Подходы к организации баз данных

Иерархические базы данных

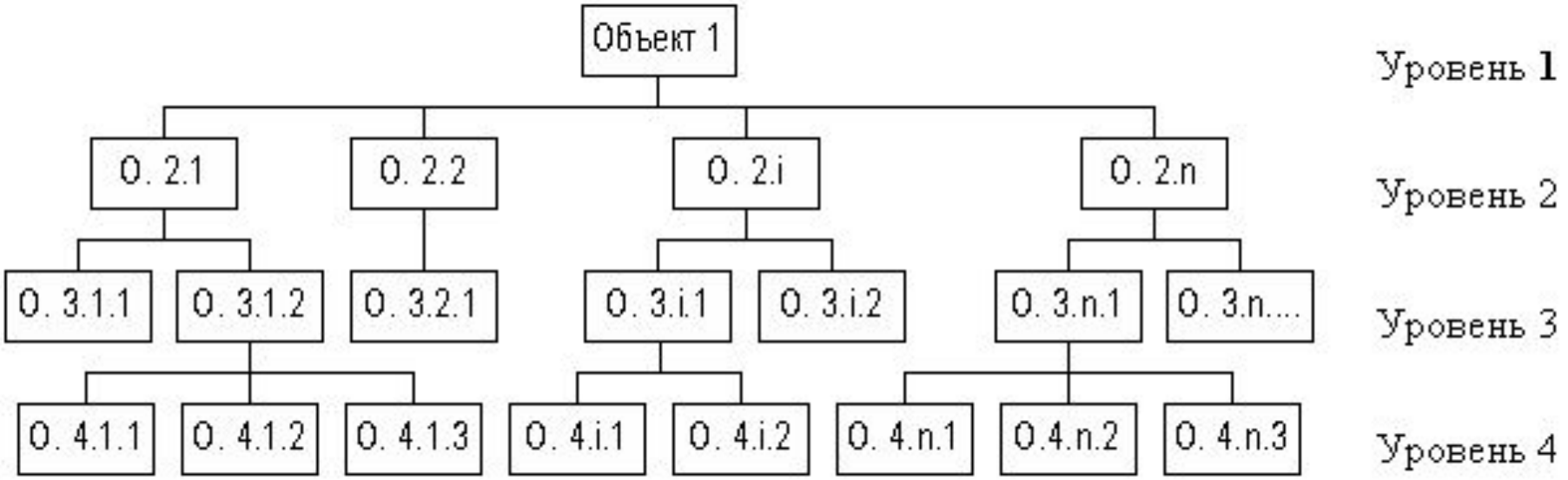


Рис. 1 Схема иерархической модели данных

Подходы к организации баз данных

Сетевые базы данных

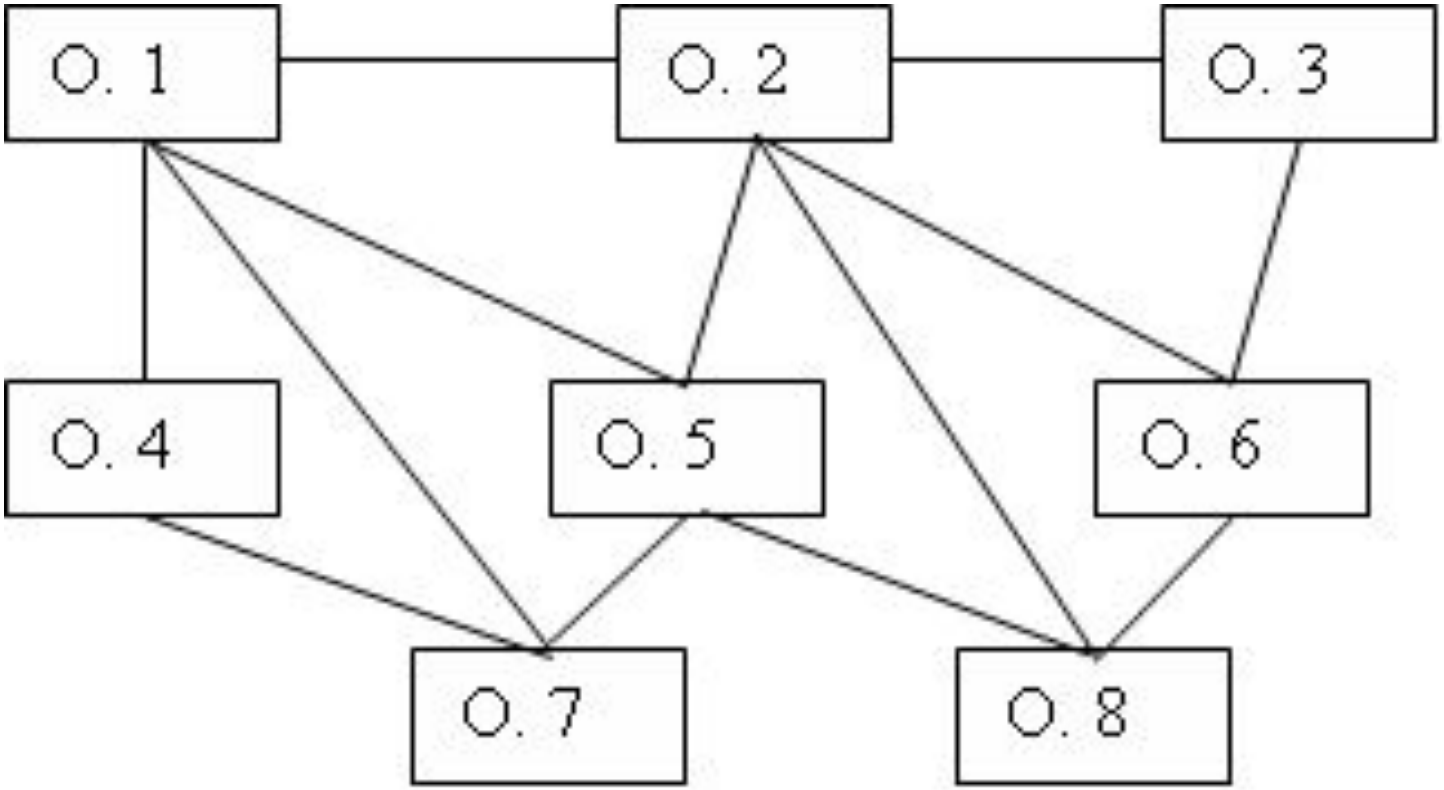


Рис. 2 Схема сетевой модели

Введение в реляционную модель данных

Основные понятия реляционной модели данных

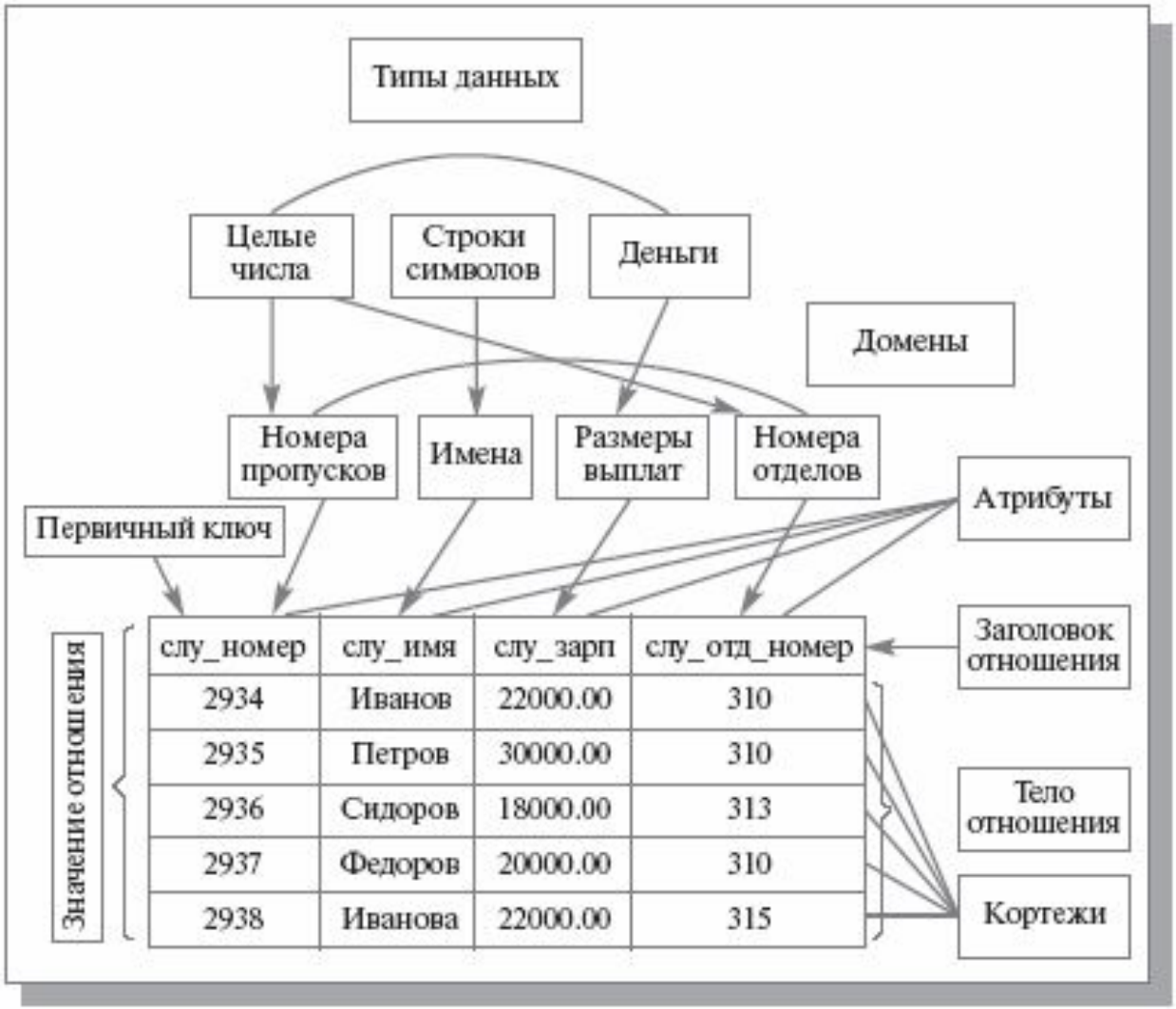


Рис. 3 Соотношение основных понятий реляционного подход

Введение в реляционную модель данных

Основные понятия реляционной модели данных

Атомарность значений атрибутов, первая нормальная форма отношения

НОМЕР_ОТДЕЛА	ОТДЕЛ		
	СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП
310	2934	Иванов	22000.00
	2935	Петров	30000.00
	2937	Федоров	20000.00
313	2936	Сидоров	18000.00
315	2938	Иванова	22000.00

Рис. 4 Ненормализованное отношение ОТДЕЛЫ-СЛУЖАЩИЕ

Введение в реляционную модель данных

Основные понятия реляционной модели данных

Атомарность значений атрибутов, первая нормальная форма отношения

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315

Рис. 5 Отношение СЛУЖАЩИЕ: нормализованный вариант отношения ОТДЕЛЫ-СЛУЖАЩИЕ

Трехзначная логика (3VL)

Трехзначная логика (3VL)

Таблица 1. Таблица истинности AND

AND	F	T	U
F	F	F	F
T	F	T	U
U	F	U	U

Таблица 2. Таблица истинности OR

OR	F	T	U
F	**F**	**T**	**U**
T	**T**	**T**	**T**
U	**U**	**T**	**U**

Таблица 3. Таблица истинности NOT

NOT	
F	T
T	F
U	U

Трехзначная логика (3VL)

Имеется несколько парадоксальных следствий применения трехзначной логики.

Парадокс 1.
Null-значение не равно самому себе.
Действительно, выражение
null = null дает значение не ИСТИНА, а НЕИЗВЕСТНО.

Парадокс 2.
Неверно также, что null-значение не равно самому себе!
Действительно, выражение null <> null также принимает значение не ИСТИНА, а НЕИЗВЕСТНО!

Парадокс 3.
a or (not(a)) не обязательно ИСТИНА.

И т.п.

Потенциальные ключи

Таблица 4. Отношение "Сотрудники"

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 5.

A	B	C
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Внешние ключи

Таблица 6. Отношение "Поставщики и поставляемые детали"

Номер поставщика	Наименование поставщика	Номер детали	Наименование детали	Поставляемое количество
1	Иванов	1	Болт	100
1	Иванов	2	Гайка	200
1	Иванов	3	Винт	300
2	Петров	1	Болт	150
2	Петров	2	Гайка	250
3	Сидоров	3	Винт	1000

Внешние ключи

Таблица 7. Отношение "Поставщики"

Номер поставщика	Наименование поставщика
1	Иванов
2	Петров
3	Сидоров

Таблица 8. Отношение "Детали"

Номер детали	Наименование детали
1	Болт
2	Гайка
3	Винт

Таблица 9. Отношение "Поставки"

Номер поставщика	Номер детали	Поставляемое количество
1	1	100
1	2	200
1	3	300
2	1	150
2	2	250
3	3	1000

Стратегии поддержания ссылочной целостности

Основные:

RESTRICT (ОГРАНИЧИТЬ)
CASCADE (КАСКАДИРОВАТЬ)

Дополнительные:

SET NULL (УСТАНОВИТЬ В NULL)
SET DEFAULT (УСТАНОВИТЬ ПО УМОЛЧАНИЮ)
IGNORE (ИГНОРИРОВАТЬ)

Технологии проектирования реляционных БД

Этапы разработки базы данных

Уровни моделирования:

- Сама предметная область
- Модель предметной области
- Логическая модель данных
- Физическая модель данных
- Собственно база данных и приложения

Технологии проектирования реляционных БД

Критерии оценки качества логической модели данных

- Адекватность базы данных предметной области
- Легкость разработки и сопровождения базы данных
- Скорость выполнения операций обновления данных (вставка, обновление, удаление кортежей)
- Скорость выполнения операций выборки данных

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

При проектировании базы данных решаются две основные проблемы:

- Каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не противоречило семантике предметной области и было, по возможности, лучшим (эффективным, удобным и т. д.)?

(*Проблема логического проектирования баз данных).*

- Как обеспечить эффективность выполнения запросов к базе данных, т. е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создания каких дополнительных структур (например, индексов) потребовать и т. д.?

(Проблема физического проектирования баз данных).

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- **первая нормальная форма (1NF);**
- **вторая нормальная форма (2NF);**
- **третья нормальная форма (3NF);**
- **нормальная форма Бойса-Кодда (BCNF);**
- **четвертая нормальная форма (4NF);**
- **пятая нормальная форма, или нормальная форма проекции-соединения (5NF или PJ/NF).**

Основные свойства нормальных форм состоят в следующем:

- **каждая следующая нормальная форма в некотором смысле лучше предыдущей нормальной формы;**

- **при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.**

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Декомпозиция без потерь и функциональные зависимости

Определение: **Функциональная зависимость**

В отношении ***r*** атрибут ***Y*** функционально зависит от атрибута ***X*** (*X* и *Y* могут быть составными) в том и только в том случае, если каждому значению ***X*** соответствует в точности одно значение ***Y***:

r.X -> r.Y.

Определение: **Минимальная (полная) функциональная зависимость**

Функциональная зависимость ***r.X -> r.Y*** называется минимальной (или полной), если атрибут ***Y*** не зависит функционально от любого точного подмножества ***X***.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Декомпозиция без потерь и функциональные зависимости

Определение: **Транзитивная функциональная зависимость**

Функциональная зависимость ***r.X -> r.Y*** называется транзитивной, если существует такой атрибут *Z*, что имеются функциональные зависимости

r.X -> r.Z и r.Z -> r.Y

и отсутствует функциональная зависимость ***r.Z -> r.X***.

(При отсутствии последнего требования мы имели бы "неинтересные" транзитивные зависимости в любом отношении, обладающем несколькими ключами.)

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Декомпозиция без потерь и функциональные зависимости

Определение: **Неключевой атрибут**

Неключевым атрибутом называется любой атрибут отношения, не входящий в состав ключа (в частности, первичного).

Определение: **Взаимно независимые атрибуты**

Два или более атрибута **взаимно независимы**, если ни один из этих атрибутов не является функционально зависимым от других.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Декомпозиция без потерь и функциональные зависимости

Декомпозиция отношения – разбиение путем проецирования

Правило:

Считаются правильными такие декомпозиции отношения, которые обратимы, т. е. имеется возможность собрать исходное отношение из декомпозированных отношений без потери информации. Такие декомпозиции называются ***декомпозициями без потерь***.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Корректные и некорректные декомпозиции отношений. Теорема Хеза.

СЛУЖАЩИЕ_ПРОЕКТЫ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
2934	Иванов	22000.00	1	Иванов
2941	Иваненко	22000.00	2	Иваненко

Декомпозиция (1). Отношения **СЛУЖ** и **СЛУ_ПРО**

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП
2934	Иванов	22000.00
2941	Иваненко	22000.00

СЛУ_НОМ	ПРО_НОМ	ПРОЕКТ_РУК
2934	1	Иванов
2941	2	Иваненко

Декомпозиция (2). Отношения **СЛУЖ** и **ЗАРП_ПРО**

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП
2934	Иванов	22000.00
2941	Иваненко	22000.00

СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
22000.00	1	Иванов
22000.00	2	Иваненко

Рис. 6. Две возможные декомпозиции отношения СЛУЖАЩИЕ_ПРОЕКТЫ

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Корректные и некорректные декомпозиции отношений. Теорема Хеза.

СЛУ_НОМ	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК
2934	Иванов	22000.00	1	Иванов
2941	Иваненко	22000.00	2	Иваненко
2934	Иванов	22000.00	2	Иваненко
2941	Иваненко	22000.00	1	Иванов

Рис. 7. Результат естественного соединения отношений СЛУЖ и ЗАРП_ПРО

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Корректные и некорректные декомпозиции отношений. Теорема Хеза.

Теорема Хеза.

Пусть задано отношение **r {A, B, C}** (A, B и C, в общем случае, являются составными атрибутами) и выполняется ***FD*** **A -> B**

Тогда:

r = (r PROJECT {A, B}) NATURAL JOIN (r PROJECT {A, C}).

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Корректные и некорректные декомпозиции отношений. Теорема Хеза.

СЛУЖАЩИЕ_ОТДЕЛЫ_ПРОЕКТЫ

СЛУ_НОМ	СЛУ_ОТД	ПРО_НОМ
2934	630	1
2941	631	1
2934	630	2
2941	631	2

Декомпозиция.
Отношения СЛУЖ_ОТДЕЛЫ и СЛУЖ_ПРОЕКТЫ

СЛУ_НОМ	СЛУ_ОТД
2934	630
2941	631

СЛУ_НОМ	ПРО_НОМ
2934	1
2941	1
2934	2
2941	2

Рис. 8. Декомпозиция без потерь по теореме Хеза

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Первая нормальная форма

Определение: **Первая нормальная форма**

Переменная отношения находится ***в первой нормальной форме***, если обладает следующими свойствами:

- в отношении нет одинаковых кортежей.
- кортежи не упорядочены.
- атрибуты не упорядочены.
- все значения атрибутов атомарны

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

СЛУ_НОМ	СЛУ_УРОВ	СЛУ_ЗАРП	ПРО_НОМ	СЛУ_ЗАДАН
2934	2	22400.00	1	A
2935	3	29600.00	1	B
2936	1	20000.00	1	C
2937	1	20000.00	1	D
2934	2	22400.00	2	D
2935	3	29600.00	2	C
2936	1	20000.00	2	B
2937	1	20000.00	2	A

Рис. 11. Возможное значение переменной отношения СЛУЖАЩИЕ_ПРОЕКТЫ_ЗАДАНИ

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

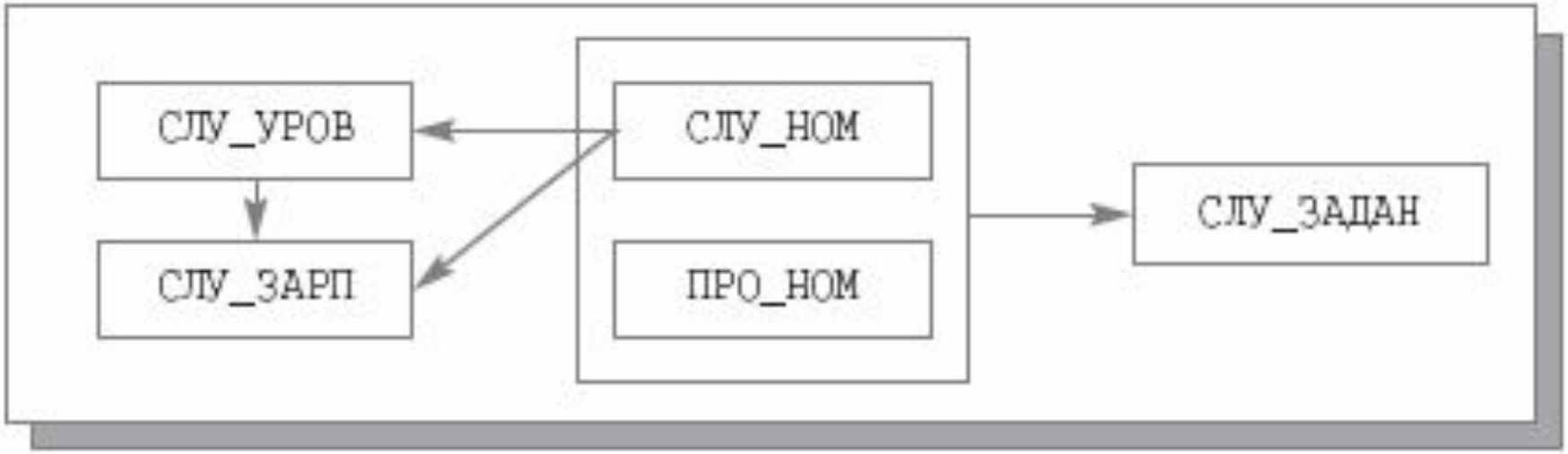


Рис. 10. Диаграмма множества FD отношения СЛУЖАЩИЕ_ПРОЕКТЫ_ЗАДАНИЯ

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Аномалии обновления, возникающие из-за наличия неминимальных функциональных зависимостей
(на примере отношения *СЛУЖАЩИЕ_ПРОЕКТЫ_ЗАДАНИЯ*)

- **Добавление кортежей**. Мы не можем дополнить отношение ***СЛУЖАЩИЕ_ПРОЕКТЫ_ЗАДАНИЯ*** данными о служащем, который в данное время еще не участвует ни в одном проекте (***ПРО_НОМ*** является частью первичного ключа и не может содержать неопределенных значений).

- **Удаление кортежей.** Мы не можем сохранить в отношении ***СЛУЖАЩИЕ_ПРОЕКТЫ_ЗАДАНИЯ*** данные о служащем, завершившем участие в своем последнем проекте (по той причине, что значение атрибута ***ПРО_НОМ*** для этого служащего становится неопределенным).

- **Модификация кортежей.** Чтобы изменить разряд служащего, мы будем вынуждены модифицировать все кортежи с соответствующим значением атрибута ***СЛУ_НОМ***.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Возможная декомпозиция

Рис. 12 Диаграммы FD в переменных отношений СЛУЖ и СЛУЖ_ПРО_ЗАДАН

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Значение переменной отношения СЛУЖ

СЛУ_НОМ	СЛУ_УРОВ	СЛУ_ЗАРП
2934	2	22400.00
2935	3	29600.00
2936	1	20000.00
2937	1	20000.00

Значение переменной отношения СЛУЖ_ПРО_ЗАДАН

СЛУ_НОМ	ПРО_НОМ	СЛУ_ЗАДАН
2934	1	A
2935	1	B
2936	1	C
2937	1	D
2934	2	D
2935	2	C
2936	2	B
937	2	A

Рис. 13. Значения переменных отношений

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Вторая нормальная форма

Определение: **Вторая нормальная форма**

Переменная отношения находится во ***второй нормальной форме (2NF)*** тогда и только тогда:

- когда она находится в первой нормальной форме, и
- каждый неключевой атрибут минимально функционально зависит от первичного ключа.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Аномалии обновлений, возникающие из-за наличия транзитивных функциональных зависимостей (на примере отношения ***СЛУЖ***)

Добавление кортежей. Невозможно сохранить данные о новом разряде (и соответствующем ему размере зарплаты), пока не появится служащий с новым разрядом. (Первичный ключ не может содержать неопределенные значения.)

Удаление кортежей. При увольнении последнего служащего с данным разрядом мы утратим информацию о наличии такого разряда и соответствующем размере зарплаты.

Модификация кортежей. При изменении размера зарплаты, соответствующей некоторому разряду, мы будем вынуждены изменить значение атрибута ***СЛУ_ЗАРП*** в кортежах всех служащих, которым назначен этот разряд (иначе не будет выполняться **FD** ***СЛУ_УРОВ ->СЛУ_ЗАРП***).

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Возможная декомпозиция

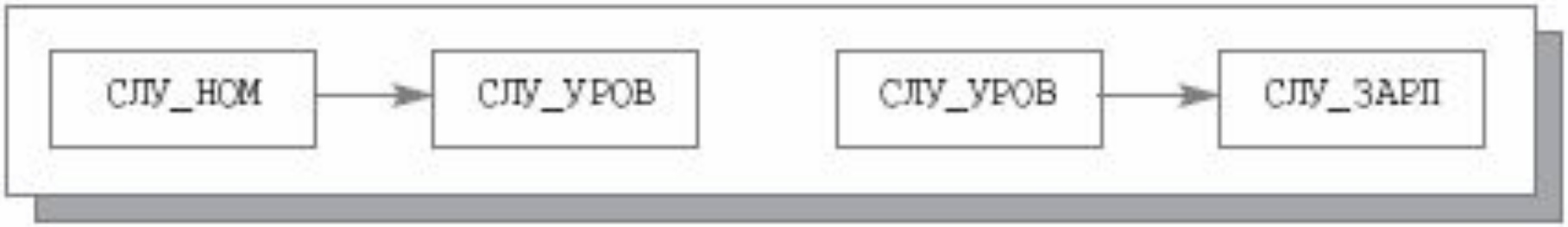


Рис. 14 Диаграммы FD в отношениях СЛУЖ1 и УРОВ

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Третья нормальная форма

Определение: **Третья нормальная форма**

Переменная отношения находится в ***третьей нормальной форме (3NF)*** в том и только в том случае, когда она

- находится во второй нормальной форме, и
- каждый неключевой атрибут нетранзитивно функционально зависит от первичного ключа.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Независимые проекции отношений. Теорема Риссанена

Рис. Варианты проекций отношения

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Независимые проекции отношений. Теорема Риссанена

Теорема Риссанена

Проекции ***r1*** и ***r2*** отношения ***r*** являются ***независимыми*** тогда и только тогда, когда:

- каждая ***FD*** в отношении ***r*** логически следует из ***FD*** в ***r1*** и ***r2***;
- общие атрибуты ***r1*** и ***r2*** образуют возможный ключ хотя бы для одного из этих отношений.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Определение

Атомарным отношением называется отношение, которое невозможно декомпозировать на независимые проекции.

Например,

отношение ***СЛУЖ2 {СЛУ_НОМ, СЛУ_ЗАРП, ПРО_НОМ}*** с множеством **FD** ***{СЛУ_НОМСЛУ_ЗАРП, СЛУ_НОМПРО_НОМ}*** не является атомарным, т.к. возможна декомпозиция на независимые проекции:

СЛУЖ3 {СЛУ_НОМ, СЛУ_ЗАРП} и ***СЛУЖ4 {СЛУ_НОМ, ПРО_НОМ}***.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Аномалии обновлений, связанные с наличием перекрывающихся возможных ключей

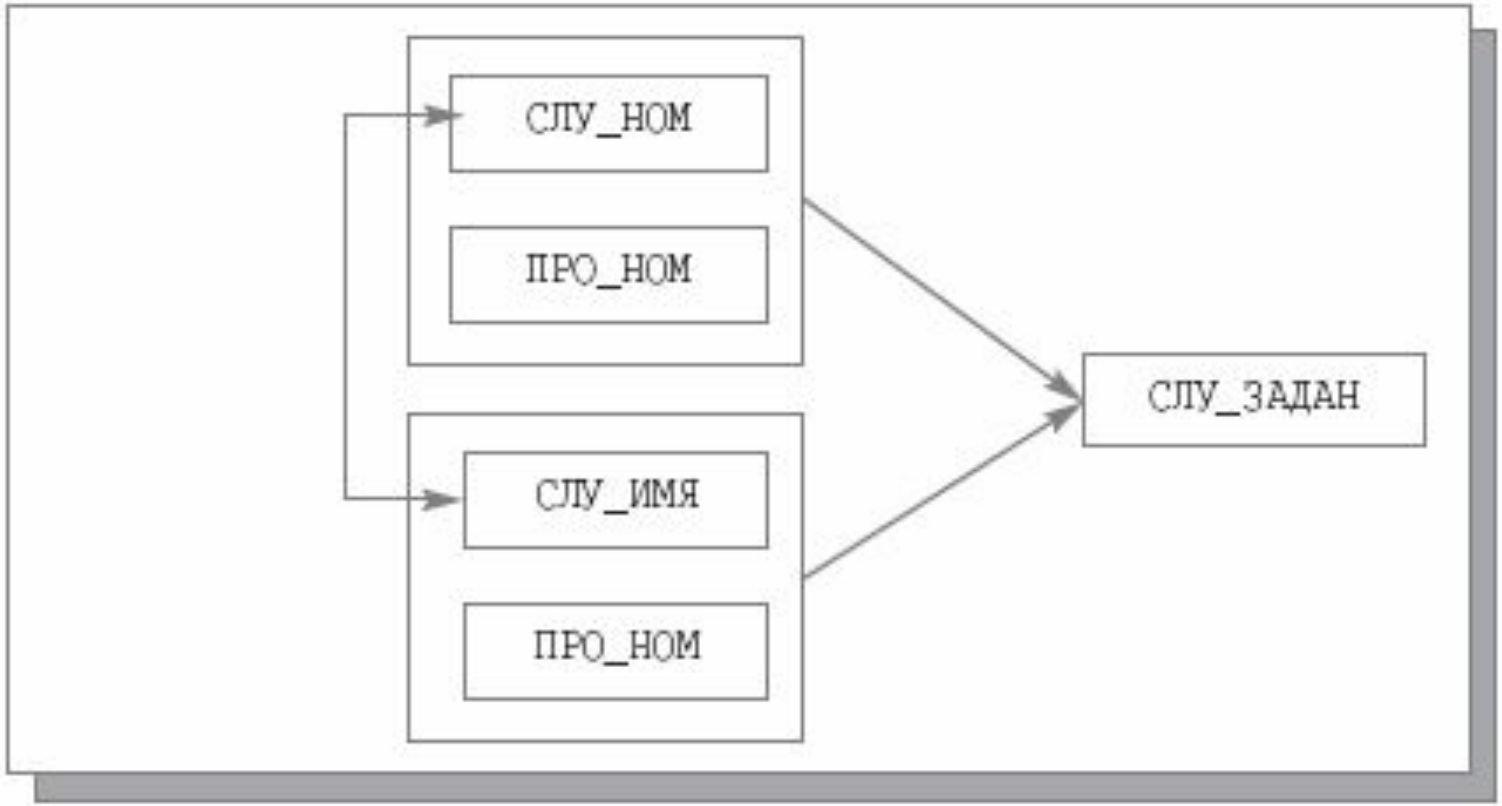


Рис. 16 Диаграмма FD отношения СЛУЖ_ПРО_ЗАДАН1

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Аномалии обновлений, связанные с наличием перекрывающихся возможных ключей

СЛУ_НОМ	СЛУ_ИМЯ	ПРО_НОМ	ПРО_ЗАДАН
2934	Иванов	1	A
2941	Иваненко	2	B
2934	Иванов	2	B
941	Иваненко	1	A

Рис. 17. Возможное значение переменной отношения СЛУЖ_ПРО_ЗАДАН1

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Третья нормальная форма

Определение: **Нормальная форма Бойса-Кодда**

Переменная отношения находится в ***нормальной форме Бойса-Кодда (BCNF)*** в том и только в том случае,

Когда любая выполняемая для этой переменной отношения нетривиальная и ***минимальная FD*** имеет в качестве детерминанта некоторый возможный ключ данного отношения.

Возможная декомпозиция

Диаграммы FD отношений СЛУЖ_НОМ_ИМЯ и СЛУЖ_НОМ_ПРО_ЗАДАН

Значение переменной отношения СЛУЖ_НОМ_ИМЯ

СЛУ_НОМ	СЛУ_ИМЯ
2934	Иванов
2941	Иваненко

Значение переменной отношения СЛУЖ_ПРО_ЗАДАН

СЛУ_НОМ	ПРО_НОМ	ПРО_ЗАДАН
2934	1	A
2941	2	B
29342		B
2941	1	A

Рис. 18. Диаграммы FD и значения переменных отношений СЛУЖ_НОМ_ИМЯ и СЛУЖ_НОМ_ПРО_ЗАДА

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Многозначные зависимости и четвертая нормальная форма

СЛУ_НОМ	ПРО_НОМ	СЛУ_ЗАДАН
2934	1	A
2934	1	B
2934	2	A
2934	2	B
....	....	....
2941	1	A
2941	1	D

Рис. 22. Возможное значение переменной отношения СЛУЖ_ПРО_ЗАДАН

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Аномалии обновлений:

- **Добавление кортежа.** Если уже участвующий в проектах сотрудник присоединяется к новому проекту, то к телу значения переменной отношения **СЛУЖ_ПРО_ЗАДАН** требуется добавить столько кортежей, сколько заданий выполняет этот сотрудник.

- **Удаление кортежей.** Если сотрудник прекращает участие в проектах, то отсутствует возможность сохранить данные о заданиях, которые он может выполнять.

- **Модификация кортежей.** При изменении одного из заданий сотрудника необходимо изменить значение атрибута ***СЛУ_ЗАДАН*** в стольких кортежах, в скольких проектах участвует сотрудник.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Многозначные зависимости и четвертая нормальная форма

Значение переменной отношения СЛУЖ_ПРО_НОМ

СЛУ_НОМ	ПРО_НОМ
2934	1
2934	2
....	....
2941	1

Значение переменной отношения СЛУЖ_ЗАДАНИЕ

СЛУ_НОМ	СЛУ_ЗАДАН
2934	A
2934	B
....	....
2941	A
2941	D

Рис. 23. Значения переменных отношений СЛУЖ_ПРО_НОМ и СЛУЖ_ЗАДАНИЕ

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Многозначные зависимости и четвертая нормальная форма

Определение: **Четвертая нормальная форма**

Переменная отношения *r* находится в ***четвертой нормальной форме (4NF)*** в том и только в том случае, когда она находится в ***BCNF***, и все ***MVD r*** являются ***FD*** с детерминантами – возможными ключами отношения ***r***.

Вариант:

Отношение ***r*** находится в четвертой нормальной форме (4NF) в том и только в том случае, если в случае существования многозначной зависимости ***A -->> B*** все остальные атрибуты ***r*** функционально зависят от ***A***.

Технологии проектирования реляционных БД

Проектирование реляционных баз данных на основе принципов нормализации

Заключение по разделу:

Процесс проектирования реляционной базы на основе метода нормализации преследует две основных цели:

- избежать избыточности хранения данных;
- устранить аномалии обновления отношений.

Классический подход к проектированию реляционных баз данных

Анализ критериев для нормализованных и ненормализованных моделей данных

Сравнение нормализованных и ненормализованных моделей

Критерий	Отношения слабо нормализованы (1НФ, 2НФ	Отношения сильно нормализованы (3НФ)
Адекватность базы данных предметной области	ХУЖЕ (-)	ЛУЧШЕ (+)
Легкость разработки и сопровождения базы данных	СЛОЖНЕЕ (-)	ЛЕГЧЕ (+)
Скорость выполнения вставки, обновления, удаления	МЕДЛЕННЕЕ (-)	БЫСТРЕЕ (+)
Скорость выполнения выборки данных	БЫСТРЕЕ (+)	МЕДЛЕННЕЕ (-)

Классический подход к проектированию реляционных баз данных

Анализ критериев для нормализованных и ненормализованных моделей данных

OLTP и OLAP-системы

OLTP-приложения (On-Line Transaction Processing (OLTP)-*оперативная обработка транзакций).*

Основополагающий признак: скорость и надежность выполнения коротких операций обновления данных.

Примеры: системы складского учета, системы заказов билетов, банковские системы, выполняющие операции по переводу денег, и т.п.

Классический подход к проектированию реляционных баз данных

Анализ критериев для нормализованных и ненормализованных моделей данных

OLTP и OLAP-системы

OLAP-приложения (On-Line Analitical Processing (OLAP) - *оперативная аналитическая обработка данных).*

OLAP-приложения оперируют с большими массивами данных, уже накопленными в OLTP-приложениях, взятыми их электронных таблиц или из других источников данных.

Разновидности OLAP-приложений:

- систем поддержки принятия решений (Decision Support System - DSS)
- хранилищ данных (Data Warehouse)
- систем интеллектуального анализа данных (Data Mining)

Классический подход к проектированию реляционных баз данных

Анализ критериев для нормализованных и ненормализованных моделей данных

OLTP и OLAP-системы

Признаки OLAP-приложений:

- Добавление в систему новых данных происходит относительно редко крупными блоками
- Данные, добавленные в систему, обычно никогда не удаляются.
- Перед загрузкой данные проходят различные процедуры "очистки", связанные с тем, что в одну систему могут поступать данные из многих источников
- Запросы к системе являются нерегламентированными и, как правило, достаточно сложными.
- Скорость выполнения запросов важна, но не критична

Концептуальные модели и схемы баз данных

Ограниченность реляционной модели:

- Модель не предоставляет достаточных средств для представления смысла данных. Семантика реальной предметной области должна независимым от модели способом представляться в голове проектировщика.

- Для многих приложений трудно моделировать предметную область на основе плоских таблиц. В ряде случаев на самой начальной стадии проектирования проектировщику приходится производить насилие над собой, чтобы описать предметную область в виде одной (возможно, даже ненормализованной) таблицы.

- Хотя весь процесс проектирования происходит на основе учета зависимостей, реляционная модель не предоставляет каких-либо средств для представления этих зависимостей. Несмотря на то, что процесс проектирования начинается с выделения некоторых существенных для приложения объектов предметной области ("сущностей") и выявления связей между этими сущностями, реляционная модель данных не предлагает какого-либо аппарата для разделения сущностей и связей.

Концептуальные модели и схемы баз данных

Семантическая модель данных – средство моделирование предметной области, обеспечение возможности выражения семантики данных.

Состав семантической модели

- структурная часть
- манипуляционная часть
- представление целостности

Одна из наиболее популярных семантических моделей данных – модель «Сущность-Связи» (кратко ER-модель).

Модель была предложена Ченом (Chen) в 1976 г.

Концептуальные модели и схемы баз данных

Основные понятия модели Entity-Relationship (Сущность-Связи)

- **Сущность** - это реальный или представляемый объект, информация о котором должна сохраняться и быть доступна.

- **Связь** - это графически изображаемая ассоциация, устанавливаемая между сущностями.

- **Атрибутом** сущности является любая деталь, которая служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности.

- **Уникальным идентификатором** сущности является атрибут, комбинация атрибутов, комбинация связей или комбинация связей и атрибутов, уникально отличающая любой экземпляр сущности от других экземпляров сущности того же типа.

Концептуальные модели и схемы баз данных

Рис. 27. Пример типа сущности

Определение: Сущность

Сущность – это реальный или представляемый объект, информация о котором должна сохраняться и быть доступной.

Определение: Связь

Связь – это графически изображаемая ассоциация, устанавливаемая между двумя типами сущностей.

Рис. 28. Пример типа связи

Концептуальные модели и схемы баз данных

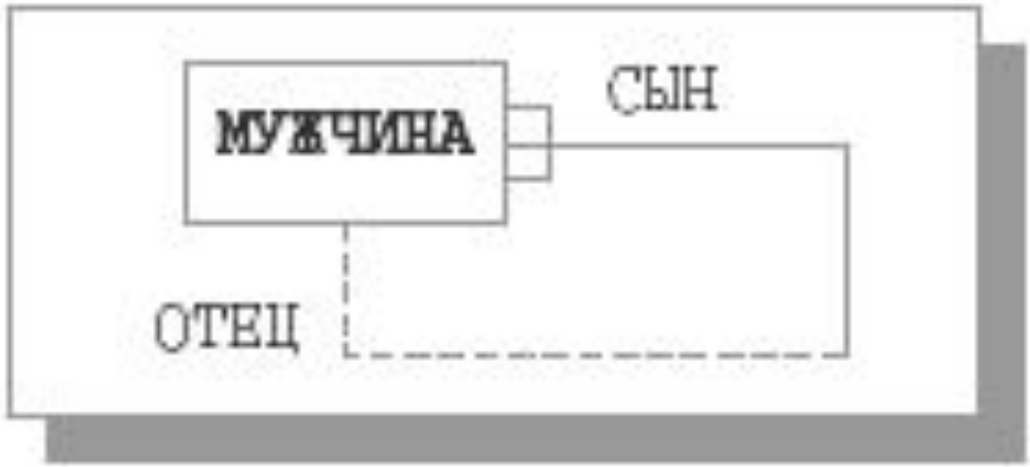


Рис. 29. Пример рекурсивного типа связи

- каждый *МУЖЧИНА* является сыном одного и только одного *МУЖЧИНЫ*;
- каждый *МУЖЧИНА* может являться отцом одного или более *МУЖЧИН*.

Определение: Атрибут

Атрибутом сущности является любая деталь, которая служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности.

Рис. 30. Пример типа сущности с атрибутами

Концептуальные модели и схемы баз данных

Уникальные идентификаторы типов сущности

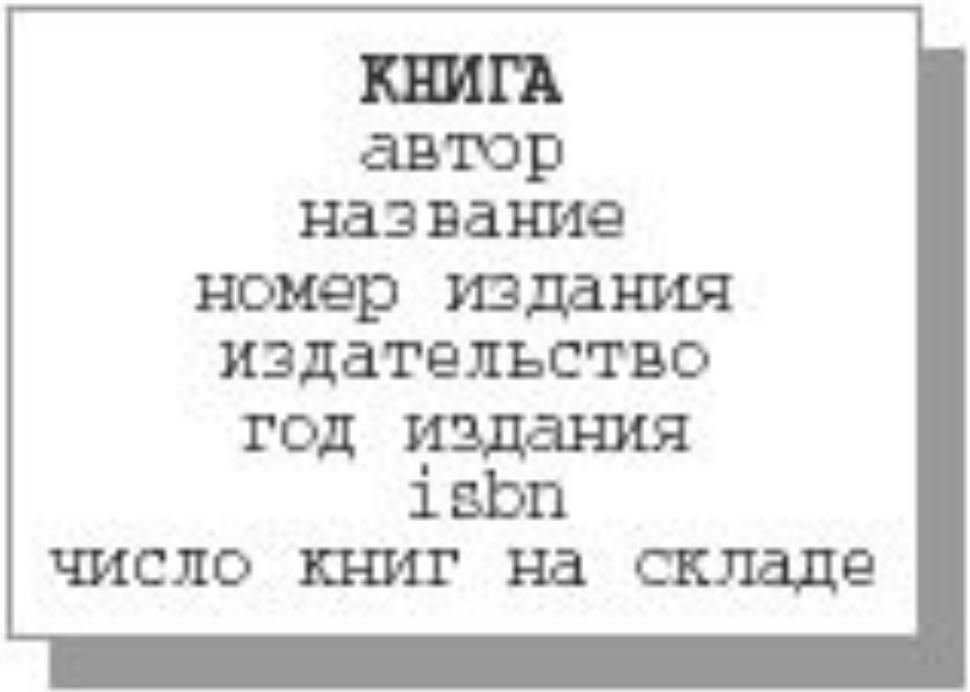


Рис. 31. Тип сущности, экземпляры которого идентифицируются атрибутами

Рис. 32. Тип сущности, экземпляры которого идентифицируются связью

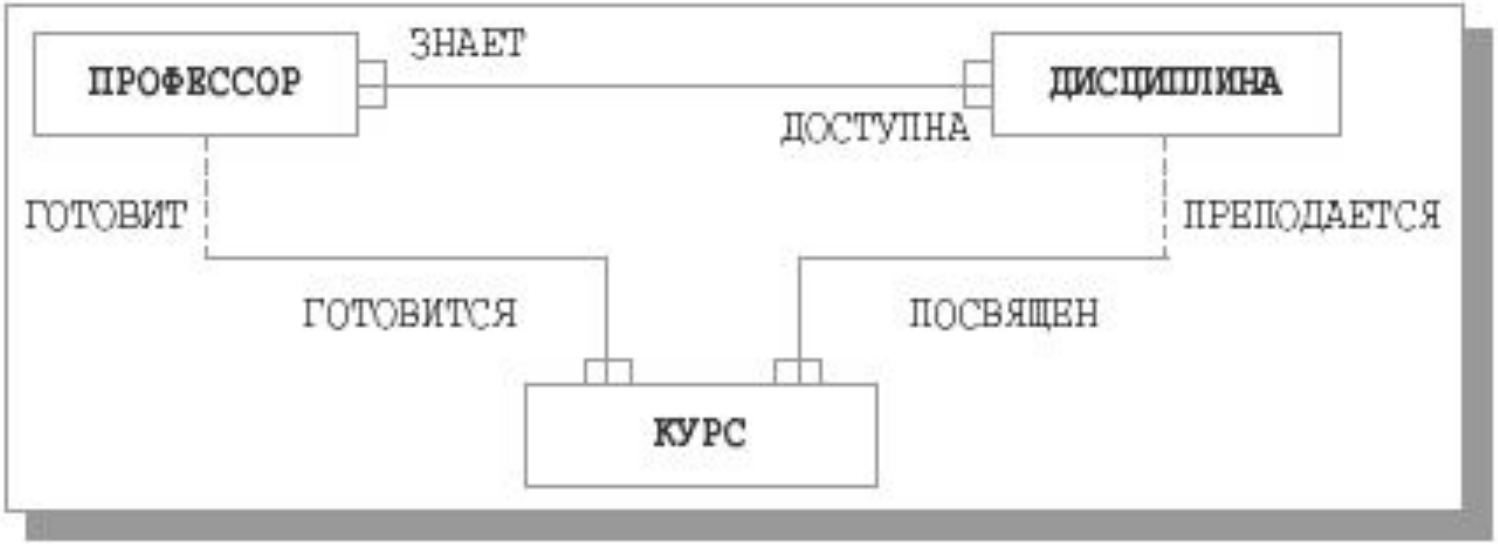


Рис. 33. Тип сущности, экземпляры которого идентифицируются комбинацией связей

Концептуальные модели и схемы баз данных

ER-диаграмма должна подчиняться следующим правилам:

- каждая сущность, каждый атрибут и каждая связь должны иметь имя (связь супертипа или ассоциативная связь может не иметь имени);
- имя сущности должно быть уникально в рамках модели данных;
- имя атрибута должно быть уникально в рамках сущности;
- имя связи должно быть уникально, если для нее генерируется таблица БД;
- каждый атрибут должен иметь определение типа данных;

Концептуальные модели и схемы баз данных

Нормальные формы ER-схем

- В первой нормальной форме ER-схемы устраняются повторяющиеся атрибуты или группы атрибутов, т.е. производится выявление неявных сущностей, "замаскированных" под атрибуты.

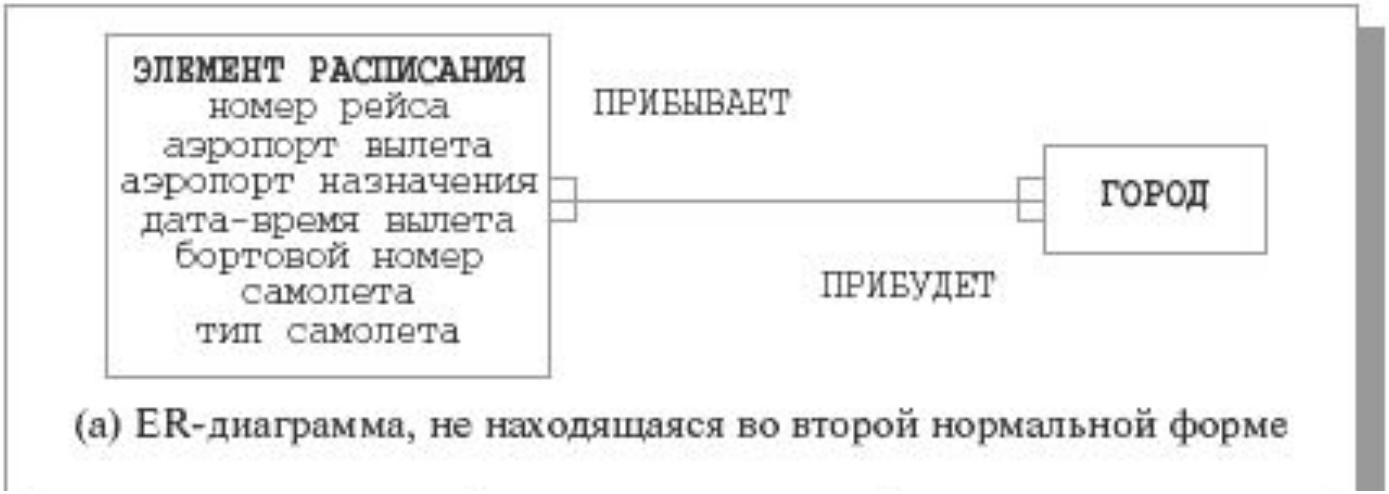
- Во второй нормальной форме устраняются атрибуты, зависящие только от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

- В третьей нормальной форме устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности.

Концептуальные модели и схемы баз данных

Рис. 35. Пример приведения ER-диаграммы к первой нормальной форме

Концептуальные модели и схемы баз данных



(а) ER-диаграмма, не находящаяся во второй нормальной форме

имеются следующие FD:

- *{номер рейса, дата-время вылета} -> бортовой номер самолета;*
- *номер рейса аэропорт -> вылета;*
- *номер рейса -> аэропорт назначения;*
- *бортовой номер самолета -> тип самолета.*

Рис. 36. Пример приведения ER-диаграммы ко второй нормальной форме

Концептуальные модели и схемы баз данных

между уникальным идентификатором и другими атрибутами типа сущности *ЭЛЕМЕНТ РАСПИСАНИЯ* имеются следующие функциональные зависимости:

- *{КОГДА, НА ЧЕМ, дата-время вылета} -> бортовой номер самолета*
- *{КОГДА, НА ЧЕМ, дата-время вылета} -> тип самолета*
- *бортовой номер самолета -> тип самолета*

(b) Аналог ER диаграммы после приведения ко второй нормальной форме

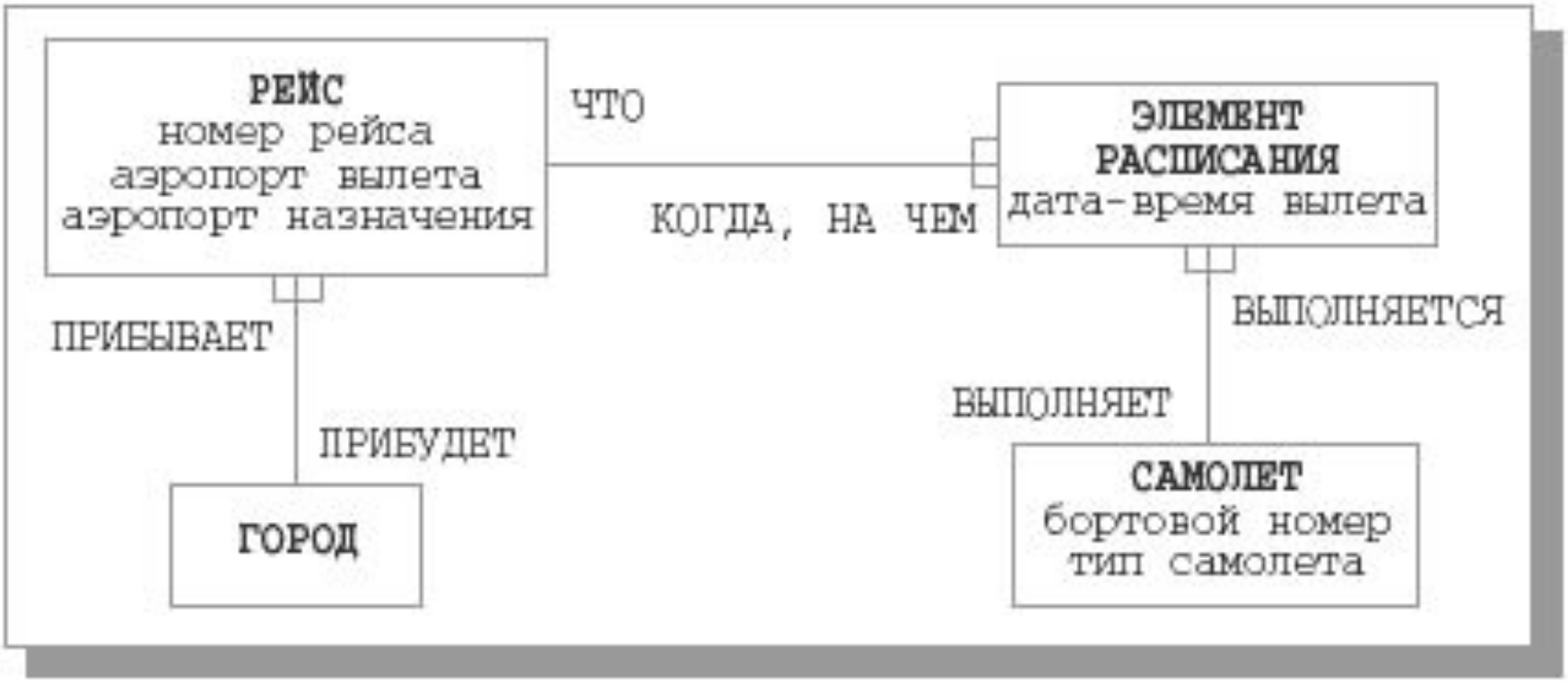


Рис. 37. Пример приведения ER-диаграммы к третьей нормальной форме

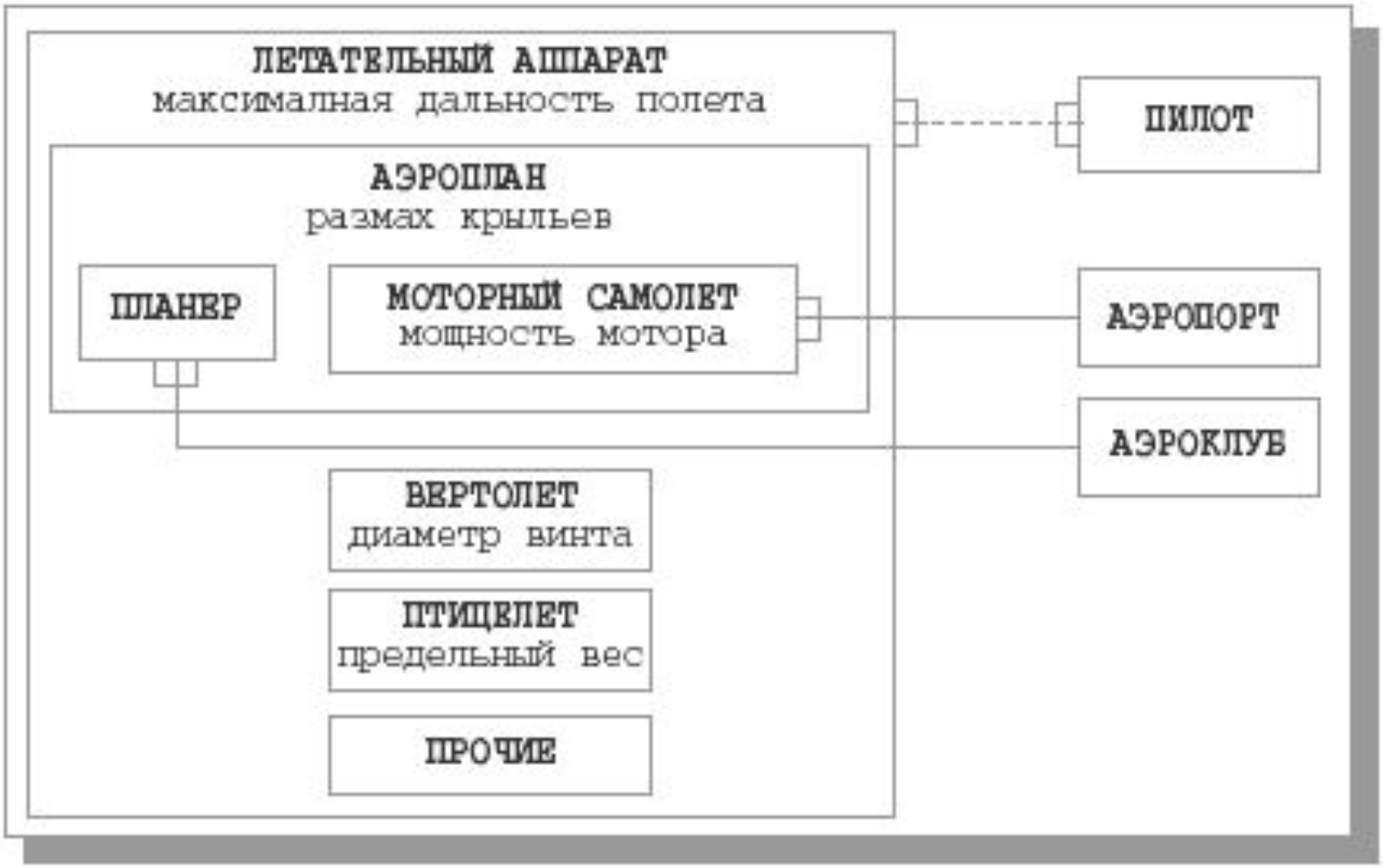


Рис. 38. Супертипы и подтипы сущности

Рис. 39. Пример ER-диаграммы со взаимно исключающими связями

Концептуальные модели и схемы баз данных

Получение реляционной схемы из ER-схемы

- **Шаг 1**. Каждая простая сущность превращается в таблицу. Простая сущность - сущность, не являющаяся подтипом и не имеющая подтипов. Имя сущности становится именем таблицы.

- **Шаг 2**. Каждый атрибут становится возможным столбцом с тем же именем; может выбираться более точный формат. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, - не могут.

- **Шаг 3**. Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. Если имеется несколько возможных уникальных идентификатора, выбирается наиболее используемый. Если в состав уникального идентификатора входят связи, к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно). Для именования этих столбцов используются имена концов связей и/или имена сущностей.

Концептуальные модели и схемы баз данных

Получение реляционной схемы из ER-схемы

- **Шаг 4.** Связи многие-к-одному (и один-к-одному) становятся внешними ключами. Т.е. делается копия уникального идентификатора с конца связи "один", и соответствующие столбцы составляют внешний ключ. Необязательные связи соответствуют столбцам, допускающим неопределенные значения; обязательные связи - столбцам, не допускающим неопределенные значения.

- **Шаг 5**. Индексы создаются для первичного ключа (уникальный индекс), внешних ключей и тех атрибутов, на которых предполагается в основном базировать запросы.

- **Шаг 6**. Если в концептуальной схеме присутствовали подтипы, то возможны два способа:
  - все подтипы в одной таблице (а)
  - для каждого подтипа - отдельная таблица (б)

Представление в реляционной схеме супертипов и подтипов сущности

Если в концептуальной схеме (ER-диаграмме) присутствуют подтипы, то возможны два способа их представления в реляционной схеме:

- (а) собрать все подтипы в одной таблице;
- (б) для каждого подтипа образовать отдельную таблицу.

Достоинства (а)) можно отнести следующее:

1. соответствие логике супертипов и подтипов;
2. обеспечение простого доступа к экземплярам супертипа и не слишком сложный доступ к экземплярам подтипов;
3. возможность обойтись небольшим числом таблиц.

Недостатки метода (а):

1. прикладная программа, имеющая дело с одной таблицей супертипа, должна включать дополнительную логику работы с разными наборами столбцов (в зависимости от значения столбца ТИП) и разными ограничениями целостности (в зависимости от особенностей связей, определенных для подтипа);

2. общая для всех подтипов таблица потенциально может стать узким местом при многопользовательском доступе по причине возможности блокировки таблицы целиком;

3. для индивидуальных столбцов подтипов должна допускаться возможность содержать неопределенные значения; таким образом, потенциально в общей таблице будет содержаться много неопределенных значений, что при использовании некоторых РСУБД может потребовать значительного объема внешней памяти.

Достоинства метода (b) состоят в следующем:

1. действуют более понятные правила работы с подтипами (каждому подтипу соответствует одноименная таблица);

2. упрощается логика приложений; каждая программа работает только с нужной таблицей.

Недостатки метода (b):

1. в общем случае требуется слишком много отдельных таблиц;

2. работа с экземплярами супертипа на основе представления, объединяющего таблицы супертипов, может оказаться недостаточно эффективной;

3. поскольку множество экземпляров супертипа является объединением множеств экземпляров подтипов, не все РСУБД могут обеспечить выполнение операций модификации экземпляров супертипа.

Представление в реляционной схеме взаимно исключающих связей

Существуют два способа формирования схемы реляционной БД при наличии взаимно исключающих связей (имеются в виду связи «один ко многим», причем конец связи «многие» находится на стороне сущности, для которой связи являются взаимно исключающими):

- (a) общее хранение внешних ключей;
- (b) раздельное хранение внешних ключей.

Рис. 40. Возможные модификации ER-диаграмм, позволяющие избежать взаимно исключающих связей

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

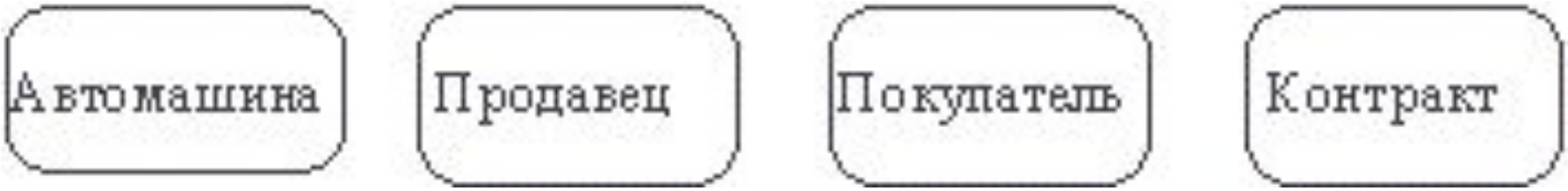


Рис. Графическое изображение сущности

- каждая сущность должна иметь уникальное имя, и к одному и тому же имени должна всегда применяться одна и та же интерпретация. Одна и та же интерпретация не может применяться к различным именам, если только они не являются псевдонимами;
- сущность обладает одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через связь;
- сущность обладает одним или несколькими атрибутами, которые однозначно идентифицируют каждый экземпляр сущности;
- каждая сущность может обладать любым количеством связей с другими сущностями модели.

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

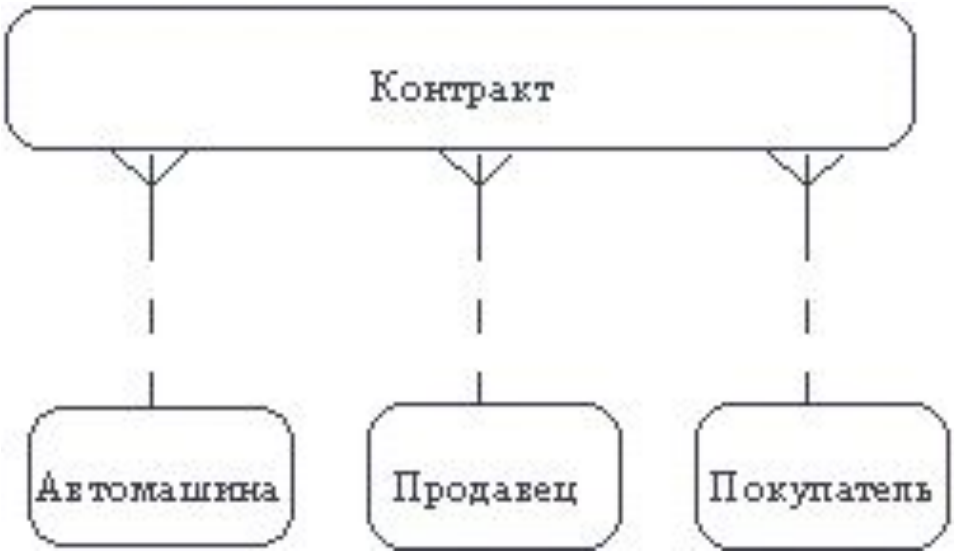


Рис. Графическое отображение связей

•

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

- Связь - это ассоциация между сущностями, при которой, как правило, каждый экземпляр одной сущности, называемой родительской сущностью, ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, называемой сущностью-потомком, а каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя. Таким образом, экземпляр сущности-потомка может существовать только при существовании сущности родителя.
- Связи может даваться имя, выражаемое грамматическим оборотом глагола и помещаемое возле линии связи.
- Имя каждой связи между двумя данными сущностями должно быть уникальным, но имена связей в модели не обязаны быть уникальными.
- Имя связи всегда формируется с точки зрения родителя, так что предложение может быть образовано соединением имени сущности-родителя, имени связи, выражения степени и имени сущности-потомка.

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

Рис. Виды связей по степени и обязательности

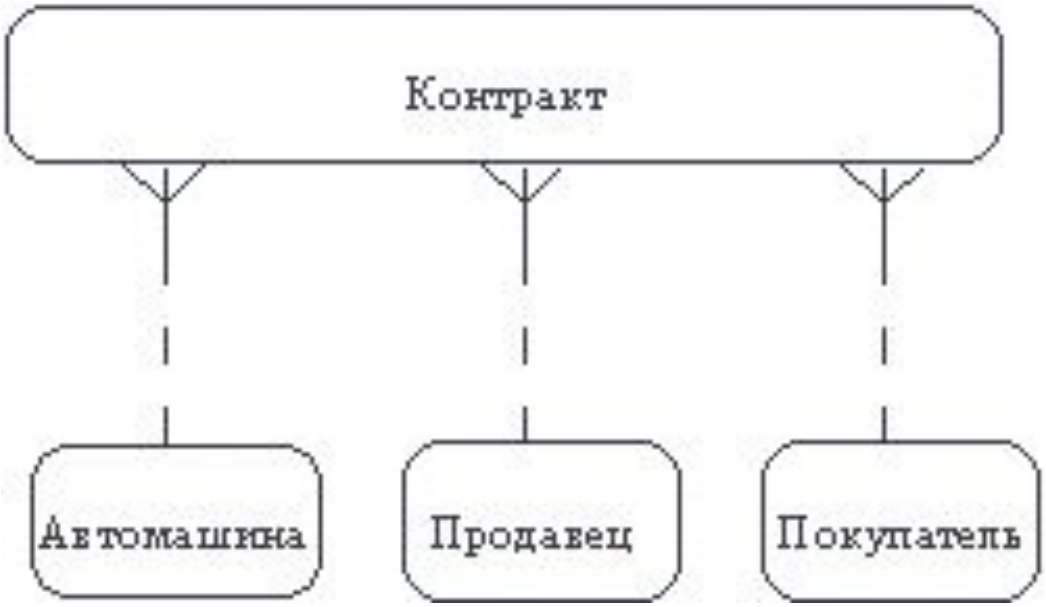


Рис. Пример отображения связей

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

Рис. Графическое отображение атрибутов на диаграмме

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

- Атрибут может быть либо обязательным, либо необязательным. Обязательность означает, что атрибут не может принимать неопределенных значений (null values).

- Атрибут может быть либо описательным (т.е. обычным дескриптором сущности), либо входить в состав уникального идентификатора (первичного ключа).

- Уникальный идентификатор - это атрибут или совокупность атрибутов и/или связей, предназначенная для уникальной идентификации каждого экземпляра данного типа сущности. В случае полной идентификации каждый экземпляр данного типа сущности полностью идентифицируется своими собственными ключевыми атрибутами, в противном случае в его идентификации участвуют также атрибуты другой сущности-родителя.

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

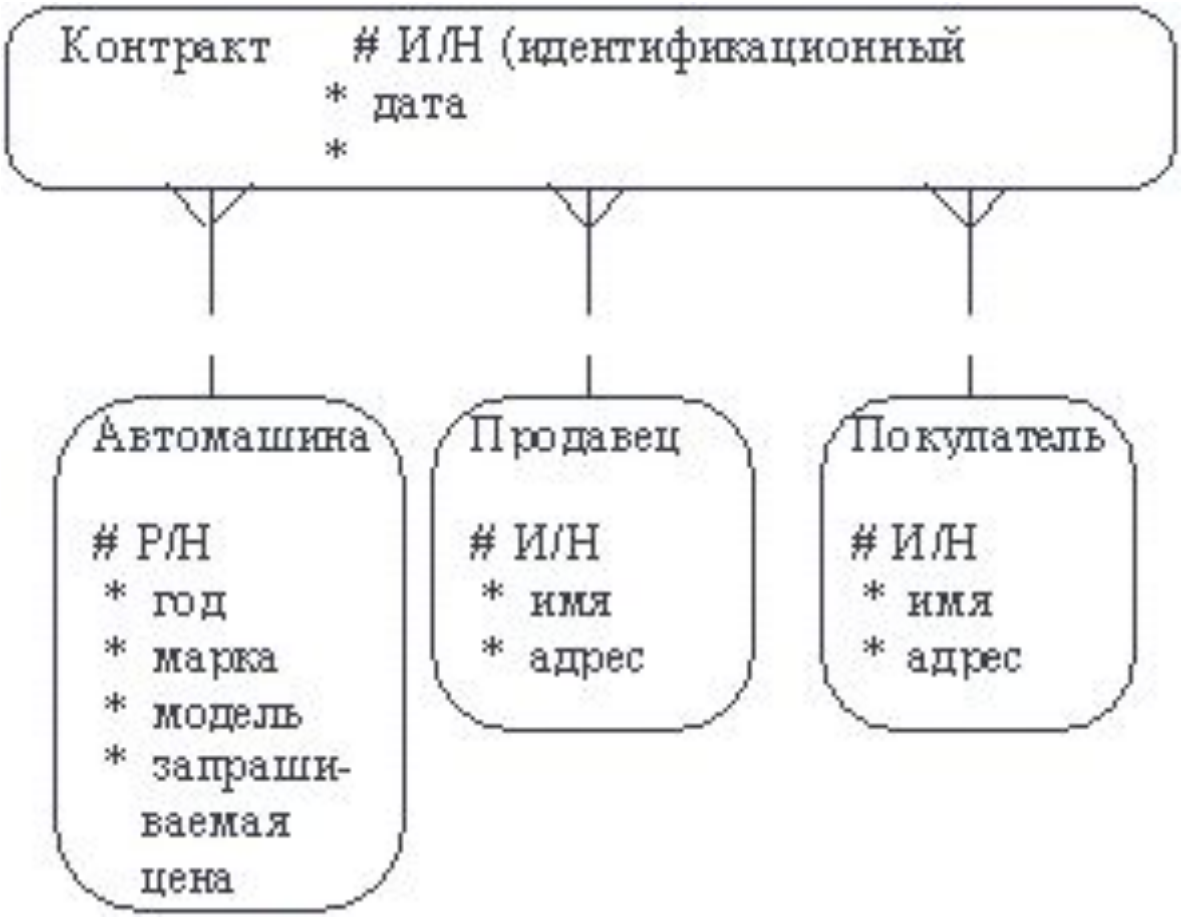


Рис. Пример отображения атрибутов на диаграмме

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

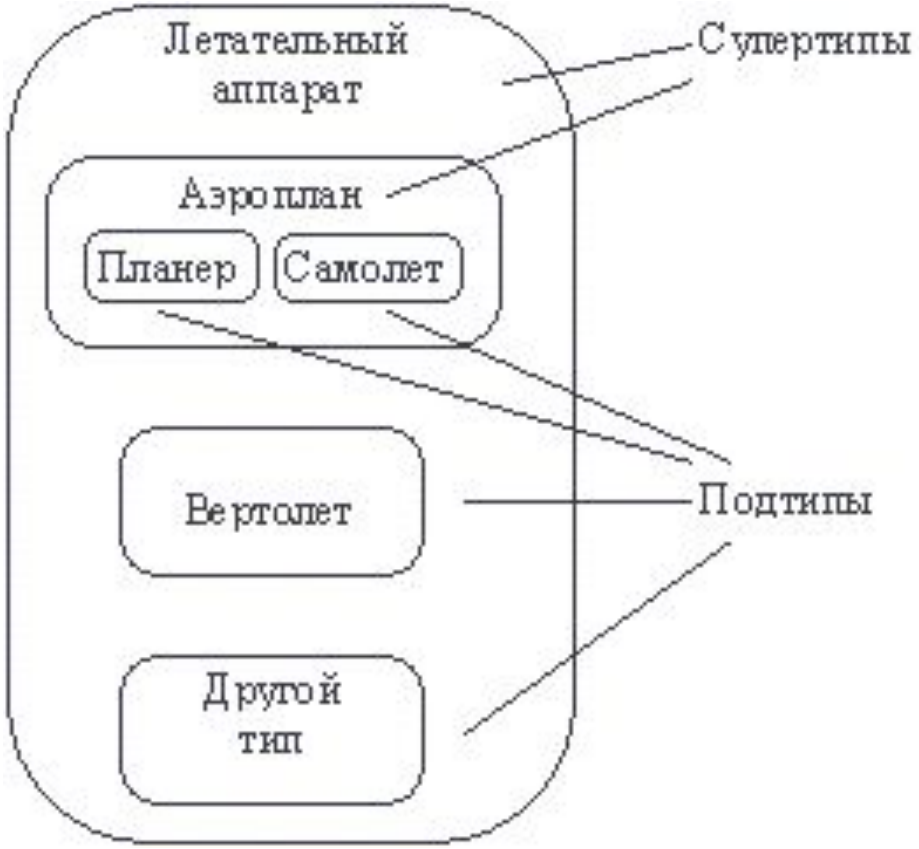


Рис. Подтипы и супертипы

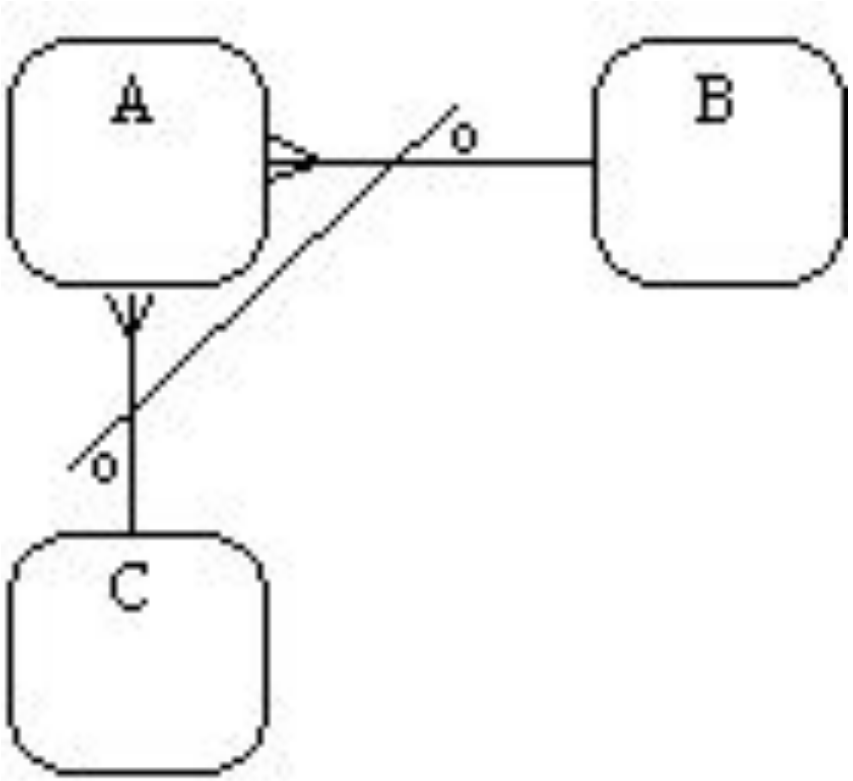


Рис. Взаимно исключающие связи

Концептуальные модели и схемы баз данных

Вариант ER-модели в нотации Баркера

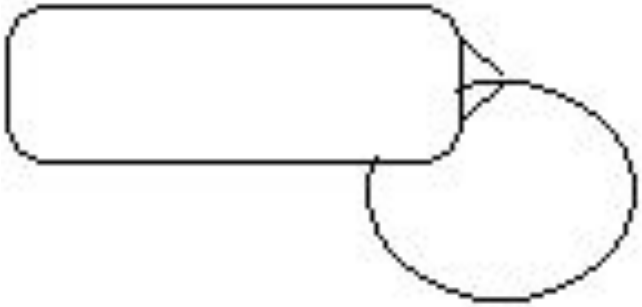


Рис. Рекурсивная связь (сущность может быть связана сама с собой)

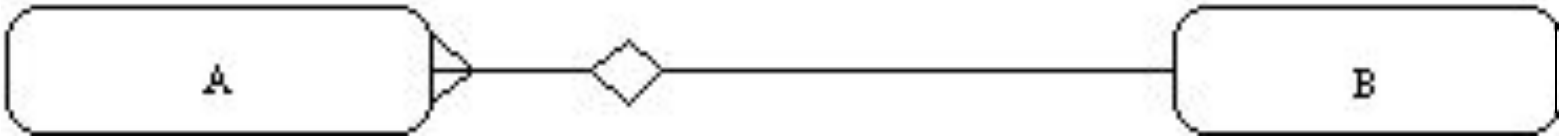


Рис. Неперемещаемая связь (экземпляр сущности не может быть перенесен из одного экземпляра связи в другой)

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

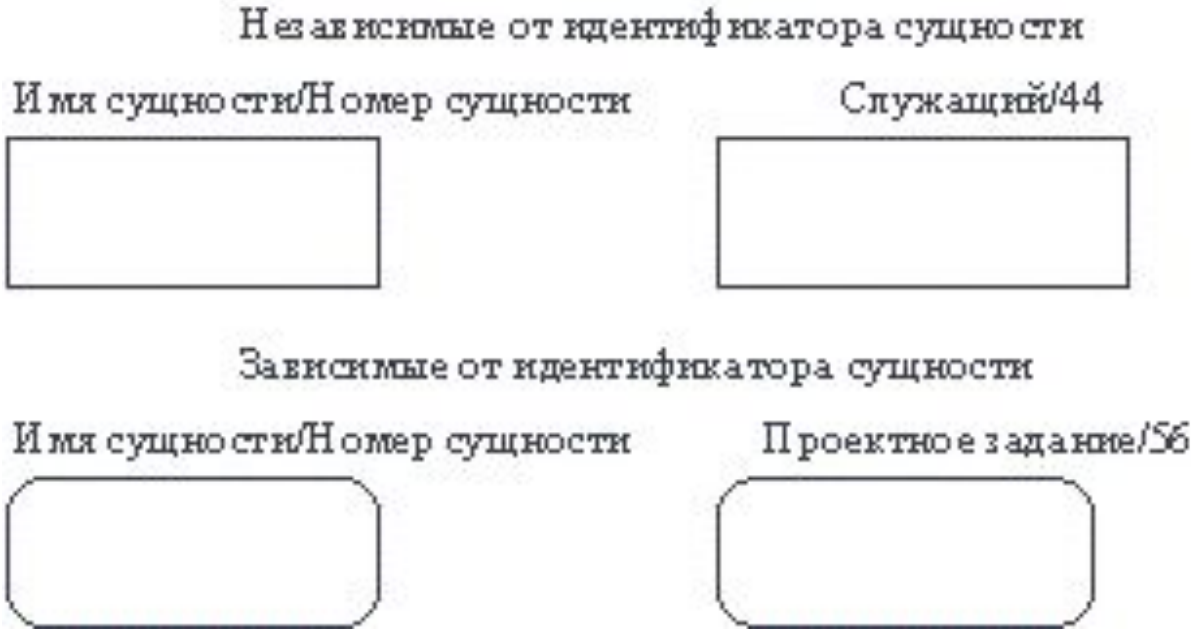


Рис. Графическое изображение сущности

- Сущность в методологии IDEF1X является независимой от идентификаторов или просто независимой, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями.
- Сущность называется зависимой от идентификаторов или просто зависимой, если однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

- Связь изображается линией, проводимой между сущностью-родителем и сущностью-потомком с точкой на конце линии у сущности-потомка.

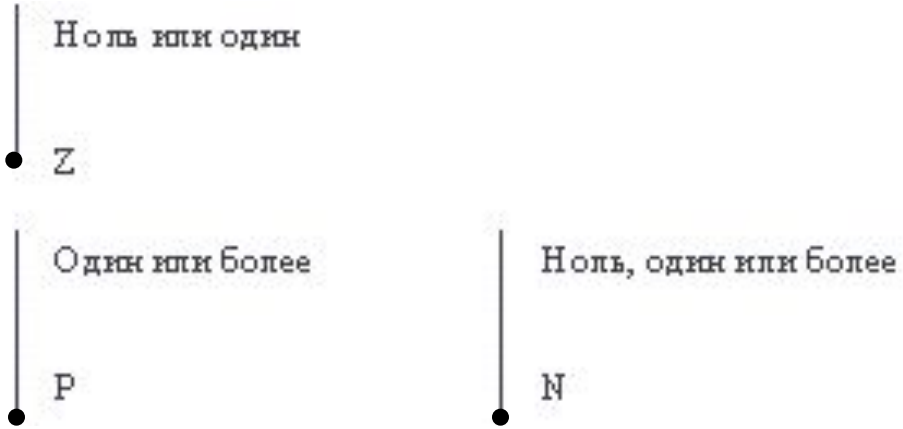


Рис. Мощность связи (количества экземпляров сущности-потомка, которое может существовать для каждого экземпляра сущности-родителя).

- **N** - каждый экземпляр сущности-родителя может иметь ноль, один или более связанных с ним экземпляров сущности-потомка (по умолчанию);
- **P**- каждый экземпляр сущности-родителя должен иметь не менее одного связанного с ним экземпляра сущности-потомка;
- **Z** - каждый экземпляр сущности-родителя должен иметь не более одного связанного с ним экземпляра сущности-потомка;
- каждый экземпляр сущности-родителя связан с некоторым фиксированным числом экземпляров сущности-потомка.

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

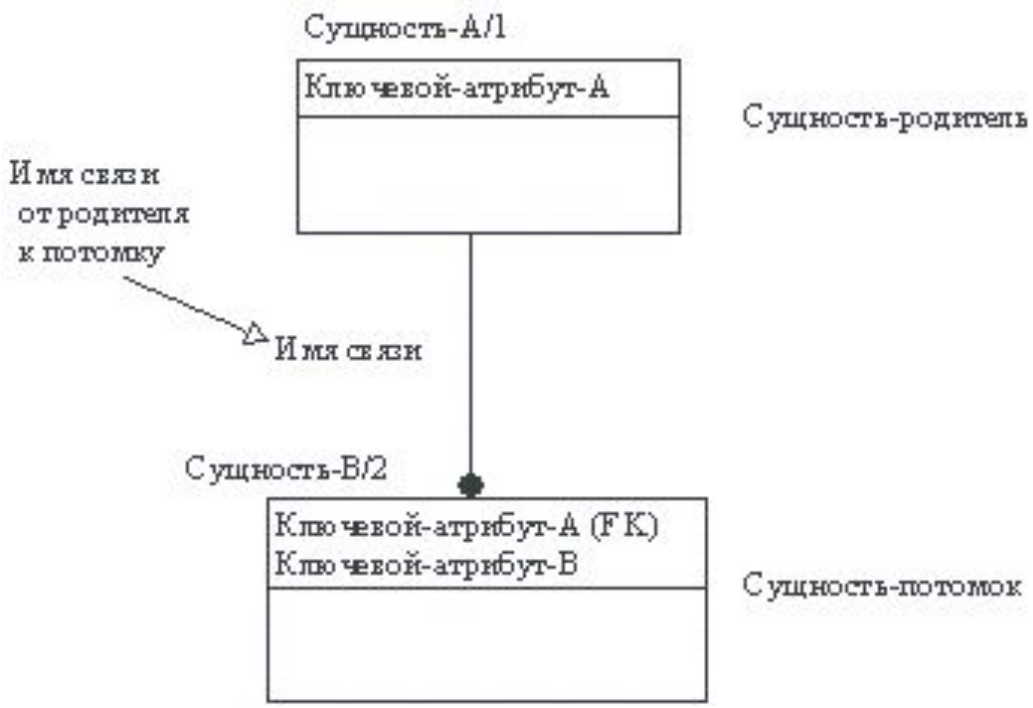


Рис. Идентифицирующая связь

- Если экземпляр сущности-потомка однозначно определяется своей связью с сущностью-родителем, то связь называется идентифицирующей, в противном случае – неидентифицирующей.

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

Рис. Неидентифицирующая связь

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

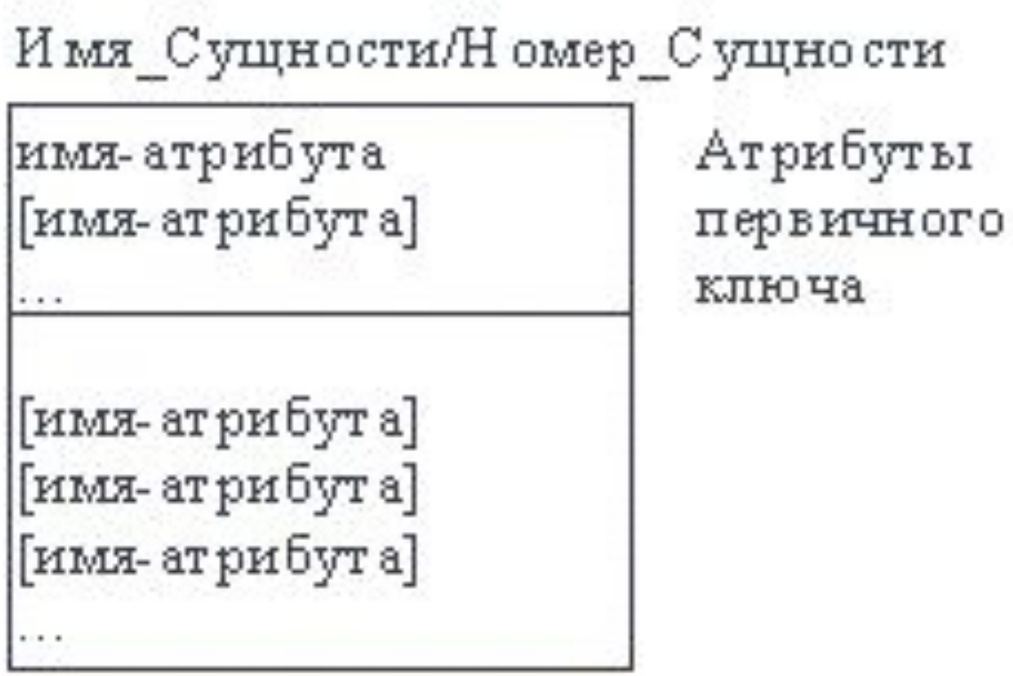


Рис. Атрибуты и первичные ключи

- Атрибуты изображаются в виде списка имен внутри блока сущности. Атрибуты, определяющие первичный ключ, размещаются наверху списка и отделяются от других атрибутов горизонтальной чертой

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

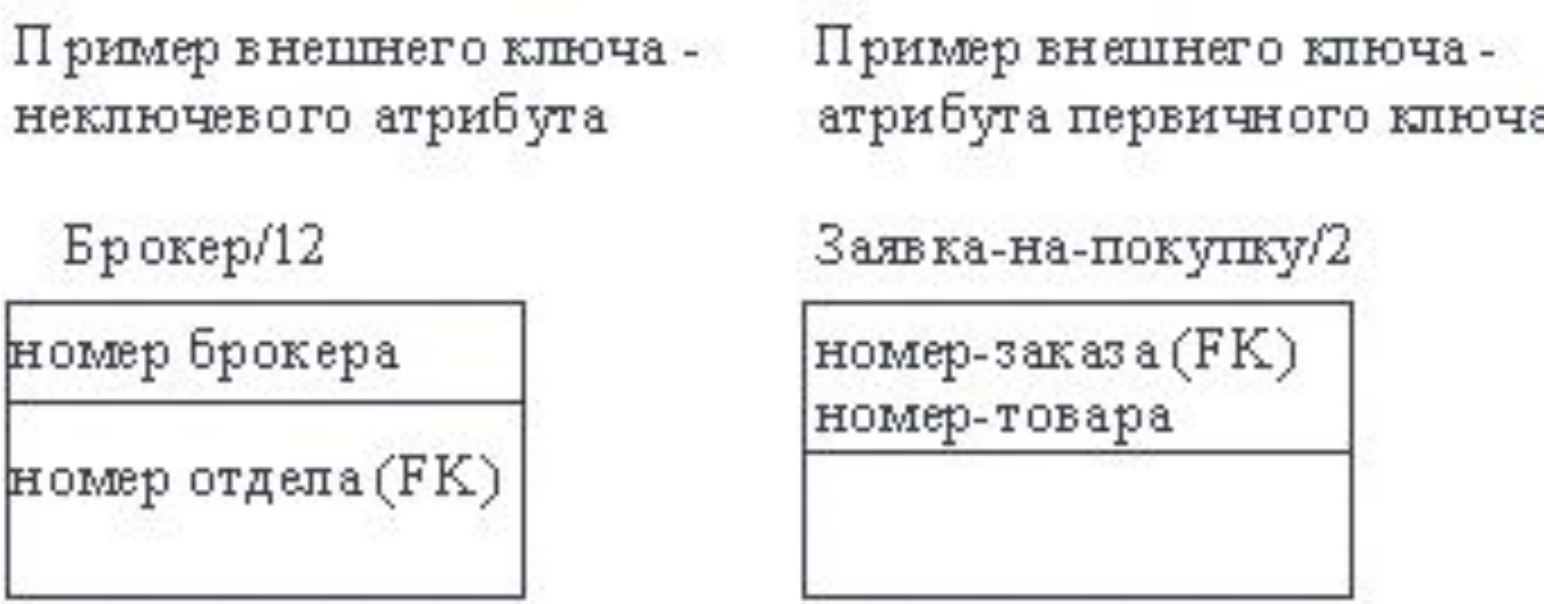


Рис. Примеры внешних ключей

- Сущности могут иметь также внешние ключи (Foreign Key), которые могут использоваться в качестве части или целого первичного ключа или неключевого атрибута. Внешний ключ изображается с помощью помещения внутрь блока сущности имен атрибутов, после которых следуют буквы FK в скобках.

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

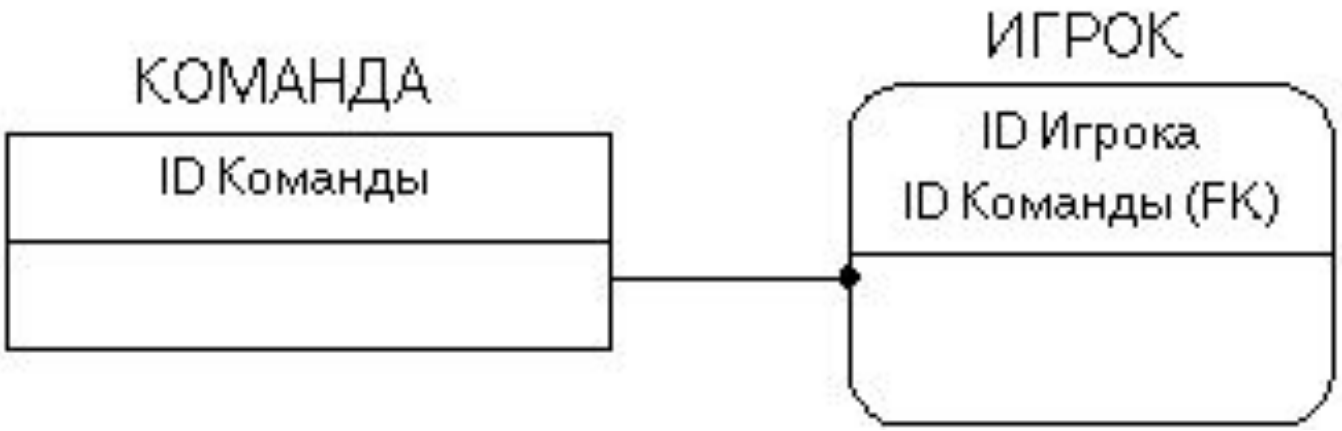
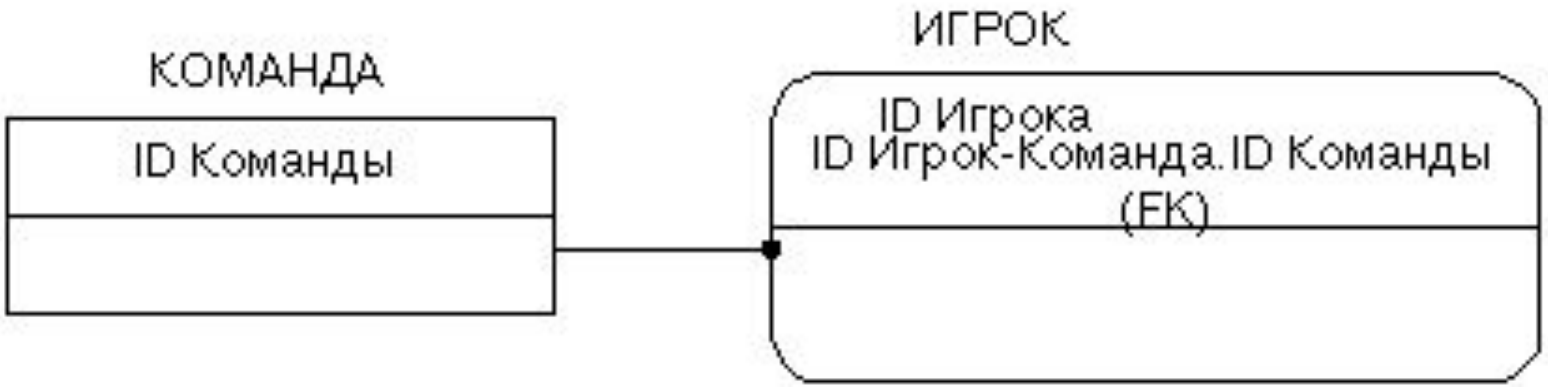


Рис. Объект с мигрировавшим внешним ключом (FK).

- **Внешние** ключи определяются как атрибуты первичных ключей родительского объекта, переданные дочернему объекту через их связь. Передаваемые атрибуты называются **мигрирующими**.

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X



Роль (Rolename)

- Когда внешние ключи мигрируют от родительской сущности через связь к дочерней сущности, они служат в модели двойную службу. Для понимания обеих ролей, иногда является полезным переименовать передаваемый ключ, для того, чтобы показать, какую он играет роль в дочерней сущности. Имя, назначаемое этому атрибуту, называется **ролью**.
- **Примечание:** Имена Ролей также используются для совместимости модели с наследуемыми моделями данных, где внешний и первичный ключи имели разные названия.

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

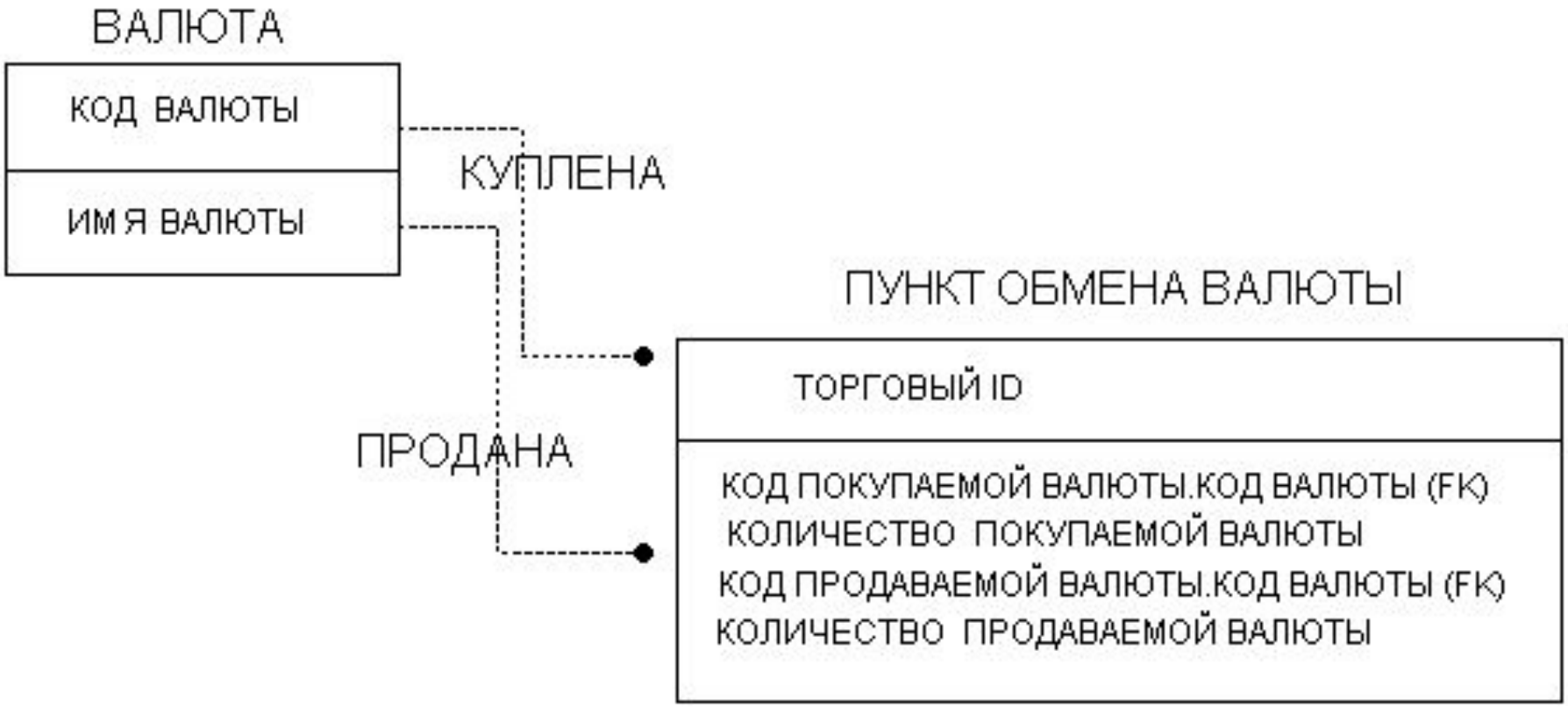


Рис. Модель Пункта обмена валюты

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

- Потенциальные ключи, не использующиеся в качестве первичных ключей, могут быть назначены в качестве **альтернативных ключей** и записаны под этим именем в модели. Символ (AKn), где n – это номер, ставится после атрибутов, составляющих альтернативный ключ.
- Альтернативные ключи часто используются для отображения различных индексов, используемых при доступе к данным.

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

- **Инверсный вход** – это атрибут или группа атрибутов, которые используются для доступа к сущности (так, как если бы они были первичными ключами), однако не обязательно находят только один экземпляр.

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X

Концептуальные модели и схемы баз данных

Вариант ER-модели в методологии IDEF1X