

## 57. Инфологическая модель. Цели и основные понятия инфологического моделирования.

Классификация сущностей. Ключи. Ограничения целостности.

Классификация сущностей: <http://bourabai.kz/dbt/project/2-3.htm>

Инфологическая модель: <http://www.intuit.ru/studies/courses/1001/297/lecture/7411>  
<http://any-book.org/download/8895.html>

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

**Сущность** – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как *тип сущности* и *экземпляр сущности*. Понятие типа сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе. Например, типом сущности может быть ГОРОД, а экземпляром – Москва, Киев и т.д.

**Атрибут** – поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей (например, ЦВЕТ может быть определен для многих сущностей: СОБАКА, АВТОМОБИЛЬ, ДЫМ и т.д.). Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Примерами атрибутов для сущности АВТОМОБИЛЬ являются ТИП, МАРКА, НОМЕРНОЙ ЗНАК, ЦВЕТ и т.д. Здесь также существует различие между типом и экземпляром. Тип атрибута ЦВЕТ имеет много экземпляров или значений: Красный, Синий, Банановый, Белая ночь и т.д., однако каждому экземпляру сущности присваивается только одно значение атрибута.

Абсолютное различие между типами сущностей и атрибутами отсутствует. Атрибут является таковым только в связи с типом сущности. В другом контексте атрибут может выступать как самостоятельная сущность. Например, для автомобильного завода цвет – это только атрибут продукта производства, а для лакокрасочной фабрики цвет – тип сущности.

**Ключ** – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Для сущности Расписание (п. 1.2) ключом является атрибут Номер\_рейса или набор: Пункт\_отправления, Время\_вылета и Пункт\_назначения (при условии, что из пункта в пункт вылетает в каждый момент времени один самолет).

**Связь** – ассоциирование двух или более сущностей. Если бы назначением базы данных было только хранение отдельных, не связанных между собой данных, то ее структура могла бы быть очень простой. Однако одно из основных требований к организации базы данных – это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи. А так как в реальных базах данных нередко содержатся сотни или даже тысячи сущностей, то теоретически между ними может быть установлено более миллиона связей. Наличие такого множества связей и определяет сложность инфологических моделей.

К.Дейт определяет три основных класса сущностей:

стержневые, ассоциативные и характеристические, а также подкласс ассоциативных сущностей –обозначения.

**Ассоциативная сущность** (ассоциация) – это связь вида "многие-ко-многим" ("-ко-многим" и т.д.) между двумя или более сущностями или экземплярами сущности (как в [примере 2.4](#)). Ассоциации рассматриваются как полноправные сущности: они могут участвовать в других ассоциациях и обозначениях точно так же, как стержневые сущности;

могут обладать свойствами, т.е. иметь не только набор ключевых атрибутов, необходимых для указания связей, но и любое число других атрибутов, характеризующих связь. Например, ассоциации "Брак" из примеров [2.1](#) и [2.4](#) содержат ключевые атрибуты "Код\_М", "Код\_Ж" и "Табельный номер мужа", "Табельный номер жены", а также уточняющие атрибуты "Номер свидетельства", "Дата регистрации", "Место\_регистрации", "Номер записи в книгу ЗАГС" и т.д.

**Характеристическая сущность** (характеристика) – это связь вида "многие-к-одной" или "одна-к-одной" между двумя сущностями (частный случай ассоциации). Единственная цель характеристики в рамках рассматриваемой предметной области состоит в описании или уточнении некоторой другой сущности. Необходимость в них возникает в связи с тем, что сущности реального мира имеют иногда многозначные свойства. Муж может иметь несколько жен (пример 2.3), книга – несколько характеристик переиздания (исправленное, дополненное, переработанное, ...) и т.д. Существование характеристики полностью зависит от характеризуемой сущности: женщины лишаются статуса жен, если умирает их муж.

Для описания характеристики используется новое предложение ЯИМ, имеющее в общем случае вид:

ХАРАКТЕРИСТИКА (атрибут 1, атрибут 2, ...) {СПИСОК ХАРАКТЕРИЗУЕМЫХ СУЩНОСТЕЙ}.
---

Расширим также язык ER-диаграмм, введя для изображения характеристики трапецию (рис. 2.2).



Рис. 2.2. Элементы расширенного языка ER-диаграмм

**Обозначающая сущность** или обозначение – это связь вида "многие-к-одной" или "одна-к-одной" между двумя сущностями и отличается от характеристики тем, что не зависит от обозначаемой сущности.

Рассмотрим пример, связанный с зачислением сотрудников в различные отделы организации.

При отсутствии жестких правил (сотрудник может одновременно зачисляться в несколько отделов или не зачисляться ни в один отдел) необходимо создать описание с ассоциацией Зачисление:

Отделы (Номер отдела, Название отдела, ...) Служащие (Табельный номер, Фамилия, ...) Зачисление [Отделы М, Служащие N] (Номер отдела, Табельный номер, Дата зачисления).
---

Однако, при условии, что каждый из сотрудников должен быть обязательно зачислен в один из отделов, можно создать описание с обозначением Служащие:

Отделы (Номер отдела, Название отдела, ...) Служащие (Табельный номер, Фамилия, ... , Номер отдела, Дата зачисления)[Отделы]
--

В данном примере служащие имеют независимое существование (если удаляется отдел, то из этого не следует, что также должны быть удалены служащие такого отдела). Поэтому они не могут быть характеристиками отделов и названы обозначениями.

Обозначения используют для хранения повторяющихся значений больших текстовых атрибутов: "кодификаторы" изучаемых студентами дисциплин, наименований организаций и их отделов, перечней товаров и т.п.

Описание обозначения внешне отличается от описания характеристики только тем, что обозначаемые сущности заключается не в фигурные скобки, а в квадратные:

ОБОЗНАЧЕНИЕ (атрибут 1, атрибут 2, ...)[СПИСОК ОБОЗНАЧАЕМЫХ СУЩНОСТЕЙ].
--

Как правило, обозначения не рассматриваются как полноправные сущности, хотя это не привело бы к какой-либо ошибке.

Обозначения и характеристики не являются полностью независимыми сущностями, поскольку они предполагают наличие некоторой другой сущности, которая будет "обозначаться" или "характеризоваться". Однако они все же представляют собой частные случаи сущности и могут, конечно, иметь свойства, могут участвовать в ассоциациях, обозначениях и иметь свои собственные (более низкого уровня) характеристики. Подчеркнем также, что все экземпляры характеристики должны быть обязательно связаны с каким-либо экземпляром характеризуемой сущности. Однако допускается, чтобы некоторые экземпляры характеризуемой сущности не имели связей. Правда, если это касается браков, то сущность "Мужья" должна быть заменена на сущность "Мужчины" (нет мужа без жены).

## Ключ

Напомним, что *ключ* или *возможный ключ* – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Каждая сущность обладает хотя бы одним возможным ключом. Один из них принимается за *первичный ключ*. При выборе первичного ключа следует отдавать предпочтение несоставным ключам или ключам, составленным из минимального числа атрибутов. Нецелесообразно также использовать ключи с длинными текстовыми значениями (предпочтительнее использовать целочисленные атрибуты). Так, для идентификации студента можно использовать либо уникальный номер зачетной книжки, либо набор из фамилии, имени, отчества, номера группы и может быть дополнительных атрибутов, так как не исключено появление в группе двух студентов (а чаще студенток) с одинаковыми фамилиями, именами и отчествами. Плохо также использовать в качестве ключа не номер блюда, а его название, например, "Закуска из плавленых сырков "Дружба" с ветчиной и соленым огурцом" или "Заяц в сметане с картофельными крокетами и салатом из красной капусты".

Не допускается, чтобы первичный ключ стержневой сущности (любой атрибут, участвующий в первичном ключе) принимал неопределенное значение. Иначе возникнет противоречивая ситуация: появится не обладающий индивидуальностью, и, следовательно не существующий экземпляр стержневой сущности. По тем же причинам необходимо обеспечить уникальность *первичного ключа*.

Теперь о *внешних ключах*:

- Если сущность С связывает сущности А и В, то она должна включать внешние ключи, соответствующие первичным ключам сущностей А и В.
  - Если сущность В обозначает сущность А, то она должна включать внешний ключ, соответствующий первичному ключу сущности А.

В п. 2.3 рассматривался пример, где "Служащие" обозначали "Отделы" и включали внешний ключ "Номер отдела", соответствующий первичному ключу сущности "Отделы".

Связь между первичными и внешними ключами сущностей иллюстрируется рис. 2.5.

Рис. 2.5. Структуры: а - ассоциации; б - обозначения (характеристики)

Здесь для обозначения любой из ассоциируемых сущностей (стержней, характеристик, обозначений или даже ассоциаций) используется новый обобщающий термин "Цель" или "Целевая сущность".

Таким образом, при рассмотрении проблемы выбора способа представления ассоциаций и обозначений в базе данных основной вопрос, на который следует получить ответ: "Каковы внешние ключи?". И далее, для каждого внешнего ключа необходимо решить три вопроса:

1. Может ли данный внешний ключ принимать неопределенные значения (NULL-значения)? Иначе говоря, может ли существовать некоторый экземпляр сущности данного типа, для которого неизвестна целевая сущность, указываемая внешним ключом? В случае поставок это, вероятно, невозможно – поставка, осуществляемая неизвестным поставщиком, или поставка неизвестного продукта не имеют смысла. Но в случае с сотрудниками такая ситуация однако могла бы иметь смысл – вполне возможно, что какой-либо сотрудник в данный момент не зачислен вообще ни в какой отдел. Заметим, что ответ на данный вопрос не зависит от прихоти проектировщика базы данных, а определяется фактическим образом действий, принятым в той части реального мира, которая должна быть представлена в рассматриваемой базе данных. Подобные замечания имеют отношение и к вопросам, обсуждаемым ниже.

2. Что должно случиться при попытке УДАЛЕНИЯ целевой сущности, на которую ссылается внешний ключ? Например, при удалении поставщика, который осуществил по крайней мере одну поставку. Существует три возможности:

КАСКАДИРУЕТСЯ	Операция удаления "каскадируется" с тем, чтобы удалить также поставки этого поставщика.
ОГРАНИЧИВАЕТСЯ	Удаляются лишь те поставщики, которые еще не осуществляли поставок. Иначе операция удаления отвергается.
УСТАНАВЛИВАЕТСЯ	Для всех поставок удаляемого поставщика NULL-значение внешний ключ устанавливается в неопределенное значение, а затем этот поставщик удаляется. Такая возможность, конечно, неприменима, если данный внешний ключ не должен содержать NULL-значений.

3. Что должно происходить при попытке ОБНОВЛЕНИЯ первичного ключа целевой сущности, на которую ссылается некоторый внешний ключ? Например, может быть предпринята попытка обновить номер такого поставщика, для которого имеется по крайней мере одна соответствующая поставка. Этот случай для определенности снова рассмотрим подробнее. Имеются те же три возможности, как и при удалении:

КАСКАДИРУЕТСЯ	Операция обновления "каскадируется" с тем, чтобы обновить также и внешний ключ в поставках этого поставщика.
ОГРАНИЧИВАЕТСЯ	Обновляются первичные ключи лишь тех поставщиков, которые еще не осуществляли поставок. Иначе операция обновления отвергается.
УСТАНАВЛИВАЕТСЯ	Для всех поставок такого поставщика NULL-значение внешний ключ устанавливается в неопределенное значение, а затем обновляется первичный ключ поставщика. Такая возможность, конечно, неприменима,

Таким образом, для каждого внешнего ключа в проекте проектировщик базы данных должен специфицировать не только поле или комбинацию полей, составляющих этот внешний ключ, и целевую таблицу, которая идентифицируется этим ключом, но также и ответы на указанные выше вопросы (три ограничения, которые относятся к этому внешнему ключу).

Наконец, о характеристиках – обозначающих сущностях, существование которых зависит от типа обозначаемых сущностей. Обозначение представляется внешним ключом в таблице, соответствующей этой характеристике. Но три рассмотренные выше ограничения на внешний ключ для данного случая должны специфицироваться следующим образом:

NULL-значения не допустимы  
УДАЛЕНИЕ ИЗ (цель) КАСКАДИРУЕТСЯ  
ОБНОВЛЕНИЕ (первичный ключ цели) КАСКАДИРУЕТСЯ

Указанные спецификации представляют зависимость по существованию характеристических сущностей.

### **Ограничение целостности**

Целостность (от англ. integrity – нетронутость, неприкосновенность, сохранность, целостность) – понимается как правильность данных в любой момент времени. Но эта цель может быть достигнута лишь в определенных пределах: СУБД не может контролировать правильность каждого отдельного значения, вводимого в базу данных (хотя каждое значение можно проверить на правдоподобность). Например, нельзя обнаружить, что вводимое значение 5 (представляющее номер дня недели) в действительности должно быть равно 3. С другой стороны, значение 9 явно будет ошибочным и СУБД должна его отвергнуть. Однако для этого ей следует сообщить, что номера дней недели должны принадлежать набору (1,2,3,4,5,6,7).

Поддержание целостности базы данных может рассматриваться как защита данных от неверных изменений или разрушений (не путать с незаконными изменениями и разрушениями, являющимися проблемой безопасности). Современные СУБД имеют ряд средств для обеспечения поддержания целостности (так же, как и средств обеспечения поддержания безопасности).

Выделяют три группы правил целостности:

1. Целостность по сущностям.
2. Целостность по ссылкам.
3. Целостность, определяемая пользователем.

Правила целостности, общие для любых реляционных баз данных.

1. Не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе, принимал неопределенное значение.
2. Значение внешнего ключа должно либо:
  1. быть равным значению первичного ключа цели;
  2. быть полностью неопределенным, т.е. каждое значение атрибута, участвующего во внешнем ключе должно быть неопределенным.
3. Для любой конкретной базы данных существует ряд дополнительных специфических правил, которые относятся к ней одной и определяются разработчиком. Чаще всего контролируется:

уникальность тех или иных атрибутов, диапазон значений (экзаменационная оценка от 2 до 5), принадлежность набору значений (пол "М" или "Ж").