

Проектирование БД

Теоретические пояснения
к курсовой работе

Часть 1

Описание информационного представления предметной области. ER-диаграмма

Иллюстрацию вводимых понятий и этапов проектирования базы данных будем проводить на примере близкой для читателя конкретной предметной области: представление данных о студентах вуза, Дадим краткое описание рассматриваемой предметной области. В вузе имеется несколько факультетов, на каждом из которых ведется подготовка по нескольким специальностям или направлениям. Для каждой специальности на факультете есть свой учебный план, в котором приводится перечень изучаемых учебных курсов с указанием количества часов занятий. Студенты изучают соответствующие дисциплины, сдают экзамены изачеты, получают оценки.

Чаще всего концептуальная модель представляется в виде диаграммы сущностей – связей (entity – relationship) или ER-диаграммы. Процесс построения ER-диаграммы называется ER-моделированием.

Введем основные понятия, с помощью которых описывается предметная область.

Сущность (Entity) или объект – то, о чем будет накапливаться информация в информационной системе (нечто такое, за чем пользователь хотел бы наблюдать).

Если в системе обрабатывается информация о факультетах, сущностью будет являться факультет, если о студентах, сущность – студент и т.п.

Имя сущности при ER-моделировании, как правило, записывается заглавными буквами. Каждая сущность обладает определенным набором свойств (рассматриваем только свойства, представляющие интерес для пользователей в рамках проводимого исследования), которые запоминаются в информационной системе. Так, например, в качестве свойств сущности ФАКУЛЬТЕТ можно указать номер факультета, название факультета, в качестве свойств сущности СТУДЕНТ можно указать фамилию, дату рождения, место рождения, в качестве свойств сущности ЭКЗАМЕН – предмет, дату проведения экзамена, экзаменаторов.

Для информационного описания сущности вводится понятие атрибута.

Атрибут – поименованное свойство (характеристика) сущности. Атрибут представляет собой информационное отображение свойства сущности и принимает конкретное значение из множества допустимых значений. Так, например, для сущности ФАКУЛЬТЕТ атрибут «название» у конкретного экземпляра сущности принимает конкретное значение «вычислительной математики и кибернетики». Таким образом, атрибут представляет информационное описание количественных или качественных свойств сущности, описывает состояние сущности, позволяет идентифицировать сущность. Информация о сущности представляется совокупностью атрибутов. Таковую совокупность атрибутов часто называют записью об объекте.

Совокупность сущностей, характеризующихся в информационной системе одним и тем же перечнем свойств, называется классом сущностей (набором объектов). Так, например, совокупность всех сущностей СТУДЕНТ составляет класс сущностей СТУДЕНТ, совокупность всех сущностей ФАКУЛЬТЕТ составляет класс сущностей ФАКУЛЬТЕТ. Класс сущностей

Экземплярм сущности будем называть конкретную сущность (сущность с конкретными значениями соответствующих свойств). Выше мы определили сущность как то, о чем будет накапливаться информация в информационной системе. Это только одна сторона. Информация должна не просто храниться сама по себе, а использоваться для удовлетворения информационных потребностей пользователя. Для реализации подавляющего числа запросов пользователю прежде всего необходимо найти интересующий его экземпляр сущности (с целью обработки, корректировки, удаления). Поэтому важнейшим свойством сущности является однозначная идентификация ее экземпляров по одному или группе атрибутов (уникальному идентификатору). У сущности ФАКУЛЬТЕТ это, например, номер факультета, у сущности СТУДЕНТ это может быть атрибут «фамилия», если у всех студентов разные фамилии, группа атрибутов «фамилия», «имя», «отчество», или специально введенный уникальный идентификатор, например дополнительно введенный атрибут «код студента».

Наиболее распространенным способом представления концептуальной модели является так называемая ER-диаграмма. В разных источниках используются разные системы обозначений в ER-диаграммах. На практике использование различных способов записи ER-диаграмм не представляет особой сложности – беглое ознакомление с соответствующим разделом документации позволяет быстро освоить используемую систему обозначений. В данном пособии в ER-диаграмме класс сущностей будем представлять в виде четырехугольника.

В четырехугольнике записано уникальное имя класса сущности (прописными буквами) и имена атрибутов строчными буквами.

Пример класса сущностей СТУДЕНТ и конкретного экземпляра сущности показан на рисунке.



Для реализации информационных потребностей пользователя недостаточно найти интересующий его экземпляр сущности. Информационные потребности тесно связаны с функциональными взаимоотношениями, существующими в организации (например, необходимо определить, на каком факультете учится конкретный студент). Для реализации таких запросов (информационных потребностей пользователя) используются существующие в предметной области взаимоотношения между сущностями. Соответствующие взаимоотношения сущностей выражаются связями (Relationships). Различают классы связей и экземпляры связей. Классы связей – это взаимоотношения между классами сущностей, а экземпляры связи – взаимоотношения между экземплярами сущностей.

Класс связей может затрагивать несколько классов сущностей. Число классов сущностей, участвующих в связи, называется степенью связи $n = 2, 3, \dots$. Так, например, класс сущностей СТУДЕНТ связан с классом сущностей ФАКУЛЬТЕТ связью «учится на факультете». Степень этой связи равна двум. При $n=2$ связь называется бинарной. Заметим, что связь нужно рассматривать как двустороннюю: «студент учится на

Рассмотрим классификацию бинарных связей. В зависимости от того, сколько экземпляров сущности одного класса связаны со сколькими экземплярами сущности другого класса, различают следующие типы связей:

- Связь 1:1. Одиночный экземпляр сущности одного класса связан с одиночным экземпляром сущности другого класса. Примером является связь между классами сущностей ФАКУЛЬТЕТ и УЧЕБНЫЙ ПЛАН ПО СПЕЦИАЛЬНОСТИ ДЛЯ ФАКУЛЬТЕТА (каждому факультету соответствует свой учебный план по специальности или направлению).
- Связь 1:M. Единый экземпляр сущности одного класса связан со многими экземплярами сущности другого класса. Примером является связь между классами сущностей ФАКУЛЬТЕТ и СТУДЕНТ (на одном факультете учатся много студентов).
- Связь M:N. Несколько экземпляров сущности одного класса связаны с несколькими экземплярами сущности другого класса. Примером является связь между классами сущностей ФАКУЛЬТЕТ и СПЕЦИАЛЬНОСТЬ (на факультете может быть несколько специальностей и одна и также специальность может быть на нескольких факультетах).

Числа, описывающие типы бинарных связей (1:1, 1:M, M:N), обозначают максимальное количество сущностей на каждой стороне связи. Эти числа называются максимальными кардинальными числами, а соответствующая пара чисел называется максимальной

- В данном пособии на ER-диаграммах связи между сущностями будем обозначать стрелками, рядом со стрелками указываем имя связи, а также тип связи. Пример ER-диаграммы, представляющего сущности **СТУДЕНТ**, **ФАКУЛЬТЕТ**, **СПЕЦИАЛЬНОСТЬ** и их взаимосвязи приводится на рисунке



Напомним, что каждый экземпляр сущности должен уникально идентифицироваться (иметь уникальный идентификатор). Так как могут быть несколько студентов с одинаковой фамилией, введем дополнительный атрибут «код студента». У сущностей ФАКУЛЬТЕТ и СПЕЦИАЛЬНОСТЬ атрибут «номер» является уникальным идентификатором.

Заметим, что по этой ER-диаграмме можно указать последовательность действий, производимых при реализации запроса пользователей. Например, для реализации запроса «на каком факультете учится студент Иванов» необходимо выполнить следующие действия: найти среди экземпляров сущности СТУДЕНТ экземпляр с фамилией Иванов, перейти по связи «Студент учится на факультете» к экземпляру сущности ФАКУЛЬТЕТ, значение атрибута «Название» этого экземпляра и есть искомое название факультета. Отметим также, что иногда на ER-диаграммах две связи между сущностями изображают одной двухсторонней стрелкой или просто линией. Заметим, что на приведенной ER-диаграмме не представлены

какие-либо способы реализации этих связей (на логическом и, тем более, на физическом уровнях). Соответствующие способы реализации связей зависят от возможностей модели данных конкретной СУБД и будут рассмотрены в следующей части на второй стадии концептуального проектирования при представлении концептуальной модели средствами модели данных СУБД.

Построение концептуальной модели в виде ER-диаграммы

Основные этапы построения

Как уже отмечалось, концептуальная модель представляет собой обобщение представлений разных пользователей о данных. В связи с этим построение концептуальной модели, как правило, происходит в два этапа. На первом этапе производится сбор и анализ характеристик данных и строятся так называемые модели локальных представлений (локальные модели). Чаще всего локальная модель отражает представление отдельного пользователя (отдельной функциональной задачи). Иногда такая модель может описывать и некоторую независимую область данных нескольких функциональных задач (нескольких приложений).

Здесь необходимо отметить, что моделирование представлений отдельных пользователей приводит к снижению уровня интеграции данных, а моделирование совместных представлений группы пользователей – к повышению сложности проектирования. В связи с этим при выборе области данных для локального моделирования приходится выбирать компромиссное решение между вышеуказанными вариантами.

При разработке концептуальной модели, прежде всего, следует определить сущности.

С этой целью нужно сделать следующее:

- необходимо понять, какая информация должна храниться и обрабатываться и можно ли это определить как сущность;
- присвоить этой сущности имя;
- выявить атрибуты сущности и присвоить им имя;
- определить уникальный идентификатор сущности.

Выявив сущности, необходимо определить, какие связи имеются между ними.

При определении связей (естественно, рассматриваем только те связи, которые имеют отношение к решаемым задачам обработки данных) необходимо учитывать следующее:

- то, как экземпляр одной сущности связан с экземпляром другой сущности;
- то, как должны быть установлены связи, чтобы была возможность ответа на все запросы пользователей (исходя из их информационных потребностей).

Далее необходимо присвоить связям имена и определить тип связей.

На втором этапе построенные локальные модели объединяются в **обобщенную концептуальную модель.**

Моделирование локальных представлений

Прежде всего, необходимо отметить, что построенная модель должна удовлетворять ряду требований:

- адекватно отражать представление пользователя о данных;
- давать возможность ответа на возможные запросы пользователя, причем делать это с минимальными затратами по количеству просматриваемых сущностей;
- представлять данные с минимальным дублированием.

Процесс построения модели, удовлетворяющей указанным требованиям, является творческим, и формализовать его, как правило, невозможно. Тем не менее можно указать некоторые способы порождения вариантов при моделировании. Выбор одного из таких вариантов на основе оценок объемов дублирования и числа просматриваемых объектов при ответах на запросы пользователей позволяет улучшить эксплуатационные характеристики проектируемой базы данных.

Вариативность моделирования обуславливается неоднозначностью выбора сущностей, атрибутов и связей. В одном варианте можно что-то взять за сущность, в другом варианте это же можно взять за атрибут (несколько атрибутов), в третьем варианте это можно определить как связь.

Так, например, ранее мы определили сущность ФАКУЛЬТЕТ с атрибутами «номер факультета», «название факультета». Введем сущность КАФЕДРА с атрибутами «номер кафедры», «название кафедры». Между этими сущностями есть связь «факультет состоит из кафедр». Возможен другой вариант, в котором вышеуказанная связь представляется через атрибуты сущности (у сущности ФАКУЛЬТЕТ введем дополнительные атрибуты, представляющие номера и названия всех кафедр этого факультета).

После того как выбран рациональный вариант локальной модели, производится редактирование введенных наименований сущностей, атрибутов и связей. Здесь выполняются следующие действия:

- устраняются расплывчатые наименования (все наименования должны однозначно пониматься каждым пользователем);
- устраняются синонимы (различные наименования одного и того же понятия);
- устраняются омонимы (одно и то же наименование разных понятий).

Эти действия, вообще говоря, носят итерационный характер, т.к. после их выполнения вновь могут возникать и расплывчатые наименования, и синонимы, и омонимы.

Объединение локальных моделей

На этом этапе ранее построенные модели локальных представлений отдельных пользователей (или групп пользователей) объединяются в единую концептуальную модель. Объединение локальных моделей производится следующими путями:

- слияние идентичных элементов;
- установление связей между наборами сущностей разных моделей;
- введение новых агрегированных элементов для представления связей между элементами разных моделей;
- обобщение различных подобных типов сущностей, позволяющее трактовать эти сущности как одну обобщенную сущность.

Рассмотрим каждый из этих путей.

Слияние идентичных элементов

Два или более элементов модели идентичны, если они имеют одинаковое смысловое значение. Объединение моделей с идентичными элементами осуществляется путем «слияния» этих элементов в один. Два набора сущностей СПЕЦИАЛЬНОСТЬ в модели 1 и 2 имеют одинаковое смысловое значение (рис.):



и могут быть заменены одним набором сущностей (рис.):

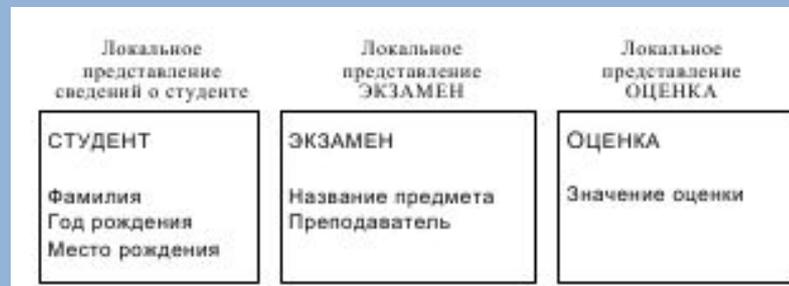


Установление связей между наборами сущностей разных моделей
При рассмотрении наборов сущностей объединяемых моделей необходимо выявление связей между ними, т.к. именно эти связи и определяют в конечном итоге интегрированную базу данных.

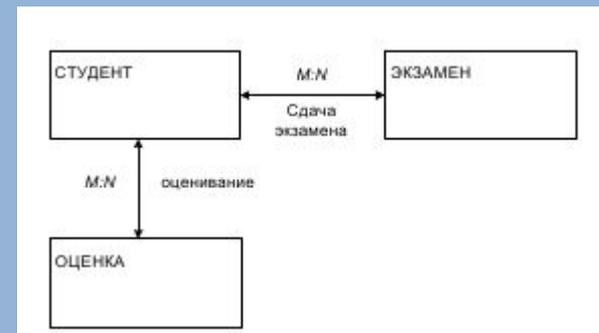
Введение агрегированных элементов

При объединении моделей связь между элементами разных моделей может рассматриваться как новый элемент.

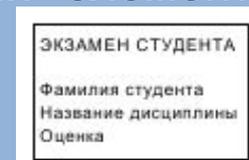
Рассмотрим в качестве примера моделирование информационного представления сдачи студентом экзаменов. Можно выделить ряд локальных представлений (рис.).



Объединяя локальные представления, устанавливаем новые связи (рис.).



Как уже отмечалось, одним из показателей «зрелости» модели является возможность ответа на запросы пользователей, и установление связей преследует именно эту цель. Нетрудно видеть, что какие бы связи в рассматриваемой модели ни вводились, невозможно ответить на запрос «какую оценку получил студент А по дисциплине В». В таком случае необходимо использовать принцип агрегации – необходимую связь между элементами модели ввести как некоторый новый элемент. В данном примере можно определить этот новый агрегированный элемент как ЭКЗАМЕН СТУДЕНТА (рис.).



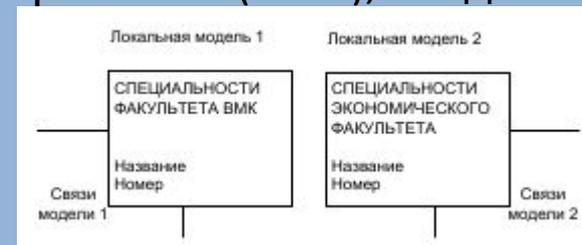
Далее процесс объединения локальных моделей продолжается обычным образом.

Обобщение подобных типов сущностей

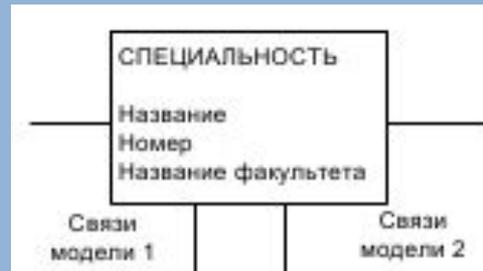
Рассмотрим локальные модели разных факультетов, например – модель факультета вычислительной математики и кибернетики (ВМК), модель

В локальную модель факультета ВМК входят сущности СПЕЦИАЛЬНОСТИ ФАКУЛЬТЕТА ВМК и

СТУДЕНТЫ ФАКУЛЬТЕТА ВМК, в локальную модель экономического факультета входят, соответственно, сущности СПЕЦИАЛЬНОСТИ ЭКОНОМИЧЕСКОГО ФАКУЛЬТЕТА и СТУДЕНТЫ ЭКОНОМИЧЕСКОГО ФАКУЛЬТЕТА (рис.).



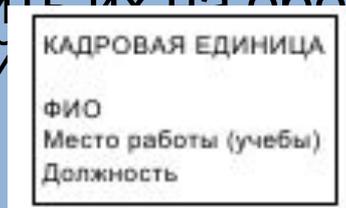
Два набора сущностей с СПЕЦИАЛЬНОСТИ ФАКУЛЬТЕТА ВМК и СПЕЦИАЛЬНОСТИ ЭКОНОМИЧЕСКОГО ФАКУЛЬТЕТА в моделях 1 и 2 имеют одинаковое смысловое значение и могут быть заменены одним набором сущностей с добавлением нового атрибута – название факультета (рис.).



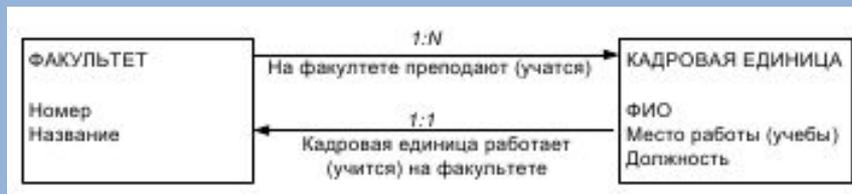
Отметим, что в данном случае подобным образом можно слить и все остальные сущности локальных моделей факультетов, так как сущности СТУДЕНТЫ ЭКОНОМИЧЕСКОГО ФАКУЛЬТЕТА и СТУДЕНТЫ ВМК также имеют одинаковое смысловое значение и их также можно объединить. Однако в общем случае каждая локальная модель может содержать сущности и связи, которых нет в других локальных моделях.

Рассмотрим другой пример. Предположим, что мы храним данные о студентах (фамилия, имя, отчество, курс, группа) и о преподавателях (фамилия, имя, отчество, кафедра, должность). Соответственно, в предметной области выделяем две сущности – СТУДЕНТ и ПРЕПОДАВАТЕЛЬ.

Эти разные сущности можно в некоторых случаях трактовать как подобные. Для обобщения соответствующих сущностей необходимо, прежде всего, обобщить их атрибуты. Заметим, что атрибуты «Фамилия, Имя, Отчество» у обеих сущностей совпадают, атрибуты «Кафедра» и «Курс», «Группа» показывают место работы (учебы) и их можно заменить обобщенным атрибутом «Место работы (учебы)». Атрибут «Должность» можно использовать и у сущности СТУДЕНТ, если в качестве значения соответствующего атрибута использовать значение «студент». Тогда две сущности ПРЕПОДАВАТЕЛЬ и СТУДЕНТ можно трактовать как подобные и заменить их на обобщенную сущность. Дадим этой обобщенной сущности название КАДРОВАЯ ЕДИНИЦА (рис.).



У студента атрибут «Место работы (учебы)» будет принимать значение соответствующее атрибутам «Курс. Группа», у преподавателя – название кафедры. Обобщенная модель представлена на рис.



В этом случае почти в два раза упрощается структура концептуальной модели, и соответственно, структура базы данных. Для работы с данными о преподавателях и студентах достаточно одного набора программ. Таким образом, обобщение подобных типов объектов может существенно сократить последующие затраты на программирование.

В процессе объединения локальных представлений, как и при локальном моделировании, производится редактирование наименований (т.к. здесь появляются новые наименования). Процесс объединения также носит итерационный характер и продолжается до тех пор, пока не будут интегрированы все представления, согласованы и устранены все противоречия, отредактированы все наименования. Полученное в результате объединения локальных представлений обобщенное представление и является концептуальной моделью.

Ограничения целостности

Под целостностью базы данных понимается то, что в ней содержится полная, непротиворечивая и адекватно отражающая предметную область (правильная) информация.

Огромный объем данных, вводимых в базу данных (причем разные данные могут вводиться разными пользователями), обуславливает большое число ошибок ввода (занесения).

Заметим, что при традиционной «бумажной» обработке информации также достаточно часто встречаются данные, записанные неверно. Но человек, работая с определенными данными, неявно использует для контроля имеющиеся у него представления об этих данных. Например, сотрудник отдела кадров, увидев в карточке работника год рождения 1693, сразу заметит эту ошибку и предположит, что просто переставлены две цифры и реальный год рождения 1963. То есть в представлениях сотрудника заключены некоторые логические ограничения на данные. Очевидно, что для контроля правильности вводимых данных при работе с базой данных целесообразно сформировать и использовать ограничения.

Соответствующие ограничения обычно разделяют на 3 группы: внешние, специально конструируемые и внутренние. К предметной области относятся первые две группы, которые мы кратко охарактеризуем в этом подразделе. Внутренние ограничения относятся уже к

модели данных и будут рассматриваться в разделе, посвященном модели данных.

Внешние ограничения.

Эти ограничения связаны с адекватностью отражения предметной области. Например, сотрудник организации не может быть моложе 17 и старше 90 лет. Соответствующее ограничение на год рождения (GR) можно записать следующим образом:

$$\text{Текущий год} - 17 > GR > \text{Текущий год} - 90.$$

Одним из способов задания таких ограничений является перечисление конечного множества допустимых значений какого-либо атрибута (так называемый «перечислимый» тип данных). Например, должность преподавателя в вузе может принимать одно из следующих значений: профессор, доцент, старший преподаватель, преподаватель, ассистент. Вводимое значение должности для конкретного экземпляра, не совпадающее с одним из перечисленных значений, является ошибкой.

Ограничения, описанные с помощью специальных конструкций.

Например, в базу данных вуза вводятся данные о числе студентов и преподавателей. По нормативным документам задано конкретное значение отношения числа студентов к числу преподавателей. Проверку этого отношения можно использовать для контроля достоверности данных. Такие конструкции строятся исходя из специфики данных рассматриваемой предметной области. Можно, например, построить много конструкций следующего вида:

сумма значений по заданному атрибуту по всем экземплярам сущностей должна совпадать со значением определенного атрибута в экземпляре другой сущности.

Таким образом, на стадии ER-моделирования для повышения достоверности данных необходимо сформулировать соответствующие ограничения на данные. В идеальном случае

каждое значение атрибута должно каким-то образом контролироваться. Использование этих ограничений позволяет существенно повысить достоверность данных в базе данных.

[Возврат к содержанию](#)