**Лекция 3. Проектирование баз данных**

Инфологическое моделирование (иногда используется термин семантическое моделирование) применяется на втором этапе проектирования БД, то есть после системного анализа предметной области. На этапе системного анализа были сформированы понятия о предметах, фактах и событиях, которыми будет оперировать БД. Инфологическое проектирование связано с представлениемсемантики предметной области в модели БД, т.е. моделирование структур данных, опираясь на смысл этих данных. Инфологическое моделирование было предметом исследований в конце 1970-х и начале 1980-х годов. Было предложено несколько моделей данных, названных семантическими моделями.  
Наибольшее распространение получила модель "сущность-связь" (entity-relationshipmodel, ERмодель), предложенная в 1976 г. Питером Пин-ШэнЧеном.  
Модель «сущность-связь» является концептуальной моделью, т.е. не учитывает особенности  
конкретной СУБД. Из модели "сущность-связь" могут быть получены все основныефактографические модели данных (иерархическая, сетевая, реляционная, объектноориентированная).  
Модели "сущность-связь" удобны тем, что процесс создания модели является итерационным.  
Разработав первый приближенный вариант модели, можно уточнять ее, опрашивая экспертов  
предметной области. При этом документацией, в которой фиксируются результаты бесед, является сама модель "сущность-связь".

В этой главе мы рассмотрим основные понятия модели "сущность-связь":

* Сущности
* Атрибуты
* Ключевые атрибуты
* Связи
* Степени связей
* Классы принадлежности сущностей в связях

А также рассмотрим пример построения диаграммы "сущность-связь".

**2. Модель «сущность-связь»**

Основными понятиями модели "сущность-связь" являются: сущность, связь и атрибут.  
Любой фрагмент предметной области может быть представлен как *множество сущностей*,  
между которыми существует некоторое *множество связей*.

*Сущность* - это реальный или представляемый объект, информация о котором должна сохраняться в проектируемой системе.

Сущность имеет имя, уникальное в пределах системы. Сущность соответствует некоторому классу однотипных объектов, то есть в системе существует множество экземпляров данной сущности.

*Примеры сущностей*: люди, продукты, студенты и т.д. *Примеры экземпляров сущностей*: конкретный человек, конкретный продукт, конкретный студент и т.д.  
Сущности не обязательно должны быть непересекающимися. Например, экземпляр сущности  
СТУДЕНТ, также принадлежит сущности ЛЮДИ.

Объект, которому соответствует понятие сущности, имеет свой набор *атрибутов* –  
характеристик, определяющих свойства данного объекта. Атрибут должен иметь имя, уникальное впределах данной сущности.

*Пример:*

Рассмотрим множество пищевых продуктов, имеющихся в магазине. Каждый продукт можно  
представить следующими характеристиками: Код продукта, Продукт, , Срок хранения, Условия хранения.

В дальнейшем для определения сущности и ее атрибутов будем использовать обозначение  
видаПродукты(КодПродукта, Продукт, ЕдиницаИзмерения, СрокХранения, УсловияХранения)  
Например, поставщиков, поставляющих продукты в магазин, можно описать как Поставщики(КодПоставщика, Поставщик, Адрес)

Множество допустимых значений (область определения) атрибута называется *доменом*.

Например, атрибут *СрокХранения*хранит информацию о количестве дней, в течение которых  
продукт годен к употреблению. То есть этот атрибут принадлежит домену *КоличествоДней*, которыйзадается интервалом целых чисел больших нуля, поскольку количество дней отрицательным быть неможет.

Набор атрибутов сущности должен быть таким, чтобы можно было однозначно найти  
требуемый экземпляр сущности.

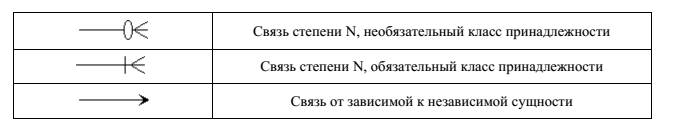
Например, сущность *Продукты* однозначно определяется атрибутом *КодПродукта*,  
поскольку все коды продуктов различны.

Отсюда определяется *ключ сущности* - это минимальный набор атрибутов, по значениям  
которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, чтоисключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.  
*Пример*. Сущность *Продажа* с атрибутами *ДатаПродажи*, *КодПродукта*, *Количество, Цена*содержит информацию о продаже продуктов за конкретный день. Для этой сущности ключом будутатрибуты *ДатаПродажи*и *КодПродукта*, поскольку за день могут быть проданы несколькопродуктов, а конкретный продукт может быть продан в разные дни. Исключение любого атрибута из ключа не позволит однозначно найти требуемый экземпляр сущности, т.е. будет нарушено условиеминимальности. Обозначим эту сущность какПродажа(ДатаПродажи, КодПродукта, Количество, Цена). Ключевой атрибут выделен подчеркиванием.

Между сущностями могут быть установлены связи.  
*Связь* - это ассоциация, установленная между несколькими сущностями, и показывающая как  
взаимодействуют сущности между собой.  
*Примеры*:  
• в магазине происходит продажа продуктов, т.е. между сущностями *Продукты* и *Продажа*существует связь «*происходит*» или *Продукты-Продажа* (обычно, но необязательно, связь  
обозначается глаголом или двойным названием сущностей, между которыми установлена  
эта связь)  
• так как продукты в магазин поставляют поставщики, то между сущностями *Продуты* и  
*Поставщики* существует связь «*поставка*» или *Продукты-Поставщики*• могут существовать и связи между экземплярами одной и тойже сущности (рекурсивная  
связь), например связь *Родитель-Потомок* между экземплярами сущности *Человек*Связь также может иметь атрибуты. Например, для связи *Продукты-Поставщики* можно  
задать атрибуты *ДатаПоставки*, *Цена* и т.д.  
Связь, существующая между двумя сущностями, называется *бинарной связью*.  
Связь, существующая между n сущностями, называется *n-арной связью*.  
*Рекурсивная связь* – это связь между экземплярами одной сущности.  
Доказано, что любую n-арную связь всегда можно заменить множеством бинарных, однако  
первые лучше отображают семантику предметной области.  
То число экземпляров сущностей, которое может быть ассоциировано через связь с  
экземплярами другой сущности, называют *степенью связи*. Рассмотрение степеней особенно  
полезно для бинарных связей.  
Очень важным свойством модели "сущность-связь" является то, что она может быть  
представлена в виде графической схемы (диаграммы). Это значительно облегчает анализ предметнойобласти. Существует несколько вариантов обозначения элементов диаграммы "сущность-связь"(нотаций). Для обозначения сущностей, связей и атрибутов будем использовать нотацию Чена, а для обозначения степеней и кардинальностей связей – нотацию Мартина.

В Таблице 2 приводится список используемых здесь обозначений.

Таблица 2. Список обозначений.



Рассмотрим теперь существующие степени бинарных связей:  
• один-к-одному (обозначается 1:1). Это означает, что в такой связи в каждый момент  
времени каждому экземпляру сущности A соответствует 1 или 0 экземпляров сущности B.  
Данный факт представлен на Рисунок 1, где прямоугольники обозначают сущности, а ромб -  
связь. Так как степень связи для каждой сущности равна 1, то они соединяются одной  
линией.

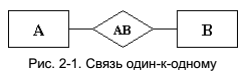


Рисунок 1. Связь один-к-одному

один-ко-многим (1:N). Одному экземпляру сущности A соответствуют 0, 1 или N  
экземпляров сущности B. Графически степень связи N отображается "древообразной"  
линией, так это сделано на следующем рисунке (Рисунок 2).

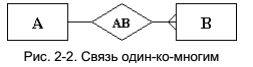


Рисунок 2. Связь один-ко-многим

многие-к-одному (N:1). Эта связь аналогична отображению 1:N. Одному экземпляру  
сущности B соответствуют 0, 1 или N экземпляров сущности A (Рисунок 3).

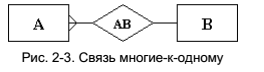


Рисунок 3. Связь многие-к-одному

многие-ко-многим (M:N). В этом случае одному экземпляру сущности A соответствуют 0,  
1 или N экземпляров сущности B, и наоборот, одному экземпляру сущности B  
соответствуют 0, 1 или N экземпляров сущности A (Рисунок 4).

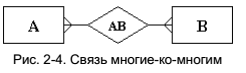


Рисунок 4. Связь многие-ко-многим

Другой важной характеристикой связи помимо ее степени является *класс принадлежности*входящих в нее сущностей. Существует *обязательный* и *необязательный* классы принадлежности.  
Рассмотрим *примеры* степени связи один-к-одному:  
1. Если рассматривать традиционный брак, то степень связи между сущностями *Мужчина* и  
*Женщина* будет 1:1. Кроме того, каждый мужчина, состоящий в браке, обязательно ДОЛЖЕН  
иметь жену. Таким образом, говорят, что сущность *Женщина* имеет *обязательный класс  
принадлежности*. И, наоборот, каждая женщина, состоящая в браке, ДОЛЖНА иметь мужа. То  
есть, сущность *Мужчина* также имеет *обязательный класс принадлежности* (2-5).



Рисунок 5. Связь 1:1, обязательные классы принадлежности.

Сотрудник руководит отделом (Рис. 6). Поскольку сотрудник может руководить только  
ОДНИМ отделом, а в отделе может быть только ОДИН руководитель, то степень связи в этом  
примере 1:1. Кроме того, в каждом отделе ДОЛЖЕН быть руководитель, т.е. каждому экземплярусущности *Отдел* ДОЛЖЕН соответствовать экземпляр сущности *Сотрудник* (сущность*Сотрудник* имеет обязательный класс принадлежности). С другой стороны, далеко не всесотрудники должны быть руководителями, т.е. сотрудник МОЖЕТ (но не должен) быть  
руководителем. Таким образом, есть экземпляры сущности *Сотрудник*, которым не соответствуетни один экземпляр сущности *Отдел* (необязательный класс принадлежности).



Рисунок 6. Связь 1:1, обязательный и необязательный классы принадлежости

Человек читает книгу (Рис. 7). Человек может читать сразу только ОДНУ книгу, а конкретная  
книга может быть читаема только ОДНИМ человеком, следовательно, степень связи 1:1. Человек  
МОЖЕТ читать книгу, а может ничего не читать. С другой стороны не каждая книга должна  
читаться, некоторые стоят на полке. Таким образом, обе сущности имеют необязательный класс  
принадлежности.



Рисунок 7. Связь 1:1, необязательные классы принадлежности

Рассмотрим примерыстепени связи один-ко-многим (многие-к-одному):  
4. В процессе обучения студенты объединены в группы. Каждая группа может содержать  
множество студентов, а каждый студент может входить только в одну группу, т.е. степень связи  
1:N (Рис. 8). Каждая группа ДОЛЖНА содержать студентов, а каждый студент ДОЛЖЕН быть  
зачислен в конкретную группу, т.е. обе сущности имеют обязательные классы принадлежности



Рисунок 8. Связь1:N, обязательные классы принадлежности

Поставщики продуктов имеют один юридический адрес, следовательно, ДОЛЖНЫ находиться в  
одном конкретном городе. А в одном городе МОГУТ находиться один, несколько или ни одного  
поставщика. Т.е. связь будет N:1, сущность Города будет иметь обязательный, а сущность  
Поставщики – необязательный классы принадлежности (Рис. 9).

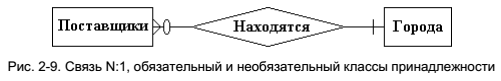


Рисунок 9. Связь 1:N, обязательный и необязательный классы принадлежности

Если существование сущности *x* зависит от существования сущности *y*, то *x* называется  
зависимой сущностью (иногда сущность *x* называют "слабой", а "сущность" *y - сильной).*Степень связи для сильной сущности всегда будет 1 и обязательный класс принадлежности.  
Класс принадлежности и степень связи для зависимой сущности могут быть любыми.  
6. В магазине происходит продажа продуктов. Продукт не может быть продан, если его нет в  
магазине. Поэтому сущность *Продажи* является зависимой от сущности *Продукты* (Рис. 2-10).  
Продукт МОЖЕТ быть продан в разные дни (а может быть вообще не продан), конкретная  
продажа связана только с одним продуктом. Таким образом, степень связи N:1, сущность  
*Продажи* имеет необязательный, а сущность *Продукты* - обязательный классы принадлежности  
(в самом деле, продажа без продукта теряет смысл).



Рисунок 10. Зависимая и независимая сущности

Рассмотрим *пример* степени связи многие-ко-многим: Продукты в магазин поставляются поставщиками. Каждый продукт, имеющийся в магазине,  
ДОЛЖЕН быть поставлен одним или несколькими поставщиками, а каждый из поставщиков  
МОЖЕТ поставлять один или несколько продуктов или не поставлять ни одного. Т.е. степень  
связи M:N (Рис. 11), а класс принадлежности для *Поставщиков* – обязательный, для *Продуктов*– необязательный.



Рисунок 11. Связь M:N, обязательный и необязательный классы принадлежности

Между одними и теми же сущностями могут существовать несколько связей, например:  
8. С одной стороны продукты в магазин поставляются заказчиками, с другой стороны, чтобы  
продукты были поставлены в магазин, необходимо заказать поставщикам необходимые  
продукты. Таким образом, между сущностями *Продукты* и *Поставщики* существуют связи  
«*Поставляют*» и «*Заказаны*» (Рис. 12). Связь «*Поставляют*» рассмотрена в предыдущем  
примере. Рассмотрим подробнее связь «*Заказаны*». Каждый продукт ДОЛЖЕН быть заказан  
одному или нескольким поставщикам, каждый поставщик МОЖЕТ получить заказ на один или  
несколько продуктов или вообще не получить заказ.

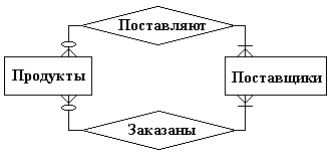


Рисунок 12. Несколько связей между двумя сущностями

Рассмотрим сущности *Врач* и *Пациент*. Пациент ДОЖЕН иметь одного лечащего врача, а врач  
МОЖЕТ лечить несколько пациентов. Кроме того, пациент МОЖЕТ иметь нескольких врачей  
консультантов, а врач МОЖЕТ консультировать нескольких пациентов (Рис.2-13).



Рисунок 13. Несколько связей между двумя сущностями

***Пример построения модели «сущность-связь»***В процессе построения диаграммы "сущность-связь" можно выделить несколько этапов:

* Определение списка сущностей выбранной предметной области
* Определение списка атрибутов сущностей
* Описание связей между сущностями (степени, классы принадлежности связей, а также  
  атрибуты связей, если они необходимы)
* Организация данных в виде диаграммы "сущность-связь"

В качестве примера построим диаграмму, отображающую связь данных для информационной  
системы учета продажи продуктов в магазине. БД должна хранить информацию о продуктах,  
поставляемых в магазин, их ежедневной продаже, заказах на поставку продуктов, а также о  
поставщиках продуктов.  
Составим список сущностей, необходимых для реализации поставленной задачи:

1. Продукты  
   Для этой сущности необходимы следующие атрибуты:

* *Код продукта* – уникальный идентификатор, ключевой атрибут
* *Продукт* – название продукта
* *Единица измерения* – литры, килограммы, штуки и т.п.
* *Срок хранения в днях* – для определения даты окончания срока годности продукта
* *Условия хранения* – температура, влажность и т.п.

1. Поставщики

* *Код поставщика* – уникальный идентификатор, ключевой атрибут
* *Поставщик* – название организации или ФИО физического лица
* *Код города* – выделим отдельно город, где находится поставщик, для удобства дальнейшей  
  работы (например, для поиска)
* *Адрес* – поскольку город выделен в отдельный атрибут, то в адресе остается улица и дом (а  
  также квартира – для физического лица)
* ФИО директора
* Телефон
* Факс
* Продажи
* Дата продажи
* Код продукта – какой именно продукт был продан
* Количество – сколько продано этого продукта в тех единицах измерения, которые указаны  
  для этого продукта в сущности Продукт
* Цена продажи – цена при продаже за единицу продукта
* Города – поскольку мы выделили отдельно город из адреса поставщика, то возникает  
  необходимость в этой сущности
* *Код города* – уникальный идентификатор, ключевой атрибут

1. *Город*

Сократив для удобства названия атрибутов, получим список сущностей:

* Продукты(КодПрод, Продукт, ЕдИзм, СрокХран(дней), УсловияХран)
* Поставщики(КодПост, Поставщик, КодГорода, Адрес, ФИОдиректора, Телефон, Факс)
* Продажи(ДатаПродажи, КодПрод, Количество, ЦенаПродажи)  
  обратите внимание, что в этой сущности ключ составной, поскольку каждый день  
  продается множество продуктов, и конкретный продукт может быть продан в разные дни
* Города(КодГорода, Город)

Рассмотрим связи, существующие между описанными выше сущностями:

1. Продукты в магазин поставляются поставщиками, т.е. существует связь M:N «Поставляют» между сущностями Продукты и Поставщики (подробно эта связь рассмотрена в примере Рис. 11). Эта связь имеет следующие атрибуты:

* Дата поставки
* Код поставщика – какой поставщик поставил этот продукт
* Код продукта – какой именно продукт был поставлен
* КоличествоП – сколько поставлено этого продукта в тех единицах измерения, которые указаны для этого продукта в сущности Продукт
* Цена поставки – цена при поставке за единицу продукта
* Дата изготовления – дата изготовления продукта  
  Ключом будет составной атрибут: *Дата поставки, Код поставщика, Код продукта*(объясните, почему именно эти атрибуты вошли в составной ключ)

1. Продукты должны быть заказаны поставщикам, т.е. существует связь M:N «Заказаны»

между сущностями Продукты и Поставщики (подробно эта связь рассмотрена в примере

Рис. 12). Эта связь имеет следующие атрибуты:

* Дата заказа
* Код поставщика – какому поставщику заказан этот продукт
* Код продукта – какой именно продукт был заказан
* КоличествоЗ – сколько поставлено этого продукта в тех единицах измерения, которые указаны для этого продукта в сущности Продукт

Ключом будет составной атрибут: *Дата заказа, Код поставщика, Код продукта*(объясните, почему именно эти атрибуты вошли в составной ключ)

1. В магазине происходит продажа продуктов, т.е. существует связь N:1 «Происходит» между  
   сущностями Продажи и Продукты (подробно эта связь рассмотрена в примере Рис. 10).
2. Поставщики находятся в определенном городе, т.е. существует связь N:1 «Находятся»  
   между сущностями Поставщики и Города (подробно эта связь рассмотрена в примере Рис. 9)  
   После объединения всех фрагментов в общую модель и добавления атрибутов, получится  
   диаграмма "сущность-связь", приведенная на Рис. 2-14.



Рисунок 14. Диаграмма «сущность-связь» учета продажи продуктов в магазине