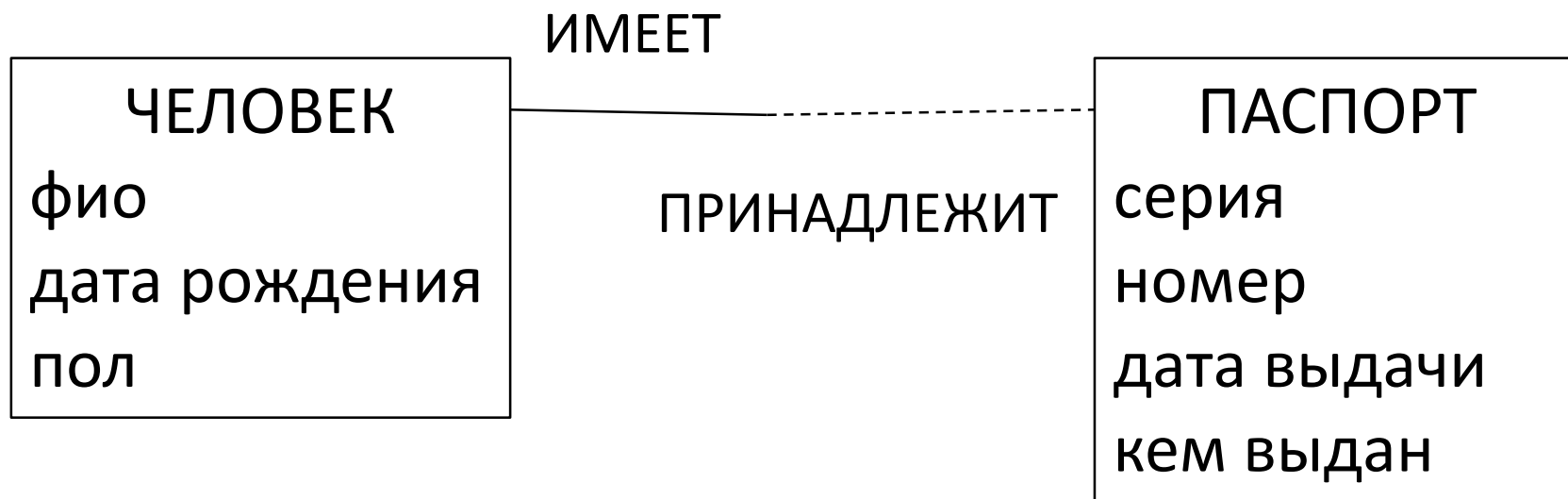
При определении сущности необходимо гарантировать, что каждый ее экземпляр является отличным от любого другого экземпляра той же сущности. Это достигается путем введения уникальных идентификаторов.

В ER-модели экземпляр сущности может идентифицироваться только своими индивидуальными характеристиками: значениями атрибутов и экземплярами связей.

В качестве уникального идентификатора сущности проектировщик может выбрать:

- Атрибут
- Комбинацию атрибутов
- Связь
- Комбинацию связей
- Комбинацию атрибутов и связей

Выбор уникального идентификатора экземпляров сущности «Человек»

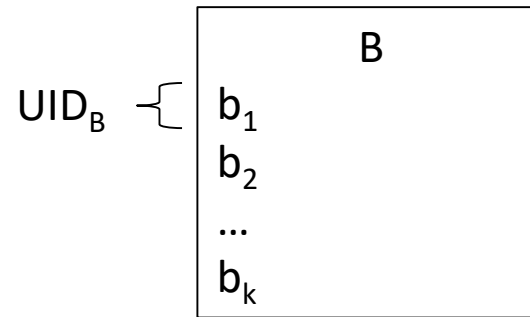
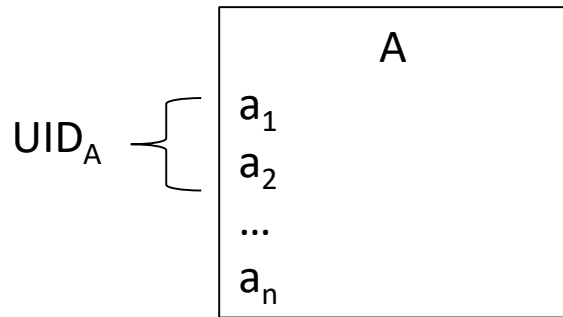


- Для БД предприятий: либо фио, либо фио + дата рождения
- Для БД крупных городов или государств: связь с паспортом, тогда в качестве УИД человека будет использоваться УИД его паспорта (серия + номер)

Алгоритм преобразования ER-модели в реляционную

1. Каждый простой тип сущности превращается в отношение. Экземплярам сущности соответствуют кортежи данного отношения.
2. Каждый атрибут сущности становится атрибутом соответствующего отношения, при этом выбирается тип для представления соответствующих данных.
3. Компоненты уникального идентификатора сущности становятся первичным ключом отношения. Если в состав уникального идентификатора входят связи, к числу атрибутов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (копия первичного ключа соответствующего отношения).

Алгоритм преобразования ER-модели в реляционную



A

a ₁	a ₂	...	a _n

PRIMARY KEY (a₁, a₂)

B

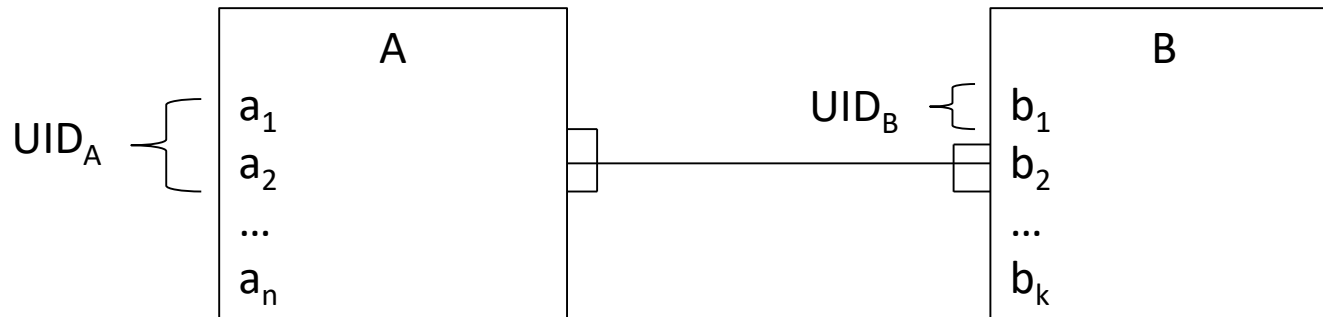
b ₁	b ₂	...	b _k

PRIMARY KEY (b₁)

Алгоритм преобразования ER-модели в реляционную

4. Связи становятся внешними ключами отношения; для связей «один-ко-многим» внешний ключ объявляется в отношении, соответствующем сущности на конце связи «многие»; для связей «один-к-одному» внешний ключ может быть объявлен в одном из двух отношений по желанию проектировщика, при этом данный внешний ключ должен быть определен и как возможный ключ отношения (для ограничения степени связи); если связь является необязательной, внешний ключ допускает наличие неопределенных значений.
5. Для поддержки связи «многие-ко-многим» создается дополнительное отношение с двумя (возможно составными) атрибутами, каждый из которых содержит копию первичного ключа соответствующего отношения, участвующего в данной связи.

Алгоритм преобразования ER-модели в реляционную



A

a_1	a_2	...	a_n	b_1

PRIMARY KEY (a_1, a_2)

FOREIGN KEY (b_1) UNIQUE (b_1)

B

b_1	b_2	...	b_k

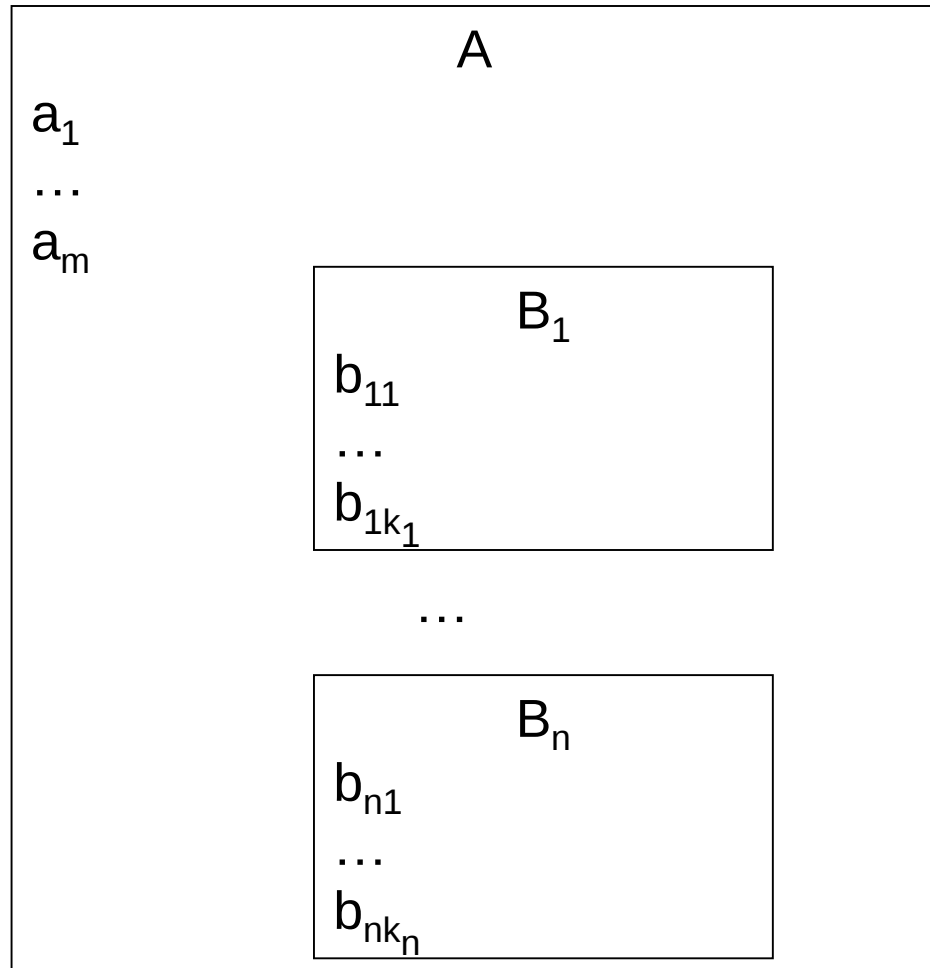
PRIMARY KEY (b_1)

AB

a_1	a_2	b_1

Алгоритм преобразования ER-модели в реляционную

6. Подтипы и супертипы могут быть представлены в реляционной модели двумя способами



Алгоритм преобразования ER-модели в реляционную

ба. Одно отношение для всех подтипов

t_c	a_1	...	a_m	b_{11}	...	b_{1k_1}	...	b_{n1}	...	b_{nk_n}
B_1	X	X	X	X	X	X	N	N	N	N
B_n	X	X	X	N	N	N	N	X	X	X

t_c – код типа (часть первичного ключа)

X – конкретное значение атрибута

N – неопределенное значение (NULL)

Доступ к экземплярам подтипов:

PROJECT A { $a_1, \dots, a_m, b_{i1}, \dots, b_{ik}$ } (A WHERE $t_c = 'B_i'$)

Достоинства:

1. Соответствие логике супертипов
2. Простой способ доступа к экземплярам супертипов и не слишком сложный – к экземплярам подтипов
3. Сокращение количества отношений

Недостатки:

1. Единственное отношение – узкое место при многопользовательском доступе
2. Усложнение логики приложений БД
3. Непроизводительный расход внешней памяти (хранение NULL)

Алгоритм преобразования ER-модели в реляционную

66. Каждый подтип представляется отдельным отношением

Diagram illustrating a matrix B_1 with columns $a_1, \dots, a_m, b_{11}, \dots, b_{1k_1}$. The matrix is partitioned into blocks of varying shades of gray, representing different types of elements or structures.

...

The diagram shows a block matrix B_n with $m + k_n$ columns. The first m columns are labeled $a_1, \dots, a_m$ and the next k_n columns are labeled $b_{n1}, \dots, b_{nk_n}$. The matrix is partitioned into three rows of blocks. The top row has blocks of size $1 \times m$ and $1 \times k_n$. The middle row has blocks of size $1 \times m$ and $1 \times k_n$. The bottom row has blocks of size $1 \times m$ and $1 \times k_n$. The blocks are represented by rectangles of varying shades of gray.

Воссоздание супертипа:

PROJECT $B_1 \{a_1, \dots, a_m\}$ UNION ... UNION PROJECT $B_n \{a_1, \dots, a_m\}$

Достоинства:

1. Более понятные правила работы с подтипами
2. Упрощение логики приложений

Недостатки:

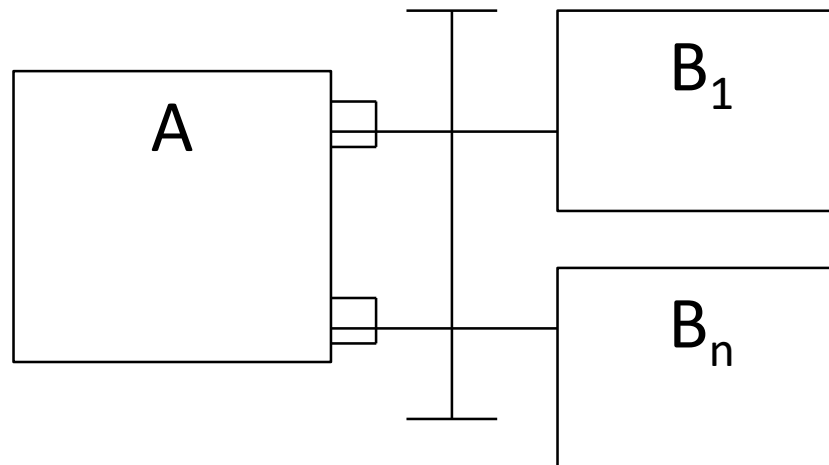
1. Увеличение количества отношений
2. Усложнение доступа к экземплярам супертипа
3. В общем случае невозможность модификации экземпляров супертипа

Алгоритм преобразования ER-модели в реляционную

7. Представление взаимно исключающих связей:

7а. Преобразование модели со взаимно исключающими связями в модель с наследованием (далее см. пункт 6)

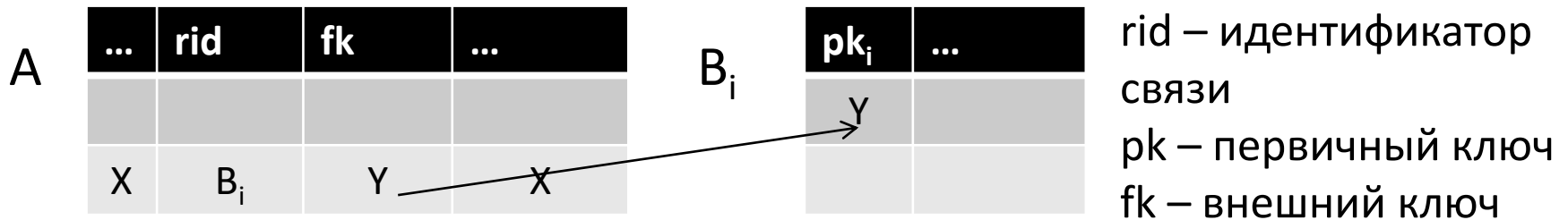
Для моделей, где связи «один-ко-многим» являются взаимоисключающими со стороны сущности со степенью роли «многие», возможно применение еще двух способов преобразования



Алгоритм преобразования ER-модели в реляционную

7. Представление взаимно исключающих связей:

7б. Общее хранение внешних ключей



Достоинства:

- Добавление небольшого количества атрибутов для представления связей

Недостатки:

- Применим при условии, что все первичные ключи сущностей B определены на одном домене;
- Сложная операция соединения: (A WHERE rid='B_i') JOIN B_i WHERE A.fk=B_i.pk_i

Алгоритм преобразования ER-модели в реляционную

7. Представление взаимно исключающих связей:

7в. Раздельное хранение внешних ключей

A

...	pk ₁	...	pk _n	...
X	Y	NULL	NULL	X

B₁

pk ₁	...
Y	

Достоинства:

- Применим независимо от доменов первичных ключей в B;
- Операция естественного соединения: A NATURAL JOIN B_i

Недостатки:

- Увеличение количества атрибутов для представления связей;
- Непроизводительный расход внешней памяти (хранение NULL)