

Моделирование информационного обеспечения.

Моделирование данных

Одной из основных частей информационного обеспечения является **информационная база**.

Информационная база (ИБ) представляет собой совокупность данных, организованная определенным способом и хранимая в памяти вычислительной системы в виде файлов, с помощью которых удовлетворяются информационные потребности управленческих процессов и решаемых задач.

Разработка БД выполняется с помощью моделирования данных. Цель моделирования данных состоит в обеспечении разработчика ИС концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных. Наиболее распространенным средством моделирования данных являются диаграммы "сущность-связь" (ERD). С помощью ERD осуществляется детализация накопителей данных DFD – диаграммы, а также документируются информационные аспекты бизнес-системы, включая идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их связей с другими объектами (отношений).

Базовые понятия ERD

Сущность (Entity) — множество экземпляров реальных или абстрактных объектов (людей, событий, состояний, идей, предметов и др.), обладающих общими **атрибутами** или **характеристиками**. Любой объект системы может быть представлен только одной **сущностью**, которая должна быть уникально идентифицирована. При этом **имя сущности** должно отражать тип или класс объекта, а не его конкретный экземпляр (например, АЭРОПОРТ, а не ВНУКОВО).

Каждая **сущность** должна обладать уникальным **идентификатором**. Каждый **экземпляр сущности** должен однозначно идентифицироваться и отличаться от всех других экземпляров данного **типа сущности**.

Свойства сущности:

- иметь уникальное имя; 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;
- иметь один или несколько атрибутов, которые либо принадлежат сущности, либо наследуются через связь;
- иметь один или несколько атрибутов, которые однозначно идентифицируют каждый экземпляр сущности.

Каждая сущность может обладать любым количеством связей с другими сущностями модели.

Связь (Relationship) — поименованная ассоциация между двумя **сущностями**, значимая для рассматриваемой предметной области. **Связь** — это ассоциация между **сущностями**, при которой каждый экземпляр одной сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, и наоборот.

Атрибут (Attribute) — любая характеристика **сущности**, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния **сущности**. **Атрибут** представляет тип характеристик или свойств, ассоциированных с множеством реальных или абстрактных объектов (людей, мест, событий, состояний, идей, предметов и т.д.).

Экземпляр атрибута — это определенная характеристика отдельного элемента множества. **Экземпляр атрибута** определяется типом характеристики и ее значением, называемым значением атрибута. На диаграмме "сущность-связь" атрибуты ассоциируются с конкретными сущностями. Таким образом, экземпляр сущности должен обладать единственным определенным значением для ассоциированного атрибута.

Наиболее распространенными методами для построения ERD-диаграмм являются метод Баркера и метод IDEF1.

Метод IDEF1

- Метод Баркера основан на нотации, предложенной автором, и используется в case-средстве Oracle Designer.
- **Метод IDEF1** основан на подходе Чена и позволяет построить **модель данных**, эквивалентную реляционной модели в **третьей нормальной форме**. На основе совершенствования метода IDEF1 создана его новая версия — метод IDEFIX, разработанный с учетом таких требований, как простота для изучения и возможность автоматизации. IDEFIX-диаграммы используются в ряде распространенных CASE-средств (в частности, ERwin, Design/IDEF).
- В методе IDEFIX **сущность** 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** ([рис. 10.1](#), [10.2](#)).

Имя сущности/ Номер
сущности

Служащий/44

Независимые от идентификации сущности

Имя сущности/ Номер
сущности

Проектное задание/56

Зависимые от идентификации сущности

Каждой **сущности** присваиваются уникальные имя и номер, разделяемые косой чертой "/" и помещаемые над блоком.

Связь может дополнительно определяться с помощью указания степени или мощности (количества **экземпляров сущности-потомка**, которое может породить каждый **экземпляр сущности-родителя**). В IDEFIX могут быть выражены следующие *мощности связей*:

- каждый **экземпляр сущности-родителя** может иметь ноль, один или более одного связанного с ним *экземпляра сущности-потомка*;
- каждый **экземпляр сущности-родителя** должен иметь не менее одного связанного с ним *экземпляра сущности-потомка*;
- 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*.

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

Связь изображается линией, проводимой между сущностью-родителем и сущностью-потомком, с точкой на конце линии у сущности-потомка ([рис. 10.3](#)). *Мощность связей* может принимать следующие значения: N — ноль, один или более, Z — ноль или один, P — один или более. По умолчанию **мощность связей** принимается равной N.



Графическое изображение
мощности связи

Идентифицирующая **связь между сущностью-родителем и сущностью-потомком** изображается сплошной линией. **Сущность-потомок** в идентифицирующей **связи** является зависимой от идентификатора **сущностью**. **Сущность-родитель** в идентифицирующей **связи** может быть как независимой, так и зависимой от идентификатора **сущностью** (это определяется ее **связями с другими сущностями**).

Пунктирная линия изображает неидентифицирующую **связь** ([рис. 10.4](#)). Сущность-потомок в неидентифицирующей **связи** будет независимой от идентификатора, если она не является также сущностью-потомком в какой-либо идентифицирующей **связи**.

Атрибу
опреде
других

Сущно
в каче
обозна
которм



буть,
от

аться
Для
после

Неидентифицирующая
связь

ERwin имеет два уровня представления модели — логический и физический.

Логический уровень — это абстрактный взгляд на данные, когда данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире, например "Постоянный клиент", "Отдел" или "Фамилия сотрудника".

Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами.

Логическая модель данных может быть построена на основе другой логической модели, например на основе модели процессов. Логическая модель данных является универсальной и никак не связана с конкретной реализацией СУБД.

Физическая модель данных, напротив, зависит от конкретной СУБД, фактически являясь отображением системного каталога.

В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД.

Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах — таблицах, колонках, индексах, процедурах и т.д.

Создание логической модели данных

Уровни логической модели

Различают три уровня **логической модели**, отличающихся по глубине **представления** информации о данных:

- **диаграмма сущность-связь** (Entity Relationship Diagram, ERD);
- **модель данных, основанная на ключах** (Key Based model, KB);
- **полная атрибутивная модель** (Fully Attributed model, FA).

Диаграмма сущность-связь представляет собой **модель данных** верхнего уровня. Она включает **сущности** и **взаимосвязи**, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные **сущности** и **связи** между ними, которые удовлетворяют основным требованиям, предъявляемым к ИС. **Диаграмма сущность-связь** может включать **связи "многие-ко-многим"** и не включать описание ключей. Как правило, ERD используется для презентаций и обсуждения структуры данных с экспертами предметной области.

Модель данных, основанная на ключах, — более подробное **представление** данных. Она включает описание всех **сущностей** и **первичных ключей** и предназначена для **представления** структуры данных и ключей, которые соответствуют предметной области.

Полная атрибутивная модель — наиболее детальное **представление** структуры данных: представляет данные в **третьей нормальной форме** и включает все **сущности**, **атрибуты** и **связи**.

Сущности и атрибуты

Основные компоненты диаграммы ERwin — это **сущности, атрибуты и связи**.

Каждая **сущность** является множеством подобных индивидуальных объектов, называемых экземплярами. Каждый **экземпляр** индивидуален и должен отличаться от всех остальных экземпляров. **Атрибут** выражает определенное свойство объекта. С точки зрения БД(физическая модель) **сущности** соответствует таблица, экземпляру **сущности** — строка в таблице, а **атрибуту** — колонка таблицы.

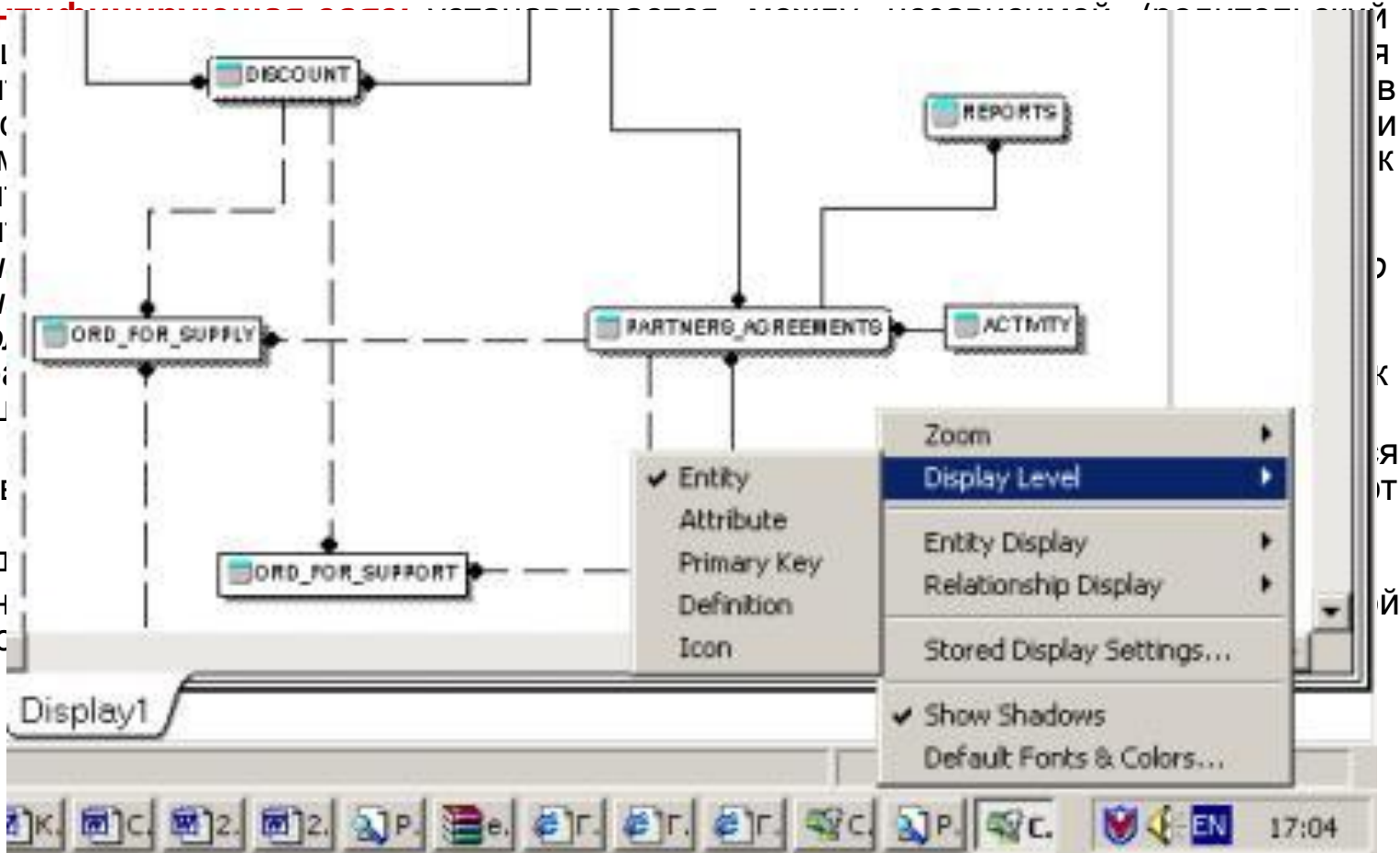
Сущности и атрибуты

- Как было указано выше, *каждый атрибут* хранит информацию об **определенном свойстве сущности**, а каждый *экземпляр сущности* должен быть уникальным. *Атрибут* или группа *атрибутов*, которые идентифицируют *сущность*, называется **первичным ключом**.
- Очень важно дать *атрибуту* **правильное имя**. *Атрибуты* должны именоваться в единственном числе и иметь четкое смысловое значение. Соблюдение этого правила позволяет частично решить проблему нормализации данных уже на этапе определения *атрибутов*.
- **Например**, создание в сущности *Сотрудник* атрибута *Телефоны* *сотрудника* противоречит требованиям нормализации, поскольку *атрибут* должен быть атомарным, т. е. не содержать множественных значений. Согласно синтаксису IDEFIX имя *атрибута* должно быть уникально в рамках модели (а не только в рамках *сущности*!). По умолчанию при попытке внесения уже существующего имени *атрибута* ERwin переименовывает его.

СВЯЗИ

- **Связь** является логическим соотношением между **сущностями**. Каждая **связь** должна именоваться глаголом или глагольной фразой. **Имя связи** выражает некоторое ограничение или бизнес-правило и облегчает чтение диаграммы. По умолчанию **ИМЯ СВЯЗИ** на диаграмме не показывается. На логическом уровне можно установить идентифицирующую связь "один-ко-многим", связь "многие-ко-многим" и неидентифицирующую связь "один-ко-многим".
- В IDEFIX различают **зависимые** и **независимые сущности**. **Тип сущности** определяется ее **связью** с другими **сущностями**.

- Иден
конел
иден
завис
углам
роди
иден
ключ
ключ
допо.
мигра
внеш
- При
неза
в
Неид
- Иден
точк



СВЯЗИ

Мощность связей (Cardinality) — служит для обозначения отношения числа экземпляров родительской *сущности* к числу экземпляров дочерней.

Различают четыре **типа сущности**:

- общий случай, когда одному экземпляру родительской *сущности* соответствуют 0, 1 или много экземпляров дочерней *сущности*; не помечается каким-либо символом;
- символом P помечается случай, когда одному экземпляру родительской *сущности* соответствуют 1 или много экземпляров дочерней *сущности* (исключено нулевое значение);
- символом Z помечается случай, когда одному экземпляру родительской *сущности* соответствуют 0 или 1 экземпляр дочерней *сущности* (исключены множественные значения);
- цифрой помечается случай точного соответствия, когда одному экземпляру родительской *сущности* соответствует заранее заданное число экземпляров дочерней *сущности*.

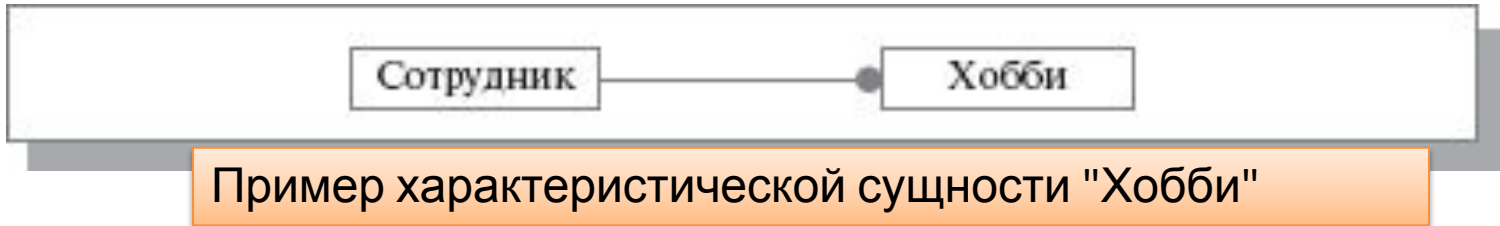
Имя связи (Verb Phrase) — фраза, характеризующая отношение между родительской и дочерней *сущностями*. Для связи "один-ко-многим", идентифицирующей или неидентифицирующей, достаточно указать имя, характеризующее отношение от родительской к дочерней *сущности* (Parent-to-Child). Для связи многие-ко-многим следует указывать имена как Parent-to-Child, так и Child-to-Parent.

Типы сущностей и иерархия

наследования

Как было указано выше, связи определяют, является ли *сущность* независимой или зависимой. Различают несколько **типов зависимых сущностей**.

Характеристическая — зависимая дочерняя *сущность*, которая связана только с одной родительской и по смыслу хранит информацию о характеристиках родительской *сущности* ([рис. 10.7](#)).



- **Ассоциативная** — *сущность*, связанная с несколькими родительскими *сущностями*. Такая *сущность* содержит информацию о связях *сущностей*.
- **Именующая** — частный случай ассоциативной *сущности*, не имеющей собственных *атрибутов* (только *атрибуты* родительских *сущностей*, мигрировавших в качестве внешнего ключа).
- **Категориальная** — дочерняя *сущность* в иерархии наследования.

- **Иерархия наследования** (или иерархия категорий) представляет собой особый тип объединения *сущностей*, которые разделяют общие характеристики. Например, в организации работают служащие, занятые полный рабочий день (постоянные служащие), и совместители. Из их общих свойств можно сформировать обобщенную *сущность* (родовой предок) Сотрудник ([рис. 10.8](#)), чтобы представить информацию, общую для всех типов служащих. Специфическая для каждого типа информация может быть расположена в категориальных *сущностях* (потомках) Постоянный сотрудник и Совместитель.
- Обычно **иерархию наследования** создают, когда несколько *сущностей* имеют общие по смыслу *атрибуты*, либо когда *сущности* имеют общие по смыслу *связи* (например, если бы Постоянный сотрудник и Совместитель имели сходную по смыслу *связь* "работает в" с *сущностью* Организация), либо когда это диктуется бизнес-правилами.
- Для каждой категории можно указать дискриминатор — *атрибут* родового предка, который показывает, как отличить одну категориальную *сущность* от другой (*атрибут* Тип на [рис. 10.8](#)).

Иерархия наследования. Неполная категория



Если категория еще не выстроена полностью и в родовом предке могут существовать экземпляры, которые не имеют соответствующих экземпляров в потомках, то такая категория будет неполной. На [рис. 10.8](#) показана неполная категория — сотрудник может быть не только постоянным или совместителем, но и консультантом, однако *сущность* Консультант еще не внесена в иерархию наследования.

Иерархия наследования. Полная категория



Иерархии категорий делятся на два типа — полные и неполные. В полной категории одному экземпляру родового предка (*сущность* С_п, [рис. 10.9](#)) обязательно соответствует экземпляр в каком-либо потомке, т. е. в примере сотрудник обязательно является либо совместителем, либо консультантом, либо постоянным сотрудником.

Литература

- <https://www.intuit.ru/studies/courses/2195/5/5/lecture/1636>
- <http://5fan.ru/wievjob.php?id=17459>