

Проектирование реляционной модели данных



Реляционная модель данных

Некоторые понятия:

- Данные хранятся в виде таблиц.
- **Таблица** состоит из заголовка, столбцов (**атрибутов**) и строк (**кортежей**).
- **Поле таблицы** - это значение, лежащее на пересечении строки и столбца.
- Множество значений, которые может принимать атрибут (или все встречающихся в столбце таблицы значения) - **домен** атрибута.
- Реляционные таблицы называются **отношениями**. (Relation – отношение).
- Число столбцов или атрибутов таблицы называется **степенью отношения**.
- Число строк или записей таблицы называется **мощностью отношения**.

Моделирование данных –

описание объектов некоторой предметной области, их атрибутов (свойств) и связей между ними.

Различаем модели данных:

- **логическая модель БД** (не зависит от СУБД)
 - для общего представления объектов, их атрибутов и связей между объектами
- **физическая модель БД** – описание на языке конкретной реляционной СУБД

Логическая модель данных

Логическая модель данных универсальна. Она отображает абстрактный взгляд на данные.

Объекты логической модели соответствуют объектам реального мира и могут носить такие же названия.

Например: Отдел, Сотрудник, Ведомость и т.п.

Логическая модель независима и не связана с конкретной СУБД. Поэтому реализация логической модели в среде конкретной СУБД для каждой СУБД будет своя.

Физическая модель данных.

Физическая модель данных - это представление логической модели данных на языке описания данных конкретной СУБД

Внутренние ограничения (требования) реляционной модели

1. В таблице каждая запись должна быть уникальна (не должно быть повторяющихся записей).
2. Конкретные данные должны храниться только в одной таблице (каждый факт хранится в одном месте).
3. Число отношений в модели должно быть оптимальным.

Дублирование (повторение) данных

Избыточное дублирование данных –

такое повторение данных, при котором данные можно удалить без потери информации

Избыточное дублирование данных вредно, так как порождает аномалии манипулирования данными

Избыточные данные исключаются методом декомпозиции отношений

Аномалии манипулирования данными

Манипулирование данными - действия над данными:

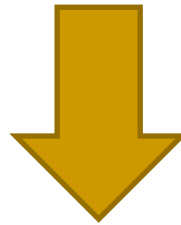
1. Добавление
2. Удаление
3. Изменение

Аномалии - ошибки

ФИО студента	ФИО куратора	Телефон куратора
Иванов Иван	Орлов НН	222
Петров Петр	Орлов НН	222
Сидоров Сидор	Соколов ТТ	333

Декомпозиция отношения с целью исключения избыточного дублирования данных

ФИО студента	ФИО куратора	Телефон куратора
Иванов Иван	Орлов НН	222
Петров Петр	Орлов НН	222
Сидоров Сидор	Соколов ТТ	333



ФИО студента	ФИО куратора
Иванов Иван	Орлов НН
Петров Петр	Орлов НН
Сидоров Сидор	Соколов ТТ

ФИО куратора	Телефон куратора
Орлов НН	222
Орлов НН	222
Соколов ТТ	333

ФИО студента	ФИО куратора
Иванов Иван	Орлов НН
Петров Петр	Орлов НН
Сидоров Сидор	Соколов ТТ

ФИО куратора	Телефон куратора
Орлов НН	222
Соколов ТТ	333

ФИО студента	Код куратора
Иванов Иван	1
Петров Петр	1
Сидоров Сидор	2

Код куратора	ФИО куратора	Телефон куратора
1	Орлов НН	222
2	Соколов ТТ	333

ФИО студента	ФИО куратора	Телефон куратора
Иванов Иван	Орлов НН	222
Петров Петр	Орлов НН	222
Сидоров Сидор	Соколов ТТ	333

Уникальность записи в таблице.

Первичный ключ отношения

Для идентификации записи в таблице служит атрибут (или совокупность атрибутов).

Называется **первичный ключ отношения**.

Для каждой записи таблицы значение первичного ключа **уникально!**

Первичный ключ может быть простым (состоит из одного атрибута) или составным (состоит из нескольких атрибутов)

Пример простого первичного ключа:

- табельный номер сотрудника
- номер паспорта

Пример составного первичного ключа:

- Номер класса, номер компьютера

Нормализация таблиц (отношений)

Нормальные формы

Нормализация.

Нормализация – декомпозиция отношений с целью исключения избыточного дублирования данных

Нормализация отношений (таблиц) основана на исследовании зависимостей неключевых атрибутов от атрибутов первичного ключа.

Первая нормальная форма (1НФ).

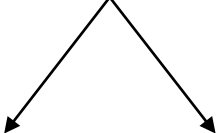
Отношение (таблица) находится в первой нормальной форме, если значение каждого атрибута в каждом поле таблицы имеет атомарное значение, то есть оно неделимо, не является списком, не содержит вложенности значений.

Вторая нормальная форма(2НФ).

Отношение находится во второй нормальной форме, если оно уже находится в первой нормальной форме, и все неключевые атрибуты функционально полно зависят от атрибутов первичного ключа, то есть полностью ими определяются. Зависят от всего ключа, если ключ составной.

Если в отношении есть атрибуты, которые зависят от части первичного ключа, то они выносятся в отдельную таблицу и им сопоставляется часть ключа, от которой они зависят.

первичный ключ
составной



Номер класса	Номер компьютера	ФИО администратора класса	Телефон администратора класса	Тип процессора	Наличие жесткого диска

Атрибуты ФИО администратора класса и Телефон администратора класса зависят только от части ключа - от атрибута Номер класса.

Для приведения таблицы ко 2-й нормальной форме эти атрибуты необходимо вынести в отдельную таблицу и сопоставить им часть ключа, от которой они зависят – атрибут Номер класса.

Таблица Компьютер

Номер класса	Номер компьютера	Тип процессора	Наличие жесткого диска
--------------	------------------	----------------	------------------------

Таблица Класс

Номер класса	ФИО администратора класса	Телефон администратора класса
--------------	---------------------------	-------------------------------

Третья нормальная форма (3НФ).

Отношение находится в третьей нормальной форме, если оно уже находится во второй, и в нем отсутствуют транзитивные зависимости между атрибутами.

Транзитивная зависимость – это зависимость одного атрибута от другого через третий.

Если А зависит от В ($B \rightarrow A$), а С зависит от А ($A \rightarrow C$),

то С зависит от В транзитивно ($B \rightarrow A \rightarrow C$).

Для исключения транзитивной зависимости атрибуты, которые зависят от первичного ключа транзитивно, выносятся в отдельную таблицу, где им сопоставляется атрибут, через который они зависят от первичного ключа.

Например.

Номер класса	Номер компь- ютера	Тип процессора	Фирма-произ- водитель процессора	Телефон фирмы- произ- водителя Процессора	Наличие жесткого диска
-----------------	--------------------------	-------------------	--	--	------------------------------

В данной таблице атрибуты Фирма-производитель процессора и Телефон фирмы-производителя процессора зависят от первичного ключа транзитивно через атрибут Тип процессора. Для приведения таблицы к 3 НФ таблица декомпозируется следующим образом:

Таблица Компьютер

Номер класса	Номер компь- ютера	Тип процессора	Наличие жесткого диска
-----------------	--------------------------	-------------------	------------------------------

Таблица Процессор

Тип процессора	Фирма-произ- водитель процессора	Телефон фирмы-произ- водителя Процессора
-------------------	--	---

Приведение таблиц к третьей нормальной форме считается достаточным для того, чтобы завершить декомпозицию.

Порядок проектирования реляционной модели данных

Последовательность действий при проектирования реляционной модели данных :

- Обозначить цель создания базы данных (БД)
- Выделить информационные объекты моделируемой системы с позиции цели БД.
- Описать каждый информационный объект набором характеристик (атрибутов), которые представляют важность с точки зрения цели БД и заложенных в БД функций.
- Для каждого информационного объекта определить первичный ключ - атрибут или совокупность атрибутов.
- Данные каждого информационного объекта внести в отдельную таблицу так, чтобы значения в каждом поле таблицы были атомарны, и, таким образом, **привести каждую таблицу к 1 нормальной форме.**

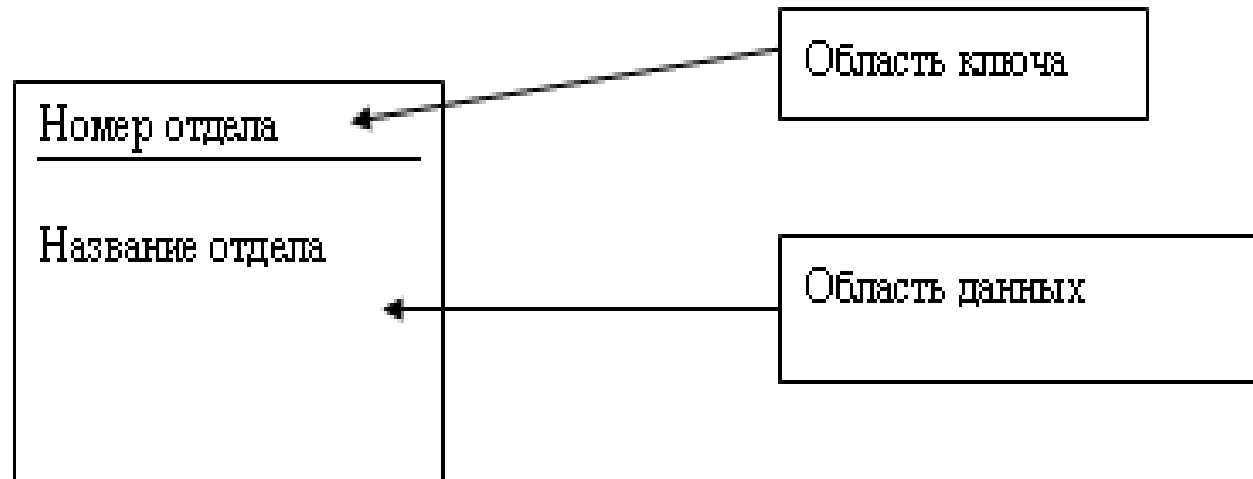
- **Привести отношения ко второй нормальной форме** - декомпозировать каждую таблицу так, чтобы в ней остались только атрибуты, которые зависят от всего первичного ключа. Элементы данных (атрибуты), зависящие от компонентов первичного ключа вынести в новые таблицы.
- **Привести отношения к третьей нормальной форме** - в новые отношения вынести элементы данных (атрибуты), которые зависят от атрибутов первичного ключа транзитивно, через другие атрибуты.

Диаграммы модели данных

На диаграмме сущность изображается в виде прямоугольника, разделенного горизонтальной линией на две части.

Верхняя часть служит для изображения ключевых атрибутов и называется областью ключа.

Нижняя часть отводится для неключевых атрибутов и называется областью данных.



Зависимые и независимые сущности

Сущность называется независимой (родительской) если для идентификации ее экземпляров (записей) не требуются атрибуты других сущностей.

Независимые сущности отображаются в модели в виде прямоугольника с прямыми углами.

Зависимые (дочерние) сущности изображаются в модели в виде прямоугольника со скругленными углами.

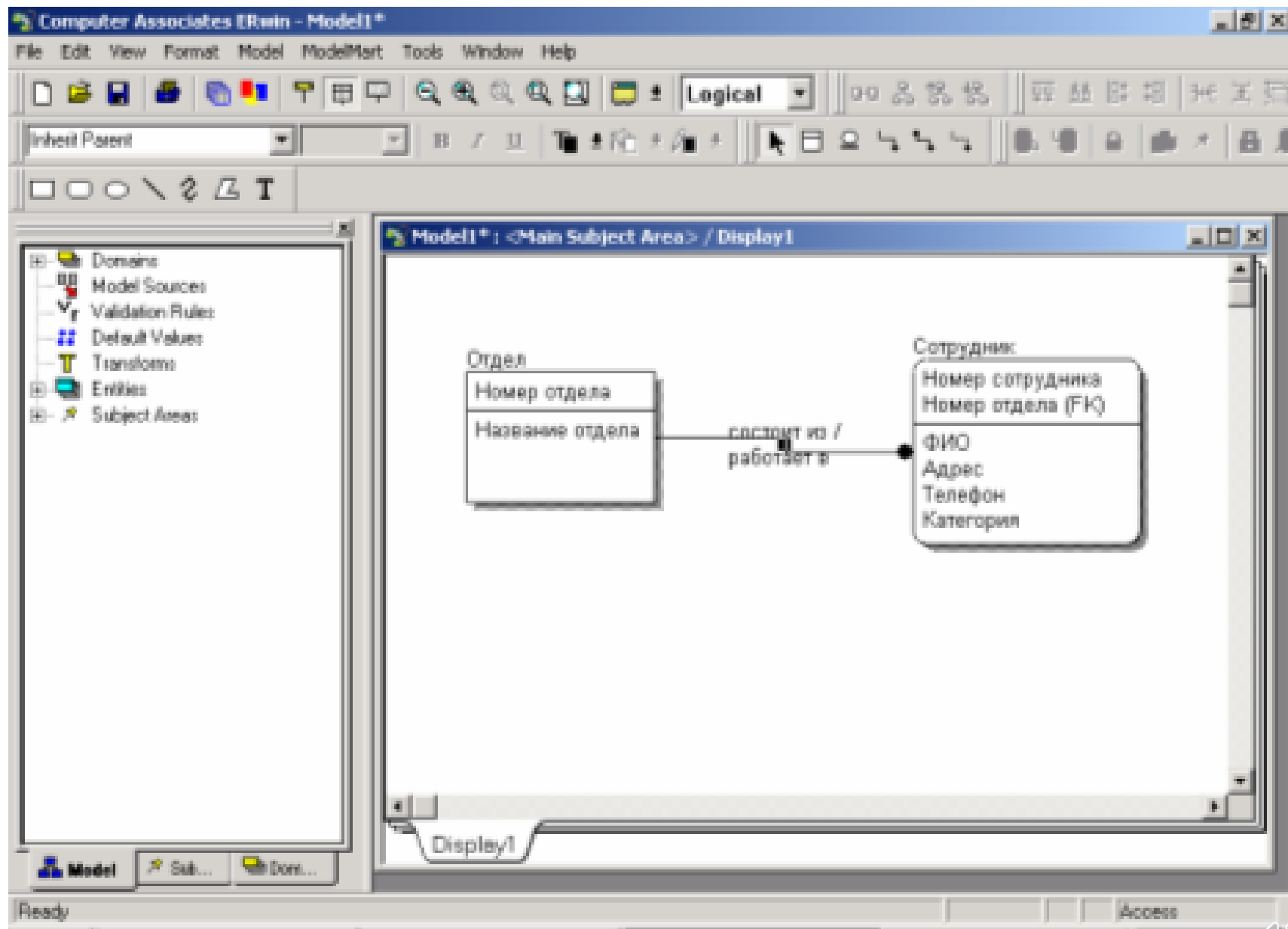
СВЯЗИ.

Связи показывают, как сущности соотносятся друг с другом логически. Связь именуется глаголом или глагольной фразой (Relationship Phrases). Например:

СТУДЕНТ <выполняет> ЗАДАНИЯ

Идентифицирующая связь.

- Установление идентифицирующей связи сопровождается миграцией ключевого атрибута родительской сущности в область ключа дочерней сущности.
- Мигрирующий ключ в дочерней сущности помечается справа FK как внешний ключ (foreign key).
- Идентифицирующая связь на диаграмме изображается в виде сплошной линии с точкой на конце связи на стороне дочерней сущности.

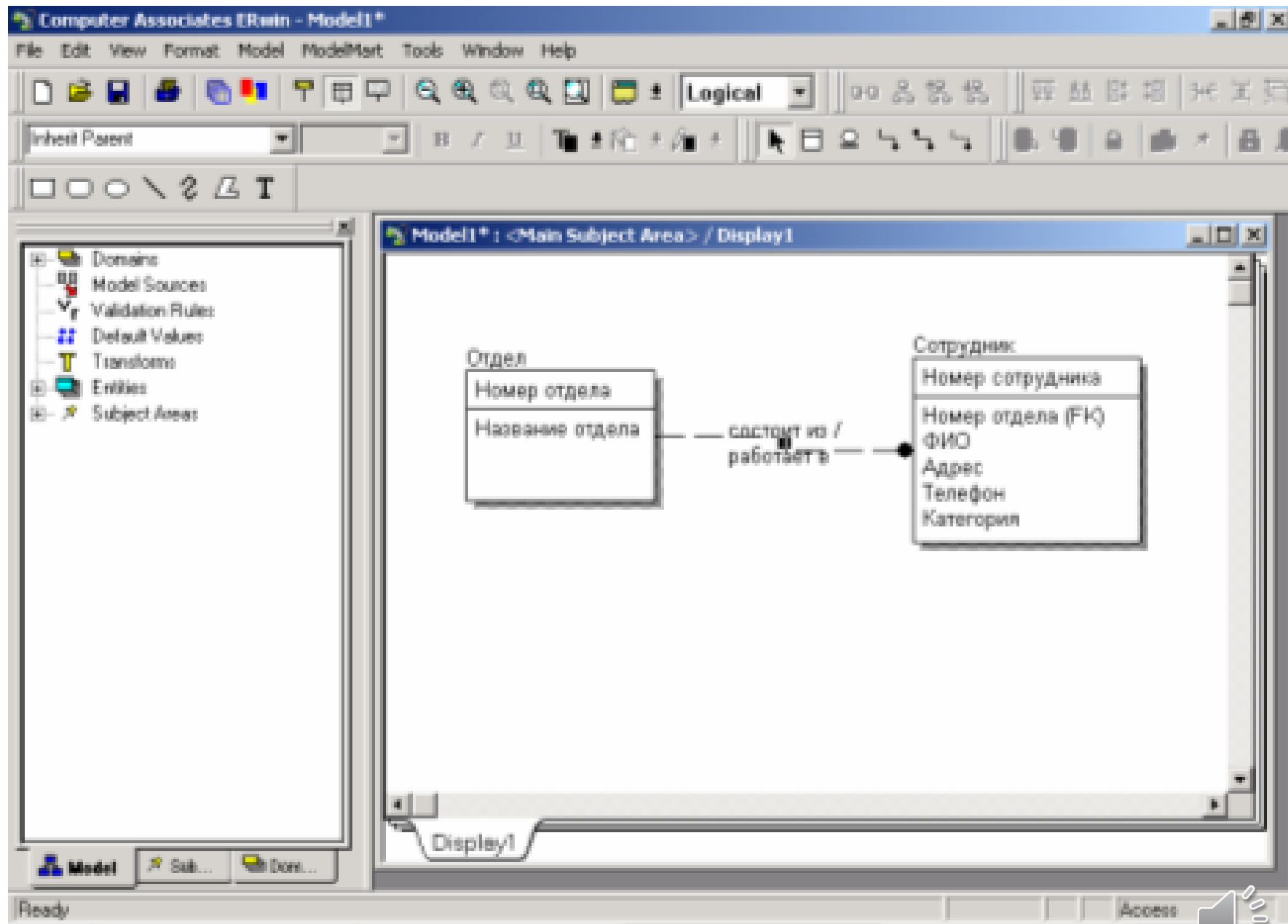


Неидентифицирующая связь.

При установлении неидентифицирующей связи атрибуты родительской сущности мигрируют в область данных, где также помечаются справа как внешний ключ(FK).

Неидентифицирующая связь изображается в виде пунктирной линии с точкой на конце связи на стороне дочерней сущности.

В случае, когда при характеристике связи разрешаются значения Null, на стороне родительской сущности возможно изображение неокрашенного ромба.



Мощность связи(Cardinality).

Отношение числа экземпляров родительской сущности к числу экземпляров дочерней отображается при помощи Мощности связи – опция **Cardinality** на вкладке General окна Relationship Editor. Различают 4 типа мощности связи:

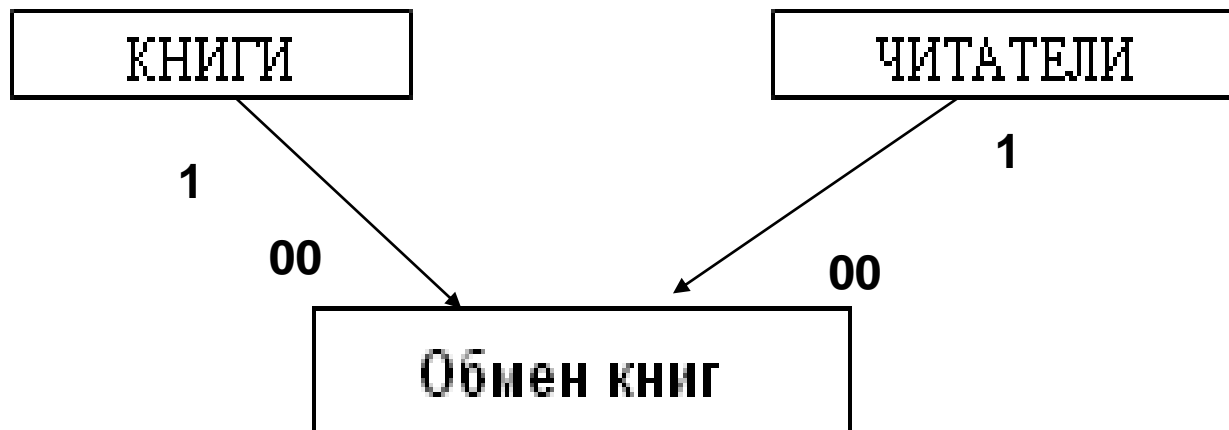
Родитель	Потомок	Описание
1	0,1 или много	одному экземпляру родительской сущности соответствует 0,1 или много экземпляров дочерней
1	1 или много	одному экземпляру родительской сущности соответствует 1 или много экземпляров дочерней. Связь помечается символом R
1	0 или 1	одному экземпляру родительской сущности соответствует 0 или 1 экземпляров дочерней. Множественные значения исключены. Связь помечается символом Z
1	Конкретное число	одному экземпляру родительской сущности соответствует точно установленное количество экземпляров дочерней.

Связь типа многие-ко-многим

возможна только в логической модели.

При переходе к физической модели эта связь автоматически разрешается в связи типа один-ко-многим. Преобразование выполняется путем добавления новой таблицы, отображающей взаимодействие между исходными сущностями. Например:





Ссылочная целостность (RI- Referential Integrity).

Целостность базы данных по ссылкам, - соответствие значений первичных и внешних ключей отношений (сущностей).

Например:

Две сущности Отдел и Сотрудник связаны идентифицирующей связью.

В таблице Сотрудники не может быть записи с номером отдела, которого нет в таблице Отдел.

Иначе такая запись как бы «повиснет в воздухе».

Сотрудник окажется без номера отдела.

Сотрудник не может существовать без Отдела, т.к. связь является идентифицирующей (значение Null для атрибута Номер Отдела запрещено).

Пример.

БД Библиотека

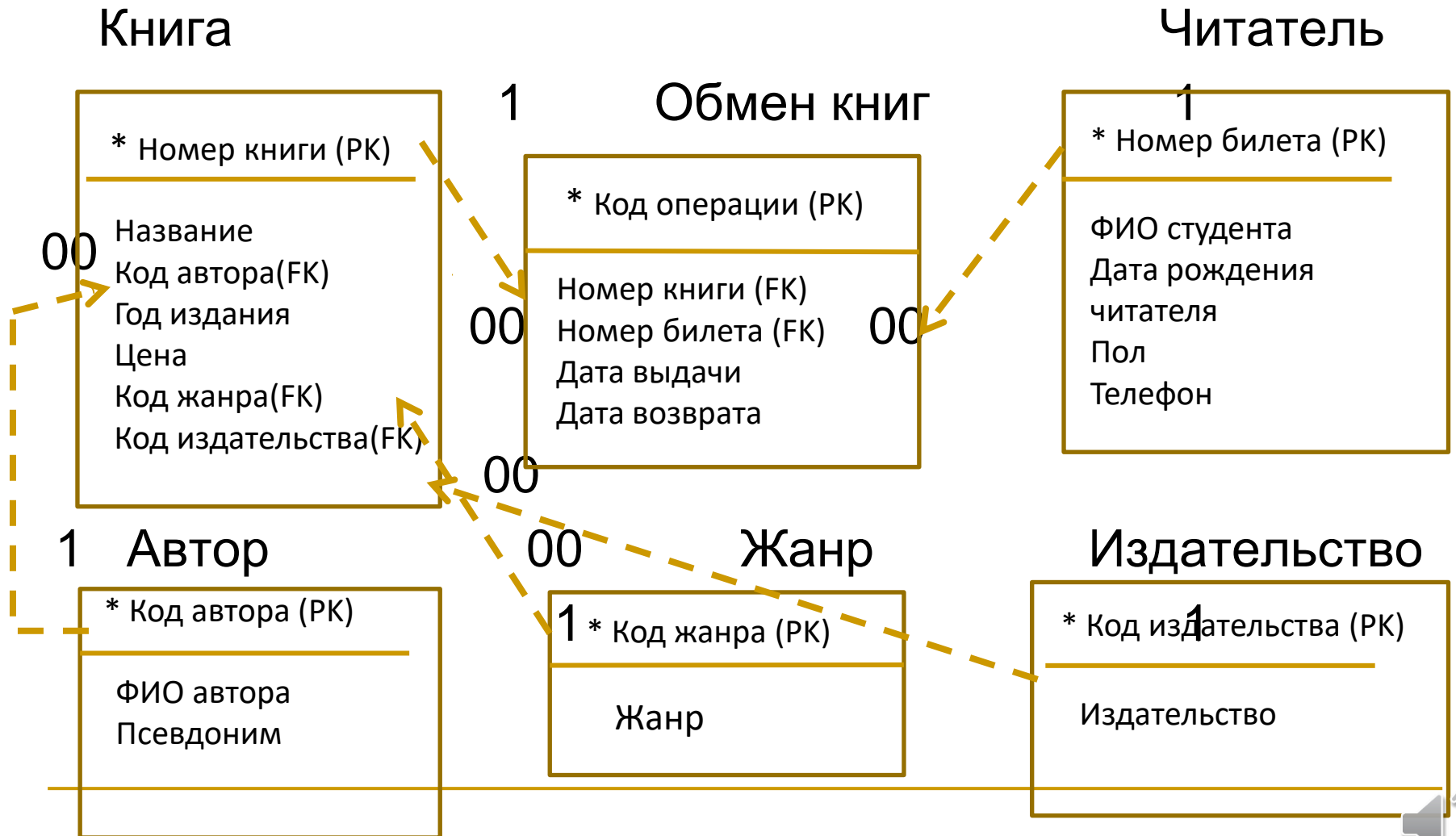
Задачи

1. Вести учет книг
2. Вести учет читателей
3. Вести учет выдачи/возврата книг

Информационные объекты

1. Книга
2. Жанр
3. Читатель
4. Обмен книг
5. Автор
6. Издательство

Атрибутивная логическая модель БД



Примеры нормализации отношений (таблиц)

Пример 1.

Код сотрудника	ФИО	Должность	Проекты
1	Иванов Иван Иванович	программист	ID: 123; Название: Система управления паровым котлом; Дата сдачи: 30.09.2011 ID: 231; Название: ПС для контроля и оповещения о превышениях ПДК различных газов в помещении; Дата сдачи: 30.11.2011 ID: 321; Название: Модуль распознавания лиц для защитной системы; Дата сдачи: 01.12.2011
2	Петров Петр Петрович	программист	ID: 231; Название: ПС для контроля и оповещения о превышениях ПДК различных газов в помещении; Дата сдачи: 30.11.2011
3	Сидоров Сидор Сидорович	оператор	ID: 123; Название: Система управления паровым котлом; Дата сдачи: 30.09.2011 ID: 231; Название: ПС для контроля и оповещения о превышениях ПДК различных газов в помещении; Дата сдачи: 30.11.2011 ID: 321; Название: Модуль распознавания лиц для защитной системы; Дата сдачи: 01.12.2011

Код сотрудника	ФИО	Должность	Проекты
1	Иванов Иван Иванович	Программист	ID: 123; Название: Система управления паровым котлом; Дата сдачи: 30.09.2011
1	Иванов Иван Иванович	Программист	ID: 231; Название: ПС для контроля и оповещения о превышениях ПДК различных газов в помещении; Дата сдачи: 30.11.2011
1	Иванов Иван Иванович	Программист	Дата сдачи: 30.11.2011 ID: 321; Название: Модуль распознавания лиц для защитной системы; Дата сдачи: 01.12.2011



Приводим к 1-ой НФ

Код сотрудника	ФИО	Должность	ID проекта	Название проекта	Дата сдачи
1	Иванов Иван Иванович	Программист	123	Система управления паровым котлом	30.09.2011
1	Иванов Иван Иванович	Программист	231	ПС для контроля и оповещения о превышениях ПДК различных газов в помещении	30.11.2011
1	Иванов Иван Иванович	Программист	321	Модуль распознавания лиц для защитной системы	01.12.2011

Код сотрудника	ФИО	Должность	ID проекта	Название проекта	Дата сдачи
1	Иванов Иван Иванович	Программист	123	Система управления паровым котлом	30.09.2011
1	Иванов Иван Иванович	Программист	231	ПС для контроля и оповещения о превышениях ПДК различных газов в помещении	30.11.2011
1	Иванов Иван Иванович	Программист	321	Модуль распознавания лиц для защитной системы	01.12.2011

Проект



Должность

ID проекта	Название проекта	Дата сдачи проекта
123	Система управления паровым котлом	30.09.2011
231	ПС для контроля и оповещения о превышениях ПДК различных газов в помещении	30.11.2011
321	Модуль распознавания лиц для защитной системы	01.12.2011

Код должности	Должность
1	программист
2	наладчик
3	оператор

Проект_Сотрудник

Сотрудник

Код сотрудника	ФИО сотрудника	Код должности
1	Иванов Иван Иванович	1
2	Петров Петр Петрович	1
3	Сидоров Сидор Сидорович	3

Код записи	Код сотрудника	Код проекта
1	1	123
2	1	321
3	2	231
4	3	321
5	3	123



Пример 2

Рейс	Маршрут	Тип самолета	Кол-во мест
3437	Москва - НьюВасюки	ИЛ-62	180
23-ис	Москва - Черноморск	ЯК-42	120
777	Москва - Гряжск	ЯК-42	120

Рейс	Маршрут	Тип самолета	Кол-во мест
3437	Москва - НьюВасюки	ИЛ-62	180
23-ис	Москва - Черноморск	ЯК-42	120
777	Москва - Грязск	ЯК-42	120



Маршрут

Код маршрута	Маршрут
1	Москва - НьюВасюки
2	Москва - Черноморск
3	Москва - Грязск

Самолет

Код типа самолета	Тип самолета	Кол-во мест
1	ИЛ-62	180
2	ЯК-42	120

Рейс

Код записи	Рейс	Код маршрута	Код типа самолета
1	3437	1	1
2	23-ис	2	2
3	777	3	2