

1.1. Проектирование баз данных

1.1.1. Цели проектирования

Проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том, из каких отношений должна состоять БД и какие атрибуты должны быть у этих отношений. При проектировании базы данных решаются две основные проблемы:

1. Оптимальное отображение объектов предметной области в абстрактные объекты модели данных.
2. Обеспечение эффективного выполнения запросов к базе данных в среде конкретных СУБД.

Организация базы данных оказывает большое влияние на успех прикладной программы. Плохая организация данных затрудняет написание программ, работу с ними, а также усложняет совершенствование программ в соответствии с растущими требованиями к программному обеспечению.

Разработка правильной структуры отношений является главной при создании нового приложения. Плохая структура приводит к неэффективным программам или к невозможности реализации некоторых функций. Хорошо продуманный набор отношений дает возможность эффективно решать не только текущие задачи работы с БД, но и позволит в будущем решать многие проблемы, что сократит время написания приложений.

Структура данных влияет на успех программы больше, чем любое другое ее свойство. Управление данными становится простым, если данные организованы согласно всего нескольким правилам. Эти правила известны как правила нормализации.

Рассмотрим наиболее важные цели, стоящие перед проектированием.

Цель 1: Возможность хранения всех необходимых данных в БД.

БД должна быть образом моделируемой предметной области, где каждому объекту предметной области соответствуют данные в памяти ЭВМ, а каждому процессу предметной области – адекватные процедуры обработки данных. Первым шагом в процессе проектирования является

определение всех атрибутов, которые будут помещены в БД. Затем определяется количество необходимых отношений и состав атрибутов этих отношений.

Цель 2: Исключение избыточности данных.

Рассмотрим понятия дублирование данных и избыточное дублирование данных.

Отношение С-К (рис. 6.4 (а)) имеет два атрибута – СЛЖ# (табельный номер преподавателя) и КАФ (название кафедры, где работает преподаватель). Названия кафедр могут неоднократно появляться в отношении. В действительности название кафедры появляется один раз для каждого преподавателя, работающего на этой кафедре. Хотя «Информатики» и «Физики» появляются дважды в экземпляре С-К, приведенном на рис. 6.4 (а), ни одна из дублируемых кафедр не является избыточной. Причина отсутствия избыточности в том, что при удалении из отношения одного из дублируемых названий кафедры будет утеряна информация. Например, на рис. 6.4 (б) показан вид экземпляра отношения С-К при удалении дублированных названий кафедр. В этом случае нет возможности узнать, на каких кафедрах работают преподаватели с табельными номерами 187 и 230.

С-К		С-К	
СЛЖ#	КАФ	СЛЖ#	КАФ
102	Информатики	102	Информатики
127	Физики	127	Физики
187	Физики	187	-
230	Информатики	230	-
а		б	

Рис. 6.4. Дублирование данных, не являющееся избыточным

На рис. 6.5 (а) приведен пример отношения с избыточным дублированием данных. Отношение С-К-Т включает дополнительный атрибут ТЕЛ# - номер внутреннего телефона кафедры. Предполагается, что каждая кафедра имеет только один такой телефонный номер. В этом экземпляре отношения номера телефонов появляются более чем один раз, и дублированная информация является избыточной. Если, например, удалить один из номеров кафедры информатики, эта информация может

быть получена из других кортежей отношения. На рис. 6.5 (б) показано, как будет выглядеть отношение в случае замещения дублированных телефонных номеров «нулями» (или так называемыми null-значениями).

Конечно, телефонные номера кафедр Физики и Информатики могут быть найдены из других кортежей, т.е. данные не утеряны. Однако необходимы дополнительные меры, чтобы узнать реальные значения «нулей», например, номер телефона служащего с табельным номером 187. Более того, отношение, представленное на рис. 6.5 (б), имеет структуру, которая может нарушить целостность данных при удалении информации. Если служащий с СЛЖ# = 127 увольняется, кортеж <127, Физики, 34> должен быть удален из отношения. Следовательно, произойдет потеря телефонного номера кафедры Физики, поскольку нигде больше в отношении он не представлен. Таким образом, рассмотренный метод управления избыточностью неудовлетворителен.

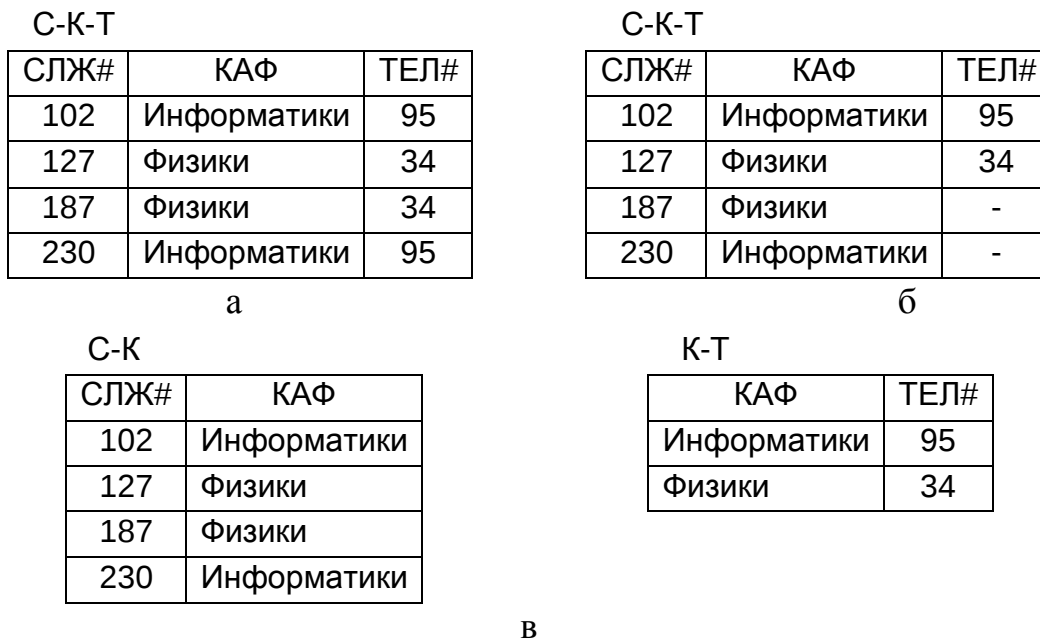


Рис. 6.5. Исключение избыточных данных

На рис. 6.5 (в) показан лучший способ исключения избыточности телефонных номеров. Отношение С-К-Т заменено двумя отношениями. Отношение С-К содержит информацию о табельных номерах служащих и месте их работы, а отношение К-Т – о телефонных номерах кафедр. Те-

перь при удалении информации о служащем 127 информация о номере телефона кафедры Физики утеряна не будет.

Цель 3: Сведение числа хранимых в БД отношений к минимуму.

Разбиение одного отношения на два или более отношений желательно для исключения ряда проблем. Но это неудобно для пользователя. Таким образом, нельзя допускать неограниченный рост числа отношений.

Процесс проектирования – это процесс нормализации всех отношений предметной области. *Нормализация* – это процесс разбиения одного отношения на два или более в соответствие со специальной процедурой разбиения.

1.1.2. Универсальное отношение

Рассмотрим несколько модифицированный пример, взятый из [7]. Пусть требуется разработать небольшую БД для деканата факультета. На первом шаге проектирования необходимо определить все атрибуты, которые необходимы деканату в БД, и связи между атрибутами:

- Сном: Номер зачетной книжки студента. Целое значение, уникальное для каждого студента университета.
- Сфам: Фамилия студента. Каждый студент имеет только одну фамилию, но не исключено, что одну фамилию носят несколько студентов.
- Кном: Номер комнаты в общежитии. Каждый студент живет в общежитии и имеет комнату. В одной комнате может проживать более одного студента.
- Тном: Номер телефона студента. Каждая комната общежития имеет один внутренний телефон и им пользуются все студенты, проживающие в этой комнате.
- Курс: Идентификационный номер курса, пройденного студентом. Например, МТА – это идентификационный номер курса математического анализа.
- Семестр: Идентификационный номер семестра, в котором данный курс был завершён студентом. Например, О99 оз-

начает осенний семестр 1999 г., а 398 – зимний семестр 1998 года. Возможно, что студент изучал один и тот же курс в различных семестрах.

Оценка: Оценка за курс. Оценка, полученная студентом за определенный курс в данном семестре.

На рис. 6.6 показан образец данных, представленных деканатом для их хранения в БД.

Указанная таблица отношением не является. Для иллюстрации этого рассмотрим одну «строку» из таблицы (рис. 6.6). На этом рисунке значения четырех полей Сном, Сфам, Кном и Тном – атомарные. *Атомарным* называется неделимое значение, а не множество, или кортеж, значений из некоторых доменов. Другими словами, в каждой таблице на пересечении столбца и строки всегда находится только одно значение и никак не группа из нескольких значений. В нашем случае (рис. 6.7) значения в полях Курс, Семестр и Оценка множественные. Эти поля – пример того, что называют **группой повторения**. Группа повторения – это поле или комбинация полей, которая содержит несколько значений данных в каждом кортеже (в общем случае разное количество значений в разных кортежах). *В реляционных базах данных группы повторения недопустимы.*

ФАКУЛЬТЕТ

Сном	Сфам	Кном	Тном	Курс	Семестр	Оценка
3297	Иванов	120	136	МТА	О97	4
				ИНФ	О97	5
				ФИЗ	398	2
				МТА	398	4
3496	Петров	238	344	МТА	О96	3
				МТА	397	4
				ПСИ	398	5
3596	Сидоров	120	136	АЛГ	397	2
				ГЕО	О97	4
				ФИЗ	О97	3
4798	Николаев	345	321	ПЕД	399	4

Рис. 6.6. Данные, необходимые деканату

Сном	Сфам	Кном	Тном	Курс	Семестр	Оценка
3297	Иванов	120	136	МТА	О97	4
				ИНФ	О97	5
				ФИЗ	398	2
				МТА	398	4

Рис. 6.7. Строка с группой повторения

Для того чтобы таблица, приведенная на рис. 6.6, могла считаться отношением реляционной базы данных, ее надо преобразовать так, чтобы она не содержала групп повторения, т.е. чтобы каждый элемент кортежа имел атомарное значение. Обычно для этого используется простая вставка (рис. 6.8).

ФАКУЛЬТЕТ

Сном	Сфам	Кном	Тном	Курс	Семестр	Оценка
3297	Иванов	120	136	МТА	О97	4
3297	Иванов	120	136	ИНФ	О97	5
3297	Иванов	120	136	ФИЗ	398	2
3297	Иванов	120	136	МТА	398	4
3496	Петров	238	344	МТА	О96	3
3496	Петров	238	344	МТА	397	4
3496	Петров	238	344	ПСИ	398	5
3596	Сидоров	120	136	АЛГ	397	2
3596	Сидоров	120	136	ГЕО	О97	4
3596	Сидоров	120	136	ФИЗ	О97	3
4798	Николаев	345	321	ПЕД	399	4

Рис. 6.8. Корректное отношение

В результате в отношении появляется большой объем избыточных данных. Но таблица на рис. 6.8 представляет собой экземпляр *корректного* отношения. Его называют *универсальным отношением* проектируемой БД. В универсальное отношение включаются все представляющие интерес атрибуты, и оно может содержать все данные, которые предполагается размещать в БД. Универсальное отношение может служить отправной точкой при проектировании БД.

1.1.3. Аномалии единственного отношения

Чаще всего единственное отношение в БД не является лучшим способом хранения данных. Это обусловлено тем, что при определенных операциях на данные будет оказываться определенное воздействие, что может привести к следующим трем проблемам:

1. Проблема, связанная с обновлением (изменением) данных в кортежах.
2. Проблема, вызываемая необходимостью удаления кортежей.
3. Проблема, обусловленная необходимостью добавления новых кортежей.

Эти проблемы называют аномалиями обновления, удаления и вставки.

Аномалия вставки

Для студента первого курса, еще не сдавшего ни один экзамен, необходимо включить в БД кортеж с null-значениями атрибутов Курс, Семестр и Оценка. Как уже отмечалось, null-значений следует избегать. Существует правило *целостности* объектов [6]: ни один элемент первичного ключа базового отношения не может быть null-значением.

В отношении ФАКУЛЬТЕТ первичным ключом является <Сном, Курс, Семестр>. Следовательно, включение в базу данных студентов первого курса в осеннем семестре невозможно вплоть до сдачи ими первого экзамена.

Аномалия обновления

В отношении ФАКУЛЬТЕТ большое число избыточных данных. Избыточность данных всегда свидетельствует о возможных ошибках модификации требуемых данных. С помощью операции обновления может быть модифицирована только часть требуемых данных.

В отношении ФАКУЛЬТЕТ присутствует как явная, так и неявная избыточность.

Явная избыточность обнаруживается в том, что фамилия студента, номер комнаты и номер телефона могут появляться в отношении не-

сколько раз. В экземпляре отношения ФАКУЛЬТЕТ (рис. 6.8) номер комнаты Иванова появляется четыре раза. Если деканат переселит Иванова в другую комнату, то будет необходимо проследить изменение номера комнаты во всех четырех кортежах во избежание противоречивости данных.

Неявная избыточность заключается в том, что один и тот же номер телефона имеют все студенты, живущие в одной комнате. На рис. 6.8 телефонный номер для комнаты 120 появляется в кортежах с фамилиями Иванов и Сидоров. Допустим, что на телефонной станции поменяли номер телефона комнаты 120 на 333. Если изменить телефонный номер только в тех кортежах, которые содержат номер телефона Иванова, забыв при этом о кортежах Сидорова, то правильный номер комнаты 120 будет фактически утерян, поскольку в отношении будут присутствовать два различных телефонных номера для одной комнаты.

На рис. 6.9 (а) приведен запрос на языке SQL для изменения номера телефона Иванова на 333. На рис. 6.9 (б) - SQL – запрос «Вывести перечень номеров телефонов для комнаты 120, исключить повторяющиеся значения» и на рис. 6.9 (в) - ответ на запрос.

```
UPDATE Факультет
    SET Тном = 333
WHERE Сфам='Иванов';
```

а

```
SELECT DISTINCT Факультет.Тном
FROM Факультет
WHERE Факультет.Кном=120;
```

б

Тном
136
333

в

Рис. 6.9. а – запрос на изменение номера телефона студента,

б – запрос на выборку номера телефона комнаты 120,

в – ошибочный результат

Аномалия удаления

В экземпляре отношения ФАКУЛЬТЕТ (рис. 6.8) есть только один кортеж, в котором Сном = 4798. Этот кортеж соответствует студенту Николаеву. Если окажется, что этот студент не сдавал экзамен по курсу

ПЕД, как это отмечено, то деканат должен удалить этот кортеж из отношения. Так как это единственный кортеж с информацией об этом студенте, его удаление приведет к исключению студента из БД. Если затем выполнить запрос на выборку всех студентов факультета, то фамилии Николаев в этом списке не будет.

1.2. Функциональные зависимости

1.2.1. Нормальные формы

Как уже говорилось в разделе 6.1, потенциально опасные с точки зрения аномалий отношения должны быть нормализованы, т.е. разбиты на два или более отношений в соответствии со специальной процедурой разбиения. Процесс нормализации основывается на концепции нормальных форм. На рис. 6.10 показано несколько нормальных форм, которые определены к настоящему времени.

Первые три нормальные формы (1НФ, 2НФ, 3НФ) были определены Коддом в 1972 г. Как показано на рис. 6.10, все нормализованные отношения находятся в 1НФ, некоторые отношения 1НФ находятся также в 2НФ и некоторые отношения 2НФ находятся также в 3НФ. Мотивом для введения дополнительных определений было то, что вторая нормальная форма в некотором смысле «более желательна», чем первая, а третья, в свою очередь, более желательна, чем вторая.

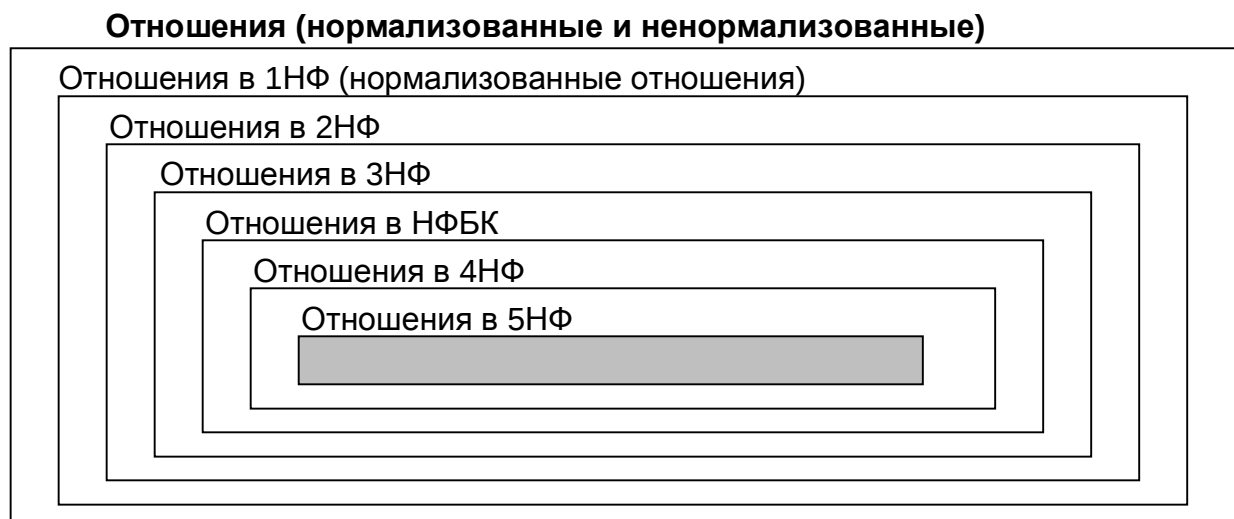


Рис. 6.10. Уровни нормализации

Нормальные формы (1НФ, 2НФ, 3НФ) были определены Коддом. В 1974 году Бойсом и Коддом дано переработанное и более точное понятие 3НФ. Новое определение 3НФ является более строгим в том смысле, что любое отношение в 3НФ по новому определению является таковым и по старому определению, но не всякое отношение в 3НФ по старому определению может являться таковым по новому определению. Для того чтобы эти определения можно было различать, отношение в 3НФ по новому определению обычно называют **нормальной формой Бойса-Кодда – НФБК**.

В 1978 году Фейгином была определена новая, четвертая нормальная форма (4НФ). Им же в 1979 году дано определение пятой нормальной формы (5НФ).

Процесс нормализации можно охарактеризовать как *последовательное приведение данного набора отношений к некоторой более желательной форме*. Эта процедура должна быть обратима, т.е. всегда можно использовать ее результат (например, множество отношений в 3НФ) для обратного преобразования (в исходное отношение в 2НФ). Возможность выполнения обратного преобразования означает, что в процессе нормализации *информация не утрачивается*.

1.2.2. Первая нормальная форма (1НФ)

Составной частью проектирования реляционной БД является процесс разбиения отношений с неудовлетворительными с точки зрения аномалий свойствами на новые отношения.

Для БД с небольшим числом атрибутов начальной точкой процедуры разбиения может служить универсальное отношение. Это отношение содержит все представляющие интерес атрибуты, и имеет структуру, в которой нет групп повторения. То есть все экземпляры отношений должны иметь форму, показанную на рис. 6.8.

Говорят, что отношение находится в первой нормальной форме (или 1НФ) тогда и только тогда, когда оно содержит только атомар-

ные значения. Отношение должно быть в 1НФ даже прежде постановки вопроса о его разбиении на два или более отношения.

1.2.3. Концепция функциональных зависимостей

Процесс разбиения отношения с целью уменьшения возможности возникновения аномалий называется *декомпозицией*. Причем декомпозиция должна быть *обратимой*, т.е. выполняться без потерь информации. Иначе говоря, интерес представляют только те операции, которые выполняются без потерь информации. Вопрос о том, происходит ли утрата информации при декомпозиции, тесно связан с концепцией **функциональной зависимости**.

Функциональная зависимость (ФЗ) определяется следующим образом. Пусть R – это отношение, а X и Y – произвольные подмножества множества атрибутов отношения R . Тогда Y функционально зависит от X тогда и только тогда, когда каждое значение множества X отношения R связано в точности с одним значением множества Y отношения R .

Иначе говоря, если два кортежа отношения R совпадают по значению X , они также совпадают и по значению Y . Значения X и Y могут изменяться, но только так, чтобы каждое уникальное значение X имело только одно значение Y , связанное с ним.

На рис. 6.11 показаны два разных способа записи ФЗ.

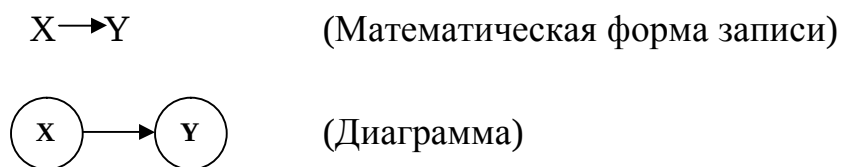


Рис. 6.11. Две формы записи ФЗ

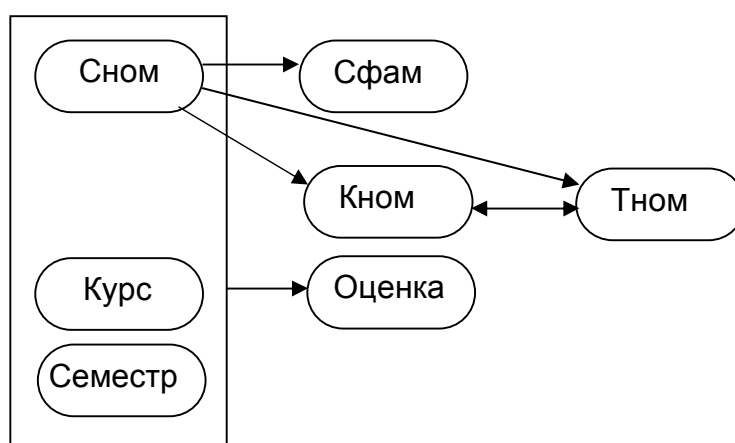
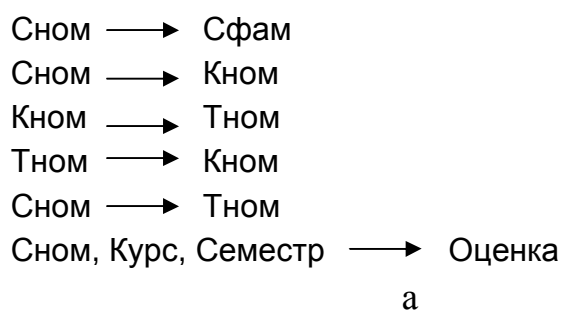
Читаются эти записи либо как « X функционально определяет Y », либо как « Y функционально зависит от X ».

В конкретной ситуации ФЗ определяется путем детализации свойств всех атрибутов в отношении и выводе заключения о том, как атрибуты соотносятся между собой.

В качестве примера вновь используем атрибуты отношения ФАКУЛЬТЕТ [7] (рис. 6.8) и описание того, как эти атрибуты связаны между собой (раздел 6.2.2.). На основе анализа описаний атрибутов могут быть выведены ФЗ, приведенные на рис. 6.12.

1. Номера зачетных книжек студентов являются уникальными. Если известен номер зачетной книжки Сном, с ним может быть связана только одна фамилия Сфам: Сном $\rightarrow$ Сфам. Обратное неверно. ФЗ Сфам $\rightarrow$ Сном неправильная, т.к. несколько студентов могут иметь одну фамилию.

2. Каждый студент прописан в одной комнате общежития, но в одной комнате может проживать несколько студентов. Таким образом, Сном $\rightarrow$ Кном верная ФЗ, а Кном $\rightarrow$ Сном – неверная.



б

Рис. 6.12. Функциональные зависимости между атрибутами
отношения ФАКУЛЬТЕТ

3. Поскольку в каждой комнате только один телефон, и каждый телефон имеет уникальный номер, получаем Кном $\rightarrow$ Тном и Тном $\rightarrow$

Кном. Говорят, что Сном и Тном *взаимозависимы*, и это обозначают $\text{Тном} \leftrightarrow \text{Кном}$.

4. Так как в каждой комнате один телефон, имеющий уникальный номер, то только один телефонный номер может быть связан со студентом, проживающим в этой комнате: $\text{Сном} \rightarrow \text{Тном}$. Обратное неверно, т.к. этим телефоном могут пользоваться несколько студентов, проживающих в данной комнате.

5. Зависимость Сном, Курс, Семестр $\rightarrow$ Оценка означает, что оценка однозначно определяется, если известен Сном студента, изучающего данный Курс в данном Семестре. Некоторые курсы могут изучаться в нескольких семестрах.

1.2.4. Нормальная форма Бойса-Кодда (НФБК)

В разделе 6.1 даны понятия потенциальных ключей и первичного ключа. Первичный ключ всегда является потенциальным ключом, однако не исключено наличие других потенциальных ключей.

Дадим определение еще одного термина. Если $A \rightarrow B$ есть ФЗ и B не зависит функционально от любого подмножества A, то говорят, что A представляет собой **детерминант** B.

Из рис. 6.12 можно заключить, что отношение ФАКУЛЬТЕТ имеет только один потенциальный ключ <Сном, Курс, Семестр>. Это минимальный набор атрибутов, значения которых однозначно определяют значения других атрибутов кортежа. Если ключ отношения содержит несколько атрибутов, то его называют **составным**.

Детерминантами в отношении ФАКУЛЬТЕТ являются левые части всех ФЗ в отношении: <Сном, Курс, Семестр>, <Сном>, <Кном> и <Тном>. Взаимные зависимости дают два детерминанта.

Коддом доказано утверждение о том, что большинство потенциальных аномалий в БД будет устранено в случае должной декомпозиции каждого отношения в *нормальную форму Бойса-Кодда* (НФБК).

Отношение находится в НФБК если и только если каждый детерминант отношения является потенциальным ключом.

Хотя существуют нормальные формы более высокого порядка (раздел 6.1), которые накладывают даже более сильные ограничения на разрабатываемые отношения, на практике стараются получить отношения в НФБК.

Отношение ФАКУЛЬТЕТ не находится в НФБК. Это видно из следующего списка детерминантов и потенциальных ключей отношения:

Потенциальные ключи

<Сном, Курс, Семестр>

Детерминанты

<Сном, Курс, Семестр>

<Сном>

<Кном>

<Тном>

Так как не каждый детерминант отношения ФАКУЛЬТЕТ является возможным ключом, отношение не находится в НФБК и его необходимо подвергнуть декомпозиции.

1.2.5. Алгоритм декомпозиции

Опишем метод проектирования реляционных БД с помощью алгоритма декомпозиции. В дальнейшем этот метод будет усовершенствован.

1. Разработка универсального отношения для БД.
2. Определение всех ФЗ между атрибутами отношения.
3. Если отношение находится в НФБК, то проектирование завершается; если нет, отношение должно быть разбито на два отношения.
4. Повторение шагов 2 и 3 для каждого нового отношения, полученного в результате декомпозиции. Проектирование завершается, когда все отношения будут находиться в НФБК.

Рассмотрим подробнее шаг 3 и определим, как осуществлять декомпозицию отношения, не приведенного к НФБК, на два отношения:

- 3.1. Если отношение $R(A,B,C,D,E,...)$ не приведено к НФБК, определяется ФЗ, например $C \rightarrow D$, для которой C является детерминантом, но не является потенциальным ключом.
- 3.2. Создаются два новых отношения $R1(A,B,C,E,...)$ и $R2(C,D)$, где аномальная ФЗ была выделена из R и опущена при формировании отношения $R1$. Выделенная ФЗ используется полностью для формирования отношения $R2$.

Теперь необходимо вернуться к шагу 2 и проверить, находятся ли в НФБК отношения $R1$ и $R2$. Про отношение $R2$ говорят, что оно является *проекцией* отношения R . Этот тип декомпозиции называется *декомпозицией без потерь* при естественном соединении. То есть процесс обратим, и если соединить отношения $R1$ и $R2$ в одно отношение – потеря информации не произойдет.

Остановимся подробнее на понятии декомпозиции без потерь. Пусть выполняется разложение одного отношения, например $R(A,B,C)$, на два новых отношения $R1(A,B)$ и $R2(B,C)$. Отношения, полученные с помощью декомпозиции, всегда будут приводить к получению согласующихся данных в том случае, когда декомпозиция была проведена способом, при котором естественное соединение отношений $R1(A,B)$ и $R2(B,C)$ дает в итоге в точности исходное отношение $R(A,B,C)$. Декомпозиция, характеризующаяся этим качеством, и есть декомпозиция без потерь. Если естественное соединение $R1$ и $R2$ приводит к появлению большего, чем в отношении R , числа кортежей, то декомпозиция называется декомпозицией с потерями. Отсутствие потерь при декомпозиции отношения $R(A,B,C)$ на $R1(A,B)$ и $R2(B,C)$ гарантируется при условии, если от общего атрибута двух получаемых отношений – в данном случае атрибута B – зависит хотя бы один атрибут из двух оставшихся. В нашем случае, если $B \rightarrow A$ или $B \rightarrow C$, то декомпозиция осуществляется без потерь.

На рис. 6.13 (а) показан пример декомпозиции с потерями. В этом примере ни $B \rightarrow A$, ни $B \rightarrow C$ не являются корректными ФЗ и естественное соединение порождает два новых кортежа.

На рис. 6.13 (б) приведен пример декомпозиции без потерь. В этом примере предполагается, что $B \rightarrow C$ является корректной ФЗ и значения

экземпляра $R(A,B,C)$ отражают этот факт. Естественное соединение $R1$ и $R2$ в этом случае дает в результате в точности исходное отношение.

Выше предполагалось, что атрибуты A , B и C являются простыми. В общем случае A , B и C могут быть как простыми, как и составными атрибутами.

Базовый алгоритм декомпозиции, рассмотренный выше, является алгоритмом декомпозиции без потерь.

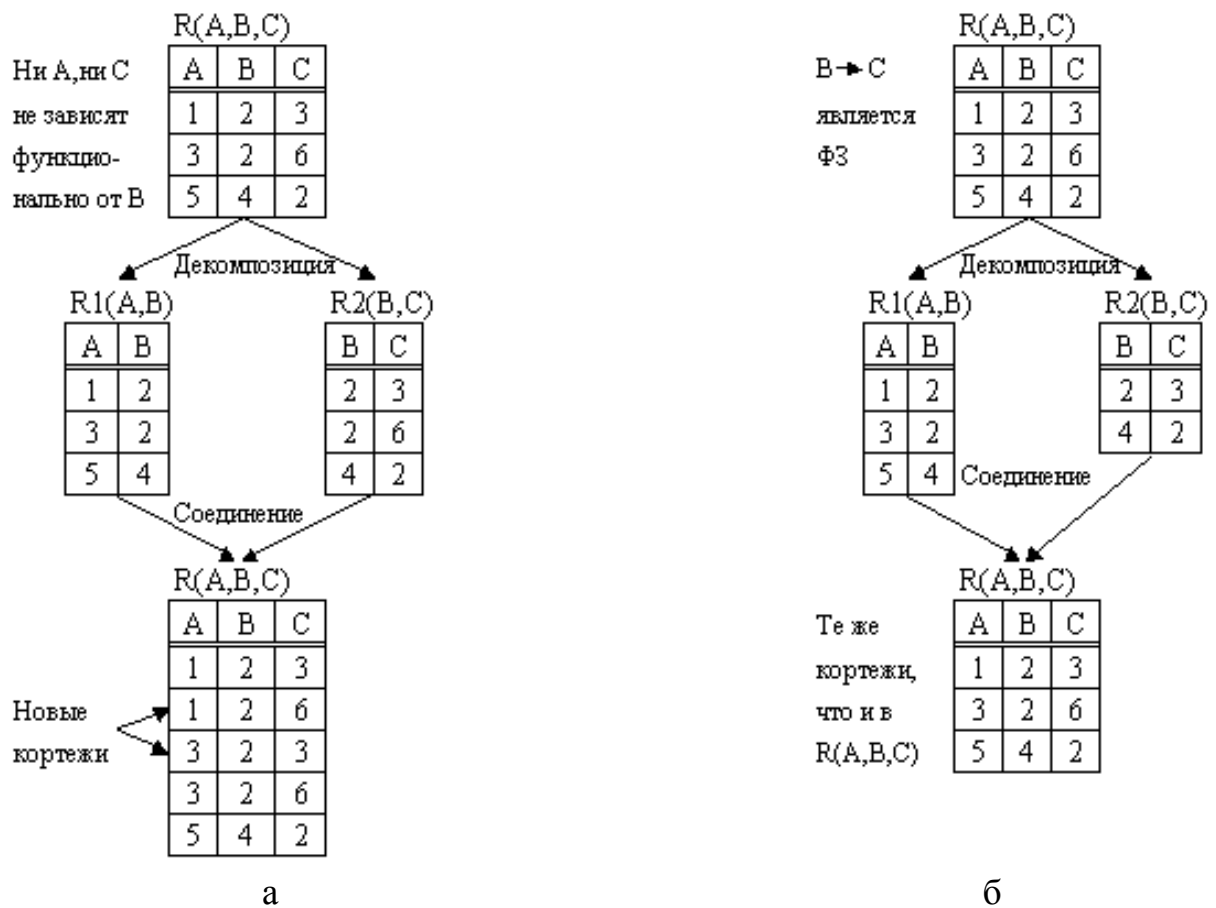


Рис. 6.13. Пример декомпозиции
а – с потерями, б – без потерь

Используем приведенный алгоритм для декомпозиции отношения ФАКУЛЬТЕТ. Рассмотрев детерминанты и потенциальные ключи отношения, видим, что имеются три детерминанта, не являющихся возможными ключами:

$\langle C_{ном} \rangle$, $\langle T_{ном} \rangle$ и $\langle K_{ном} \rangle$.

Определим следующее универсальное отношение, находящееся в 1НФ:
 ФАКУЛЬТЕТ (Сном, Курс, Семестр, Сфам, Кном, Тном, Оценка)

Экземпляр этого отношения показан на рис. 6.8. Кандидатами среди ФЗ для осуществления проекции являются:

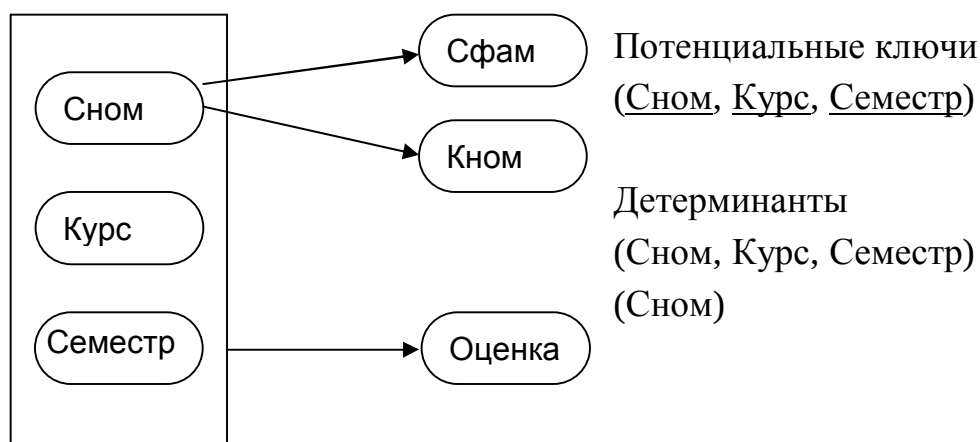
Сном $\rightarrow$ Кном; Сном $\rightarrow$ Тном; Кном $\rightarrow$ Тном и Тном $\rightarrow$ Кном.

Простым правилом для выбора ФЗ для проекции может служить поиск цепочки ФЗ вида:

$$A \rightarrow B \rightarrow C$$

с последующим использованием для проекции крайней правой зависимости. В нашем случае такой цепочкой является Сном $\rightarrow$ Кном $\rightarrow$ Тном. Конец цепочки Кном $\rightarrow$ Тном порождает первую проекцию. Полученные в результате отношения R1 и R2 приведены на рис. 6.14 вместе со связанными с ними ФЗ.

R1 (Сном, Курс, Семестр, Сфам, Кном, Оценка)



R2 (Тном, Кном)



Рис. 6.14. Отношения R1 и R2, полученные проекцией Кном $\leftrightarrow$ Тном

из отношения ФАКУЛЬТЕТ

Отношение R2 (Кном, Тном) находится в НФБК, так как все его детерминанты являются потенциальными ключами. Поэтому отношение R2 не нуждается в дальнейшей декомпозиции.

Отношение R1 (Сном, Курс, Семестр, Сфам, Кном, Оценка) не только не находится в НФБК, но и не соответствует второй нормальной форме (2НФ), которая накладывает менее сильные ограничения на отношения. Для определения 2НФ вводится понятие полной функциональной зависимости.

Неключевой атрибут **функционально полно зависит** от составного ключа, если он функционально зависит от всего ключа в целом, но не находится в функциональной зависимости от какого-либо из входящих в него атрибутов.

Отношение находится в 2НФ, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от ключа.

Из рис. 6.14 видно, что неключевые атрибуты Сфам и Кном не зависят функционально полно от потенциального составного ключа (Сном, Курс, Семестр). Поэтому в системе из двух отношений R1 и R2 возможны все исходные аномалии за исключением аномалии обновления, связанной с изменением номера телефона для определенной комнаты (раздел 6.2.3).

Итак, отношение R1 (Сном, Курс, Семестр, Сфам, Кном, Оценка) не находится в 2НФ, а значит и в НФБК. Детерминант <Сном> не является потенциальным ключом, что приводит к неполной функциональной зависимости атрибутов Сфам и Кном от потенциального ключа. Следовательно, отношение R1 необходимо подвергнуть дальнейшей декомпозиции. Детерминант <Сном>, из-за которого возникли трудности, имеет два зависимых от него атрибута

Сном -> Сфам,

Сном -> Кном,

что можно рассматривать в качестве единичной ФЗ с составной правой частью

Сном -> Сфам, Кном.

Проекция отношения R1, порождаемая этой ФЗ, приводит к получению отношений R3 и R4, показанных на рис. 6.15. Эти два отношения находятся в НФБК и вместе с отношением R2 могут использоваться для формирования БД для факультета.

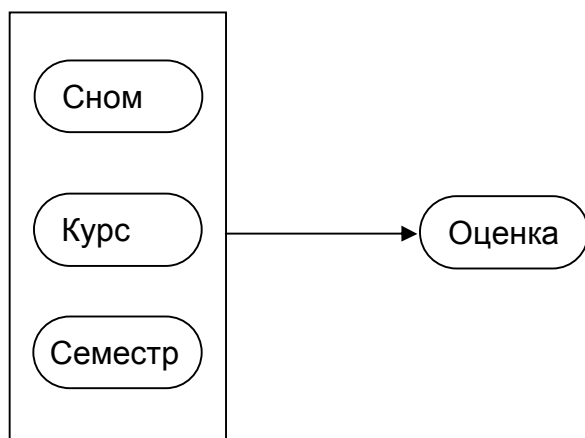
На рис. 6.16 показан окончательный вид отношений для этой БД, которую назовем ДЕКАНАТ, а также экземпляры каждого отношения.

В процессе декомпозиции исходное отношение ФАКУЛЬТЕТ было разбито на три логических компонента: R2, в котором содержится информация о комнатах и телефонах; R3, в котором содержится информация об учебных курсах и полученных студентами оценках; R4, в котором содержится информация о студентах. Такое логическое разбиение является результатом использования в процессе декомпозиции информации, детализирующей то, как различные атрибуты в исходном отношении соотносятся друг с другом, и заложенной в функциональных зависимостях.

R3 (Сном, Курс, Семестр, Оценка)

Потенциальные ключи

1. (Сном, Курс, Семестр)



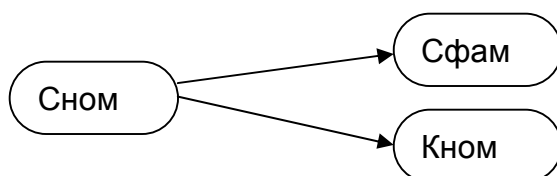
Детерминанты

1. (Сном, Курс, Семестр)

R4 (Сном, Сфам, Кном)

Потенциальные ключи

1. (Сном)



Детерминанты

1. (Сном)

Рис. 6.15. Отношения R3 и R4, полученные проекцией Сном -> Сфам, Кном из отношения R1

1.2.6. Обзор исходных аномалий

Рассмотрим, опираясь на приведенную на рис. 6.16 БД ДЕКАНАТ, удалось ли удалить аномалии вставки, удаления и обновления, характерные для единственного отношения в 1НФ (раздел 6.2.3).

Аномалия вставки

Когда основой БД ФАКУЛЬТЕТ служило единственное отношение в 1НФ, новый студент не мог быть в нее включен до сдачи первого экзамена. В БД ДЕКАНАТ информация общего характера о студентах хранится в отношении R4. Как только новый студент принимается в университет и ему отводится комната, этот студент может быть включен в БД, а именно в отношение R4. Таким образом, исходная аномалия вставки исключена с помощью декомпозиции.

R2 (Кном, Тном)

R3 (Сном, Курс, Семестр, Оценка)

R4 (Сном, Сфам, Кном)

а

R3

Сном	Курс	Семестр	Оценка
3297	МТА	О97	4
3297	ИНФ	О97	5
3297	ФИЗ	398	2
3297	МТА	398	4
3496	МТА	О96	3
3496	МТА	397	4
3496	ПСИ	398	5
3596	АЛГ	397	2
3596	ГЕО	О97	4

R2

Кном	Тном
120	136
238	344
345	321

R4

Сном	Сфам	Кном
3297	Иванов	120
3496	Петров	238
3596	Сидоров	120

3596	ФИЗ	О97	3
4798	ПЕД	399	4

4798	Николаев	345
------	----------	-----

б

Рис. 6.16 а – база данных ДЕКАНАТ; б – экземпляр базы данных, в котором используются данные, приведенные на рис. 6.8.

Аномалия обновления

При использовании отношения ФАКУЛЬТЕТ проблема возникала, когда на телефонной станции для комнаты 120 менялся номер телефона на 333, и информация об изменении номера телефона вносилась только для одного студента, проживающего в этой комнате. Результатом изменения явилось появление в БД двух различных телефонных номеров для телефона, установленного в комнате 120.

В БД ДЕКАНАТ телефонные номера располагаются в отношении R2. Каждая комната может иметь только один телефонный номер. Атрибут Кном является первичным ключом отношения R2, следовательно значения его должны быть уникальными. Теперь для изменения номера телефона для комнаты 120 достаточно в кортеже отношения R2, в котором Кном=120, изменить номер Тном на 333. То есть действие заключается в изменении телефонного номера для комнаты, так что для всех живущих в этой комнате студентов телефонный номер будет изменен автоматически. Таким образом, исходная аномалия обновления устраняется с помощью НФБК-проектирования.

Аномалия удаления

Когда единственное отношение ФАКУЛЬТЕТ использовалось в качестве БД, удаление кортежа, включавшего значения Сном=4798 и Курс=ПЕД, привело к исчезновению из БД информации о студенте с номером 4798. Этого не может произойти в случае БД ДЕКАНАТ, так как информация об оценках и информация общего характера о студентах разнесена в два различных отношения R3 и R4. При исключении факта, что студент с номером 4798 сдал экзамен по курсу ПЕД в семестре 399, из отношения R3 будет удален кортеж <4798, ПЕД, 399, 4>. Это действие

никак не скажется на информации о студенте Николаеве с номером зачетной книжки 4798, хранимой в R4.

Таким образом, все три аномалии, характерные для БД, состоящей из одного отношения в 1НФ, устраняются при НФБК-проектировании. Однако в результате устранения аномалий появились три отношения, требующих хранения, вместо одного. Это означает, что запросы для получения информации из БД, возможно, станут более сложными. На рис. 6.17 приведен пример типичного запроса на получение номера телефона студента с номером зачетной книжки 3596 как для случая отношения ФАКУЛЬТЕТ (а), так и для случая БД ДЕКАНАТ (б).

```
SELECT Факультет.Тном
FROM Факультет
WHERE Факультет.Сном=3596;
```

а

```
SELECT R2.Тном
FROM R2 INNER JOIN R4 ON R2.Кном = R4.Кном
WHERE R4.Сном=3596;
```

б

Рис. 6.17. Запросы на получение номера телефона студента с номером 3596:

а – для случая отношения ФАКУЛЬТЕТ; б – для случая БД ДЕКАНАТ

1.2.7. Другая декомпозиция отношения ФАКУЛЬТЕТ

В разделе 6.3.5 декомпозиция отношения ФАКУЛЬТЕТ на три отношения была начата проекцией, порождаемой ФЗ Кном $\rightarrow$ Тном. Эта функциональная зависимость была выбрана потому, что являлась последней в цепочке Сном $\rightarrow$ Кном $\rightarrow$ Тном на схеме, представленной на рис. 6.12. Из этой же схемы видно, что присутствует другая цепочка зависимостей с таким же их числом:

Сном $\rightarrow$ Тном $\rightarrow$ Кном.

Здесь крайней правой ФЗ является Тном $\rightarrow$ Кном. Если эта ФЗ выделяется первой из отношения ФАКУЛЬТЕТ, то в итоге будет получена следующая БД в НФБК:

R2 (Тном, Кном)

R3 (Сном, Курс, Семестр, Оценка)

R4 (Сном, Сфам, Тном)

Эта БД так же корректна, как и БД, приведенная на рис. 6.16. Единственное отличие в том, что Тном в этой БД используется в качестве первичного ключа для отношения R2 и атрибута, связывающего R4 и R2. Два различных решения одной проблемы являются результатом взаимной зависимости между атрибутами Тном и Кном.

1.3. Некоторые уточнения алгоритма декомпозиции

Алгоритм проектирования БД, описанный в разделе 6.3, не свободен от некоторых внутренних проблем. Одна проблема заключается в том, что процесс декомпозиции может усложниться в результате присутствия так называемых избыточных ФЗ.

Зависимость, не заключающая в себе такой информации, которая не могла бы быть получена на основе других зависимостей, называется **избыточной ФЗ**. Избыточная ФЗ не содержит уникальной информации и может быть удалена из набора ФЗ уже на начальном этапе проектирования до применения алгоритма декомпозиции.

1.3.1. Транзитивные зависимости

Транзитивные зависимости являются избыточными и определяются следующим образом:

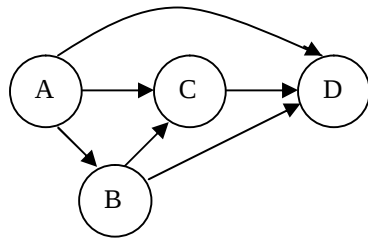
Если $A \rightarrow B$ и $B \rightarrow C$, то $A \rightarrow C$ – **транзитивная зависимость**.

На рис. 6.18 приведен пример того, как может быть упрощен набор ФЗ при помощи исключения транзитивных зависимостей [7]. На рис. 6.18 (а) представлен исходный набор функциональных зависимостей до начала

проектирования. На рис. 6.18 (г) показан набор неизбыточных ФЗ, выделенных путем удаления всех транзитивных зависимостей из исходного набора.

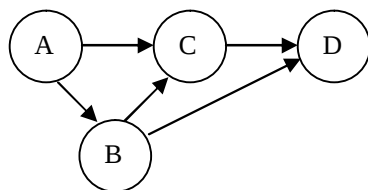
Отношение находится в 3НФ, если оно находится в 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Набор неизбыточных ФЗ, полученный путем удаления всех избыточных ФЗ из исходного набора, называется *минимальным покрытием*. Минимальное покрытие не всегда является уникальным и зависит от порядка, в котором удаляются избыточные ФЗ. Например, если вернуться к рис. 6.12, то из него видно, что при проектировании БД ДЕКАНАТ существуют два минимальных покрытия. Одно получается при удалении из исходного набора зависимости Сном $\rightarrow$ Тном, а второе – при удалении Сном $\rightarrow$ Кном. Использование этих двух минимальных покрытий приведет к построению тех же самых двух БД, проектирование которых обсуждалось выше.



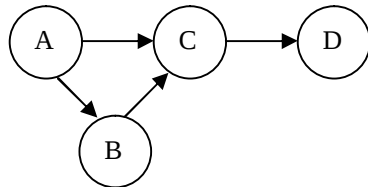
(а) Исходный набор ФЗ

$A \rightarrow B$
 $A \rightarrow C$
 $A \rightarrow D$
 $B \rightarrow C$
 $B \rightarrow D$
 $C \rightarrow D$



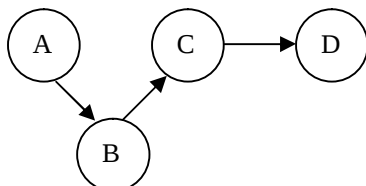
(б) $A \rightarrow D$ удалена, т.к. $A \rightarrow C$ и $C \rightarrow D$

$A \rightarrow B$
 $A \rightarrow C$
 $B \rightarrow C$
 $B \rightarrow D$
 $C \rightarrow D$



(в) $B \rightarrow D$ удалена, т.к. $B \rightarrow C$ и $C \rightarrow D$

$A \rightarrow B$
 $A \rightarrow C$
 $B \rightarrow C$
 $C \rightarrow D$



$A \rightarrow B$
 $B \rightarrow C$

C -> D

(г) A -> C удалена, т.к. A -> B и B -> C и в итоге получен неизбыточный набор ФЗ

Рис. 6.18. Удаление транзитивных зависимостей

Избыточные ФЗ необходимо удалять по одной, каждый раз заново анализируя новый набор на предмет присутствия в нем избыточных ФЗ. Эта процедура завершается, как только не останется ни одной избыточной ФЗ. Оставшийся набор является минимальным покрытием.

1.3.2. Уточненный алгоритм декомпозиции

Обобщенный алгоритм проектирования состоит из следующих шагов.

1. Построение универсального отношения для БД.
2. Определение всех ФЗ, существующих между атрибутами универсального отношения.
3. Удаление всех избыточных ФЗ из исходного набора ФЗ с целью получения минимального покрытия.
4. Использование ФЗ из минимального покрытия для декомпозиции универсального отношения в набор НФБК-отношений. Здесь должен быть применен алгоритм декомпозиции, описанный в разд. 6.2.4.5.
5. Если может быть получено более чем одно минимальное покрытие, проводится сравнение результатов, полученных на основе разных минимальных покрытий, с целью определения варианта, лучше других отвечающего требованиям предприятия.

1.3.3. Проверка отношений на завершающей фазе их проектирования

После получения набора НФБК-отношений этот набор надо контролировать на предмет невыявленных проблем.

1. Составляются списки ФЗ для каждого отношения. Одна и та же ФЗ не должна появляться более чем в одном отношении. Кроме того, набор ФЗ, полученный в результате проектирования, должен в точности совпадать с набором из минимального покрытия, полученного перед началом декомпозиции.
2. Осуществляется проверка на присутствие избыточных отношений, которые должны быть исключены из проектного набора.

Отношение называется *избыточным*, если:

а) все атрибуты в избыточном отношении могут быть найдены в другом отношении проектного набора;

б) все атрибуты в избыточном отношении могут быть найдены в отношении, которое может быть получено из других отношений проектного набора с помощью одной или нескольких операций объединения.

Первый тип избыточности иллюстрируется следующим примером. Пусть набор проектных отношений имеет вид:

$$R1(\underline{A}, B)$$

$$R2(\underline{B}, C, Y, Z)$$

$$R3(\underline{A}, B, K)$$

Отношение R1 является избыточным, т.к. все его атрибуты присутствуют в отношении R3.

Избыточность второго типа иллюстрируется следующим проектным набором:

$$R1(\underline{A}, C, X)$$

$$R3(\underline{D}, \underline{K}, F)$$

$$R5(\underline{D}, E, G, H)$$

$$R7(\underline{A}, \underline{B}, D)$$

$$R8(\underline{A}, \underline{B}, \underline{E}, G)$$

Отношение R8 является избыточным, т.к. объединение R5 и R7 по общему атрибуту D дает отношение R9(A,B,D,E,G,H), которое содержит все атрибуты отношения R8.

1.4. Нормальные формы более высокого порядка

Методы проектирования, основанные на понятии ФЗ, рассматриваемые в гл. 6.2.4 и 6.2.5, подробно описаны в литературе и хорошо зарекомендовали себя в качестве средств проектирования различных приложений. Однако могут возникать ситуации, в которых данные связаны таким образом, что для получения наилучшего результата необходимо использовать специальные методы проектирования. Примером служат нормальные формы, накладывающие более сильные ограничения, чем НФБК, и развитые для устранения нежелательных характеристик, которые могут присутствовать даже в отношениях, находящихся в НФБК.

1.4.1. Многозначные зависимости и четвертая нормальная форма (4НФ)

Четвертая нормальная форма касается отношений, в которых имеются повторяющиеся наборы данных. Декомпозиция, основанная на функциональных зависимостях, не приводит к исключению такой избыточности. В этом случае используют декомпозицию, основанную на *многозначных зависимостях*.

Многозначная зависимость является обобщением функциональной зависимости и рассматривает соответствия между *множествами* значений атрибутов.

В качестве примера рассмотрим отношение НАГРУЗКА (ИМЯ, КУРС, УЧЕБНИК), хранящее сведения о курсах, читаемых преподавателем, и написанных им учебниках. Пусть профессор Иванов читает курсы "Теория упругости" и "Теория колебаний" и имеет соответствующие учебные пособия, а профессор Петров читает курс "Теория удара" и является автором учебников "Теория удара" и "Теоретическая механика". Тогда наше отношение будет иметь вид:

НАГРУЗКА

ИМЯ	КУРС	УЧЕБНИК
Иванов	Теория упругости	Теория упругости

Иванов	Теория колебаний	Теория упругости
Иванов	Теория упругости	Теория колебаний
Иванов	Теория колебаний	Теория колебаний
Петров	Теория удара	Теория удара
Петров	Теория удара	Теоретическая механика

Это отношение имеет значительную избыточность и его использование приводит к возникновению *аномалии обновления*. Например, добавление информации о том, что профессор Петров будет также читать лекции по курсу "Теория упругости" приводит к необходимости добавить два кортежа (по одному для каждого написанного им учебника) вместо одного:

Петров	Теория упругости	Теория удара
Петров	Теория упругости	Теоретическая механика

Тем не менее, отношение НАГРУЗКА находится в НФБК, поскольку является полностью ключевым.

Заметим, что указанные аномалии исчезают при замене отношения НАГРУЗКА его проекциями:

ДИСЦИПЛИНА		УЧЕБНИК	
ИМЯ	КУРС	ИМЯ	УЧЕБНИК
Иванов	Теория упругости	Иванов	Теория упругости
Иванов	Теория колебаний	Иванов	Теория колебаний
Петров	Теория удара	Петров	Теоретическая механика
Петров	Теория упругости	Петров	Теория удара

Аномалия обновления возникает в данном случае потому, что в отношении НАГРУЗКА имеются:

- 1) зависимость *множества* значений атрибута КУРС от *множества* значений атрибута ИМЯ;
- 2) зависимость *множества* значений атрибута УЧЕБНИК от *множества* значений атрибута ИМЯ.

Такие зависимости и называются *многозначными* и обозначаются как:

ИМЯ ->> КУРС

ИМЯ ->> УЧЕБНИК

Нетрудно показать, что многозначные зависимости всегда образуют связанные пары, поэтому их часто обозначают:

ИМЯ ->> КУРС | УЧЕБНИК

Очевидно, что каждая функциональная зависимость является многозначной, но не каждая многозначная зависимость является функциональной.

Определение четвертой нормальной формы:

Отношение находится в 4НФ, если оно находится в НФБК и в нем отсутствуют многозначные зависимости, не являющиеся функциональными зависимостями.

1.4.2. Зависимости по соединению и пятая нормальная форма (5НФ)

До сих пор мы предполагали, что единственной операцией, необходимой для устранения избыточности в отношении, была декомпозиция его на две проекции. Однако существуют отношения, для которых нельзя выполнить декомпозицию без потерь на две проекции, но которые можно подвергнуть декомпозиции без потерь на три (или более) проекции. Этот факт получил название **зависимости по соединению**, а такие отношения называют **3-декомпозируемые отношения** (ясно, что любое отношение можно назвать "n-декомпозируемым", где $n \geq 2$).

Детально этот вопрос обсуждается в [7], заметим лишь, что зависимость по соединению является обобщением многозначной зависимости. Отношения, в которых имеются зависимости по соединению, не являющиеся одновременно ни многозначными, ни функциональными, также характеризуются аномалиями обновления. Поэтому вводится понятие пятой нормальной формы.

Определение пятой нормальной формы.

Отношение находится в 5НФ тогда и только тогда, когда любая зависимость по соединению в нем определяется только его возможными ключами.

Другими словами, каждая проекция такого отношения содержит не менее одного возможного ключа и не менее одного неключевого атрибута.

2. Другой подход к проектированию

Декомпозиционный подход к проектированию является вполне пригодным при условии небольшого числа задействованных атрибутов (меньше 20). В случаях, когда число атрибутов делает сложным применение метода декомпозиции, используются другие методы проектирования. Один из таких методов называется «сущность-связь» (entity – relationship model, ER – model). Он предложен в 1976 году Ченом [8, 9] и отличается от метода декомпозиции тем, что ФЗ привлекаются не на начальном, а на конечном этапе проектирования. В некоторых источниках [6] этот метод называется «объект-отношение».

2.1. Сущности и связи

Любой фрагмент предметной области может быть представлен как множество сущностей, между которыми существует некоторое множество связей. Пусть проектируется БД для хранения информации о преподавателях факультета и о тех курсах, которые они читают. Двумя главными объектами, или сущностями, представляющими в данном случае интерес, являются ПРЕПОДАВАТЕЛЬ и КУРС. Эти две сущности соотносятся с помощью связи ЧИТАЕТ, то есть можно сказать «ПРЕПОДАВАТЕЛЬ ЧИТАЕТ КУРС».

Связь ЧИТАЕТ, существующая между двумя сущностями ПРЕПОДАВАТЕЛЬ и КУРС, может быть графически представлена несколькими способами. Рассмотрим следующие два способа. Рис. 7.1 иллюстрирует использование *диаграммы ER-экземпляров* с помощью примера, показывающего, какой в точности курс читает каждый преподаватель. В этом примере каждый преподаватель идентифицируется номером преподавателя (нп), и каждый курс – номером курса (нк).

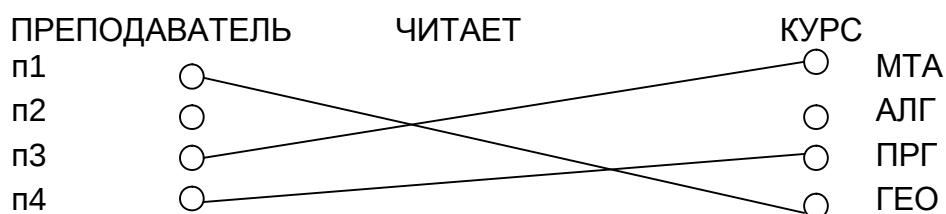


Рис. 7.1. Пример диаграммы ER-экземпляров

Рис. 7.2 называется *диаграммой ER-типа* и содержит ту же общую информацию, которая содержится на рис. 7.1.



Рис. 7.2. Пример диаграммы ER-типа

Поясним некоторые термины.

Сущность (entity) – это объект, который может быть идентифицирован неким способом, отличающим его от других объектов. Этот объект должен иметь экземпляры, отличающиеся друг от друга и допускающие однозначную идентификацию. Примерами сущностей могут служить машины, банковские счета, университеты, служащие, контракты. На рис. 7.1 и 7.2 сущностями являются КУРС и ПРЕПОДАВАТЕЛЬ, в то время как отдельные экземпляры каждой сущности идентифицируются с помощью номера курса и номера преподавателя соответственно.

Связь (relationship) – это ассоциация, установленная между несколькими сущностями. Типичными примерами связей между двумя сущностями являются: служащие РАБОТАЮТ в отделах, студенты ИЗУЧАЮТ учебные предметы, рабочие ОБСЛУЖИВАЮТ механизмы.

В методике проектирования данных обычно сущности обозначаются с помощью имен существительных, а связи – глагольными формами.

Тесно связано с предыдущими третье важное понятие, уже обсуждавшееся выше, а именно атрибут.

Атрибут есть свойство сущности. Например, атрибутами, которые могут быть свойствами сущности ПРЕПОДАВАТЕЛЬ, являются: номер преподавателя, фамилия, имя, отчество, ученая степень, ученое звание, дата рождения. Сущность фактически представляет собой множество атрибутов, которые описывают свойства всех экземпляров данной сущности.

В некоторых случаях может понадобиться набор атрибутов для идентификации каждого экземпляра сущности. Атрибут, или набор атрибутов, используемый для однозначной идентификации экземпляра сущности, называется **ключом сущности**.

Каждый экземпляр связи однозначно определяется набором ключей сущностей, соединяемых этой связью. Таким образом, <пЗ,МТА> является одним *ключом связи*.

На первом этапе процесса проектирования единственными требуемыми атрибутами являются те, которые необходимы для формирования ключей сущностей. Другие атрибуты вместе с определенными над ними ФЗ будут добавлены позднее в процессе проектирования.

На диаграммах ER-типа, подобных показанной на рис. 7.2, сущности представляются в виде прямоугольников, а связи – в виде ромбов. Ниже каждой сущности указывается ключ сущности.

В большинстве случаев для определения набора отношений проектируемой БД используют диаграммы ER-типа, а не диаграммы экземпляров.

2.2. Степень связи

В общем случае связь может объединять n сущностей (причем n не меньше 2). В случае $n=2$, т.е. когда связь объединяет две сущности, она называется *бинарной*. Доказано, что n -арный набор связей ($n>2$) всегда можно заменить множеством бинарных.

То число экземпляров сущностей, которое может быть ассоциировано через набор связей с экземплярами другой сущности, называют *степенью связи*. Рассмотрение степеней особенно полезно для бинарных связей.

Степень связи один к одному (обозначается $1 : 1$) означает, что в такой связи экземпляру одной сущности всегда соответствует не более одного экземпляра связанной с ней другой сущности. Этот факт представлен на рис. 7.3, где прямоугольники обозначают сущности, а ромб – связь. Так как степень связи для каждой сущности равна 1, то они соединяются одной линией.

Различия между диаграммами рис. 7.3 являются следствием того, должны или не должны все экземпляры сущности участвовать в связи. Если все экземпляры данной сущности должны участвовать в связи, то участие называется *обязательным* и этот факт отмечается вертикальной чертой (стилизованной единицей) перпендикулярной связи. Если экземпляры данной сущности могут не участвовать в связи, то участие называ-

ется *необязательным* и связь помечается нулем. *Класс принадлежности* сущности должен быть либо обязательным, либо необязательным и определяться правилами, регламентирующими деятельность организации.

В примере на рис. 7.3 каждая диаграмма представляет единственный набор возможных правил функционирования университета. Только одна из этих диаграмм может быть истинной для организации в каждый момент времени. Перечень правил, которых следует придерживаться для соответствия каждой диаграмме на рис. 7.3, формулируется так:

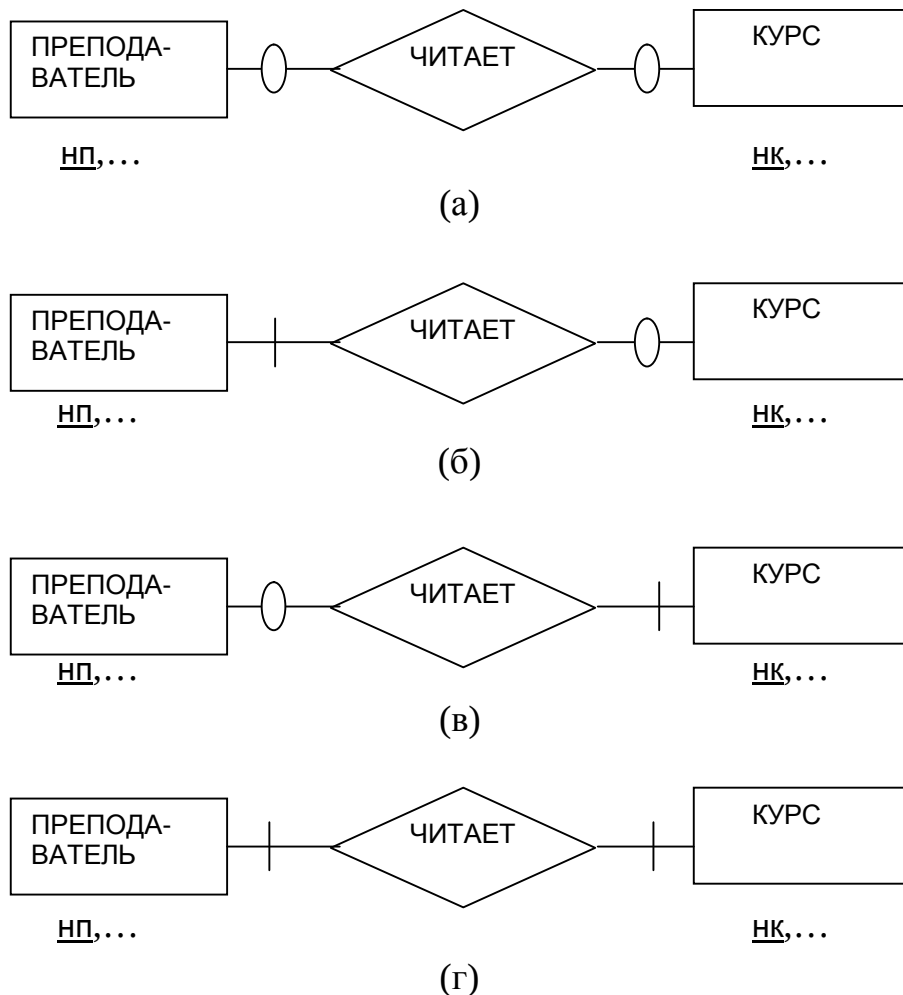


Рис. 7.3. Диаграммы ER-типа с различными сочетаниями классов принадлежности сущностей

Рис. 7.3 (а). Каждый преподаватель читает не более одного курса, и каждый курс читается не более чем одним преподавателем, т.е. допускается наличие преподавателей, не читающих ни одного курса, а также

курсов, не читаемых вовсе. Таким образом, класс принадлежности обеих сущностей необязательный.

Рис. 7.3 (б). Каждый преподаватель читает только один курс, а каждый курс читается не более чем одним преподавателем. Класс принадлежности сущности ПРЕПОДАВАТЕЛЬ является обязательным.

Рис. 7.3 (в). Каждый преподаватель читает не более одного курса, а каждый курс читается только одним преподавателем. Класс принадлежности сущности КУРС является обязательным.

Рис. 7.3 (г). Каждый преподаватель читает только один курс, и каждый курс читается только одним преподавателем. Класс принадлежности обеих сущностей является обязательным.

В диаграммах ER-типа непосредственно под блоком каждой сущности выписывается и выделяется подчеркиванием ключ этой сущности: нп (номер преподавателя) для сущности ПРЕПОДАВАТЕЛЬ и нк (номер курса) для сущности КУРС. Точки, расположенные после каждого из этих атрибутов, указывают на то, что никакие другие имеющиеся атрибуты соответствующей сущности не могут быть частью ее ключа. Эти другие атрибуты будут добавлены после разработки отношений.

На рис. 7.4 представлена диаграмма ER-экземпляров, соответствующая случаю 7.3 (б).

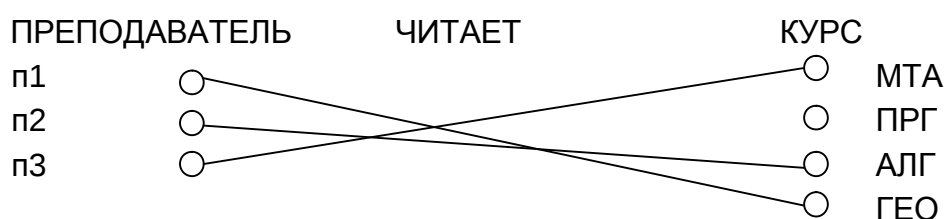


Рис.7.4. Класс принадлежности сущности ПРЕПОДАВАТЕЛЬ является обязательным

Степень связи один ко многим (1 : n). В данном случае экземпляру одной сущности может соответствовать любое число экземпляров другой сущности.

Для используемого примера правило будет звучать следующим образом. Каждый преподаватель может читать одновременно несколько курсов, но каждый курс читается не более чем одним преподавателем.

Этот случай может иметь несколько подвариантов, а именно класс принадлежности может быть обязательным или необязательным – для одной из двух, ни для одной или для обеих сущностей.

На рис. 7.5 показаны диаграммы ER-типа (а) и ER-экземпляров (б) для случая степени связи ПРЕПОДАВАТЕЛЬ-КУРС один ко многим и необязательным классом принадлежности обеих сущностей. На диаграммах ER-типа степень связи n отображается «древовидной линией».

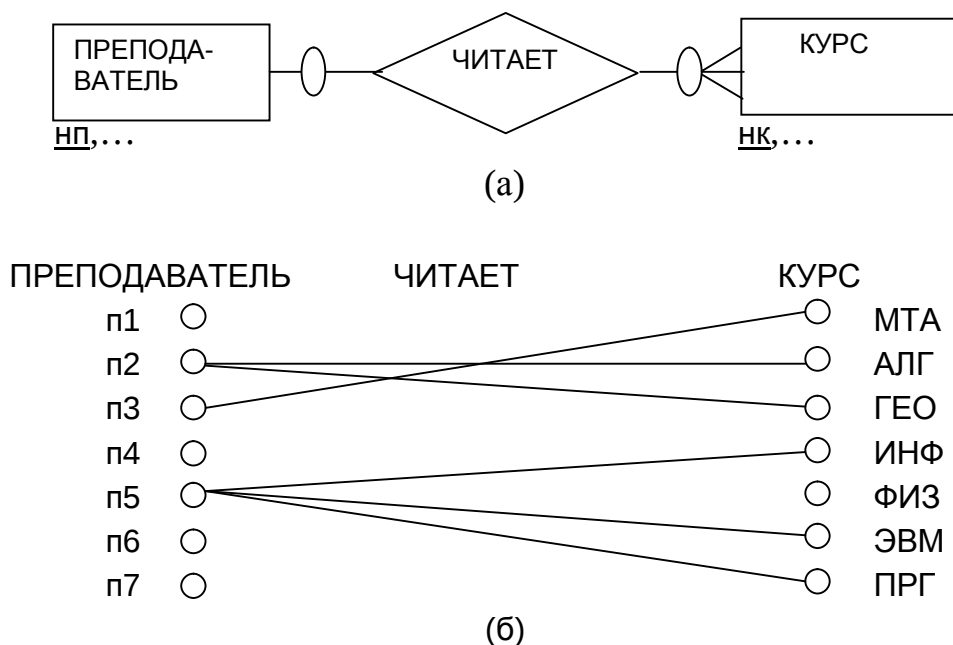


Рис. 7.5. Диаграммы для степени связи $1 : n$

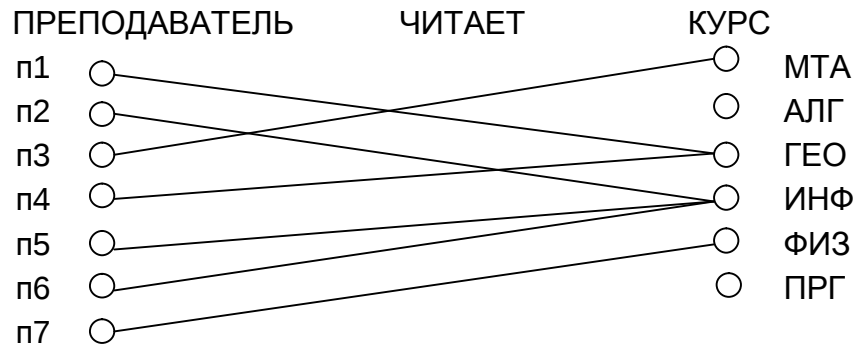
Таким образом, каждый экземпляр курса может быть связан не более чем с одним экземпляром преподавателя (отсюда 1), а каждый экземпляр преподавателя может быть связан с более чем с одним экземпляром курса (отсюда получаем n).

Степень связи многие к одному ($n : 1$). Эта связь аналогична отображению $1 : n$. Для рассматриваемого примера такая связь может быть следствием следующего правила функционирования университета: каждый преподаватель читает на более одного курса, но каждый курс может читаться сразу несколькими преподавателями. Этот случай также может иметь четыре подварианта, зависящих от класса принадлежности сущностей.

На рис. 7.6 показаны диаграммы ER-типа (а) и ER-экземпляров (б) для случая степени связи ПРЕПОДАВАТЕЛЬ-КУРС многие к одному и обязательным классом принадлежности сущности ПРЕПОДАВАТЕЛЬ.



(а)



(б)

Рис. 7.6. Диаграммы для степени связи $n : 1$

Таким образом, экземпляр курса может быть связан с более чем одним экземпляром преподавателя (получаем n), а каждый экземпляр преподавателя связывается с не более чем одним экземпляром курса (имеем 1).

Степень связи многие ко многим ($m : n$). В этом случае каждая из ассоциированных сущностей может быть представлена любым количеством экземпляров. К такой степени связности приводит следующее правило: каждый преподаватель может читать несколько курсов и каждый курс может читаться несколькими преподавателями. В зависимости от класса принадлежности каждой сущности и этот случай может иметь четыре подварианта.

На рис. 7.7 показаны диаграммы ER-типа (а) и ER-экземпляров (б) для случая степени связи ПРЕПОДАВАТЕЛЬ-КУРС многие ко многим и необязательным классом принадлежности обеих сущностей.

Таким образом, каждый экземпляр курса может быть связан более чем с одним экземпляром преподавателя (отсюда m) и каждый экземпляр преподавателя может быть связан более чем с одним экземпляром курса (получаем n).

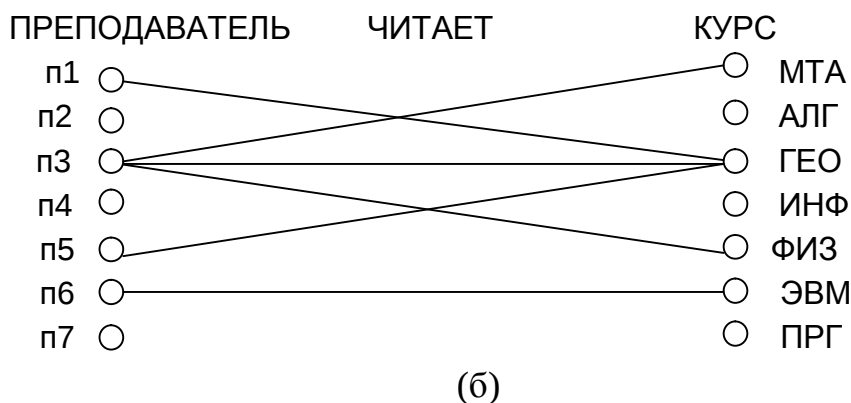
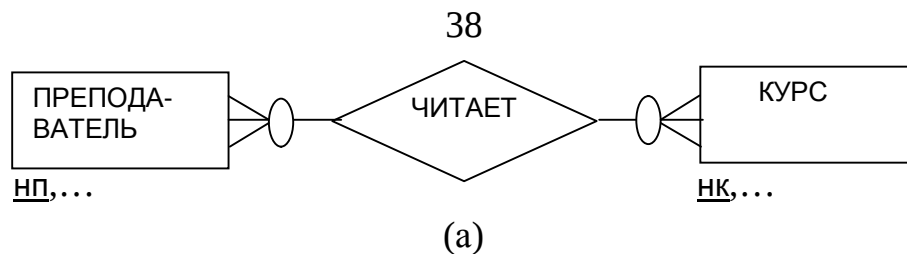


Рис. 7.7. Диаграммы для степени связи $m : n$

2.3. Получение отношений из диаграмм ER-типа

Общий подход к построению БД с использованием ER-метода состоит, прежде всего, в построении диаграммы ER-типа, включающей в себя все необходимые сущности и связи.

Второй шаг в процессе проектирования состоит в построении набора предварительных отношений и указании предполагаемого первичного ключа для каждого отношения.

Последний шаг состоит в подготовке списка всех представляющих интерес атрибутов из тех, которые не были уже перечислены в диаграмме ER-типа в качестве ключей сущности. Каждый из добавленных атрибутов назначается одному из предварительных отношений с условием, чтобы эти отношения находились в НФБК. На этом последнем шаге для каждого отношения должны быть определены межатрибутные ФЗ, с помощью которых проверяется соответствие отношений НФБК. Если полученные в итоге отношения не находятся в НФБК или если нескольким атрибутам не находится логически обоснованных мест в предварительных отношениях, то в этих случаях необходимо пересмотреть ER-диаграммы на предмет устранения возникших недостатков.

2.3.1. Предварительные отношения для бинарных связей степени 1:1

Перечень общих правил генерации отношений из диаграмм ER-типа можно получить, опираясь на класс принадлежности и степень отношения как на определяющие факторы. С целью упрощения вывода этих правил будем использовать связь ПРЕПОДАВАТЕЛЬ ЧИТАЕТ КУРС и ограничимся случаями, в которых степень бинарной связи равна 1:1.

Простейшим решением вопроса о количестве предварительных отношений, необходимых для размещения информации, содержащейся в бинарных связях степени 1:1 (рис. 7.3), является необходимость одного отношения. Назовем это отношение ПРЕПОДАВАТЕЛЬ и все дополнительные атрибуты поместим в это отношение. На рис. 7.8 приведен экземпляр такого отношения в случае, когда класс принадлежности является обязательным для обеих сущностей (рис. 7.3 (г)).

ПРЕПОДАВАТЕЛЬ(нп, пфам, птел, нк, продкурс)

ПРЕПОДАВАТЕЛЬ

нп	пфам	птел	нк	продкурс
п1	Иванов	3454	ГЕО	36
п2	Петров	2323	АЛГ	54
п3	Сидоров	7777	МТА	72
п4	Зайцев	7654	ПРГ	36

Рис. 7.8. Экземпляр единичного отношения, соответствующий рис. 7.3. (г)

В этом отношении сущность ПРЕПОДАВАТЕЛЬ была дополнена двумя атрибутами: фамилия преподавателя (пфам) и телефон преподавателя (птел). Один атрибут добавлен к сущности курс: продолжительность курса (продкурс).

В этом специальном случае одного отношения вполне достаточно. Так как степень связи здесь 1:1 и класс принадлежности является обязательным как для сущности ПРЕПОДАВАТЕЛЬ, так и для сущности КУРС, гарантируется однократное появление каждого значения нп и каждого значения нк в любом экземпляре отношения. Это значит, что от-

ношение никогда не будет содержать ни пустой информации, ни повторяющихся групп избыточных данных. Ключ сущности ПРЕПОДАВАТЕЛЬ был избран в качестве первичного ключа для отношения, но также может быть использован ключ сущности КУРС.

Итак, сформулируем первое правило генерации отношений.

ПРАВИЛО 1. Если степень бинарной связи равна 1:1 и класс принадлежности обеих сущностей является обязательным, то требуется только одно отношение. Первичным ключом этого отношения может быть ключ любой из двух сущностей.

Если степень связи равна 1:1 и класс принадлежности одной сущности является обязательным, а другой – необязательным, то одного отношения недостаточно. На рис. 7.9 приведен экземпляр отношения в том случае, когда класс принадлежности сущности ПРЕПОДАВАТЕЛЬ является обязательным, а сущности КУРС – необязательным (см. рис. 7.3 (б) и 7.4). В этом случае пробелы появляются во всех кортежах, содержащих информацию о курсах, не читаемых ни одним из преподавателей.

ПРЕПОДАВАТЕЛЬ(нп, пфам, птел, нк, продкурс)

ПРЕПОДАВАТЕЛЬ

нп	пфам	птел	нк	продкурс
п1	Иванов	3454	ГЕО	36
п2	Петров	2323	АЛГ	54
п3	Сидоров	7777	МТА	72
---	-----	-----	ПРГ	36

Рис. 7.9. Экземпляр единичного отношения, соответствующий рис. 7.3 (б) и 7.4.

Способ исключения пробелов состоит в использовании вместо одного отношения двух. Каждое отношение будет содержать информацию, касающуюся одной сущности. Кроме того, для связи двух отношений ключ сущности, класс принадлежности которой является необязательным, необходимо поместить в отношение, содержащее информацию о сущности, класс принадлежности которой является обязательным (рис. 7.10).

ПРЕПОДАВАТЕЛЬ(нп, пфам, птел, нк)
 КУРС(нк, продкурс)

ПРЕПОДАВАТЕЛЬ				КУРС	
нп	пфам	птел	нк	нк	продкурс
п1	Иванов	3454	ГЕО	ГЕО	36
п2	Петров	2323	АЛГ	АЛГ	54
п3	Сидоров	7777	МТА	МТА	72
				ПРГ	36

Рис. 7.10. Экземпляры двух отношений, в которых содержатся данные, приведенные на рис. 7.3 (б) и 7.4

ПРАВИЛО 2. Если степень бинарной связи равна 1:1 и класс принадлежности одной сущности является обязательным, а другой – необязательным, то необходимо построение двух отношений. Под каждую сущность необходимо выделить одно отношение, причем ключ сущности должен служить первичным ключом для соответствующего отношения. Кроме того, ключ сущности, для которой класс принадлежности является необязательным, добавляется в качестве атрибута в отношение, выделенное для сущности с обязательным классом принадлежности.

Воспользовавшись этим правилом в ситуации, представленной на рис. 7.3 (в), где класс принадлежности сущности КУРС является обязательным, а сущности ПРЕПОДАВАТЕЛЬ – необязательным, получим следующие отношения:

ПРЕПОДАВАТЕЛЬ(нп, пфам, птел)
 КУРС(нк, продкурс, нп)

В том случае, когда степень бинарной связи равна 1:1 и класс принадлежности ни одной из сущностей не является обязательным, одного отношения недостаточно. При использовании только одного отношения возможны два пути возникновения пробелов (рис. 7.11 (а)). Также недостаточным является использование двух отношений, так как возникают проблемы в связи с внесением ключа одной сущности в отношение, выделенное под другую сущность (рис. 7.11 (б)). Единственное решение за-

ключается в построении трех отношений: по одному для каждой сущности и одного для связи (рис. 7.11(в)). На рис. 7.11 приведены типичные экземпляры отношений, получаемые при использовании одного, двух и трех отношений (см. рис. 7.1, 7.2 и 7.3 (а)). Проблемы возникают везде, за исключением случая использования трех отношений.

ПРЕПОДАВАТЕЛЬ				
Нп	пфам	птел	нк	продкурс
п1	Иванов	3454	ГЕО	36
п2	Петров	2323	----	---
п3	Сидоров	7777	МТА	72
п4	Зайцев	7654	ПРГ	36
--	-----	-----	АЛГ	54

(а) Использование одного отношения

ПРЕПОДАВАТЕЛЬ				КУРС		
нп	пфам	птел	нк	нк	продкурс	нп
п1	Иванов	3454	ГЕО	МТА	72	п3
п2	Петров	2323	-----	ГЕО	36	п1
п3	Сидоров	7777	МТА	АЛГ	54	---
п4	Зайцев	7654	ПРГ	ПРГ	36	п4

(б) Использование двух отношений

ПРЕПОДАВАТЕЛЬ			КУРС		ЧИТАЕТ	
нп	пфам	птел	нк	продкурс	нп	нк
п1	Иванов	3454	МТА	72	п1	ГЕО
п2	Петров	2323	ГЕО	36	п3	МТА
п3	Сидоров	7777	АЛГ	54	п4	ПРГ
п4	Зайцев	7654	ПРГ	36		

(в) Использование трех отношений

Рис. 7.11. Возможные реляционные формы для случая бинарной связи 1:1, когда ни один из классов принадлежности не является обязательным

В отношении ЧИТАЕТ на рис. 7.11 (в) любые значения, как номера преподавателя, так и номера курса могут появиться только однажды, т.к. степень равна 1:1. Кроме того, это отношение содержит номера курсов только тех, которые читаются, и номера преподавателей только тех, ко-

которые читают курс. Отношение ПРЕПОДАВАТЕЛЬ содержит информацию обо всех преподавателях и отношение КУРС содержит информацию обо всех курсах.

ПРАВИЛО 3. Если степень бинарной связи равна 1:1 и класс принадлежности ни одной сущности не является обязательным, то необходимо использовать три отношения: по одному для каждой сущности, ключи которых служат в качестве первичных в соответствующих отношениях, и одного для связи. Среди своих атрибутов отношение, выделяемое связи, будет иметь по одному ключу сущности от каждой сущности.

2.3.2. Первый пример ER-проектирования

Проектируется БД, предназначенная для хранения информации о лаборантах и лабораториях информатики, которые ими обслуживаются. Разрешается закрепление не более одного лаборанта за одной лабораторией, и каждый лаборант обслуживает только одну лабораторию. Представляющими интерес атрибутами являются: фамилия лаборанта (л_фам), номер телефона (тном), оплата (плата), максимально допустимое число студентов в лаборатории (размер), номер лаборатории (л_ном), рейтинг лаборатории (рейтинг) и основной вид программного обеспечения, изучаемый в лаборатории (вид).

Сущностями в данном случае будут ЛАБОРАНТ и ЛАБОРАТОРИЯ, и связь между ними ОБСЛУЖИВАЕТ. Диаграммы ER-экземпляров и ER-типа представлены на рис. 7.12.

Предположениями, учитывающимися при построении диаграмм, являются: все лаборанты закреплены за лабораториями, фамилии лаборантов уникальны, номера лабораторий уникальны, некоторые лаборатории лаборантами не обслуживаются.

Так как степень отношения равна 1:1 и класс принадлежности одной сущности (ЛАБОРАНТ) является обязательным, а другой (ЛАБОРАТОРИЯ) – нет, то используется правило 2 для генерации предварительных отношений.

Атрибуты, не используемые в качестве ключей сущностей, распределяются по отношениям естественным образом: тном и плата – в отношение ЛАБОРАНТ, так как они содержат информацию о лаборантах;

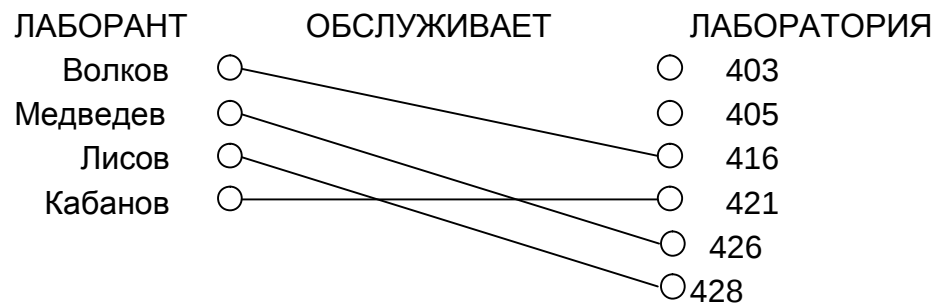
рейтинг, размер и вид помещаются в отношение ЛАБОРАТОРИЯ, так как в них содержится информация о лабораториях:

ЛАБОРАНТ(л_фам, тном, плата, л_ном)

ЛАБОРАТОРИЯ(л_ном, рейтинг, размер, вид)

ЛАБОРАНТ(л_фам,....., л_ном)

ЛАБОРАТОРИЯ(л_ном,.....)



(а)

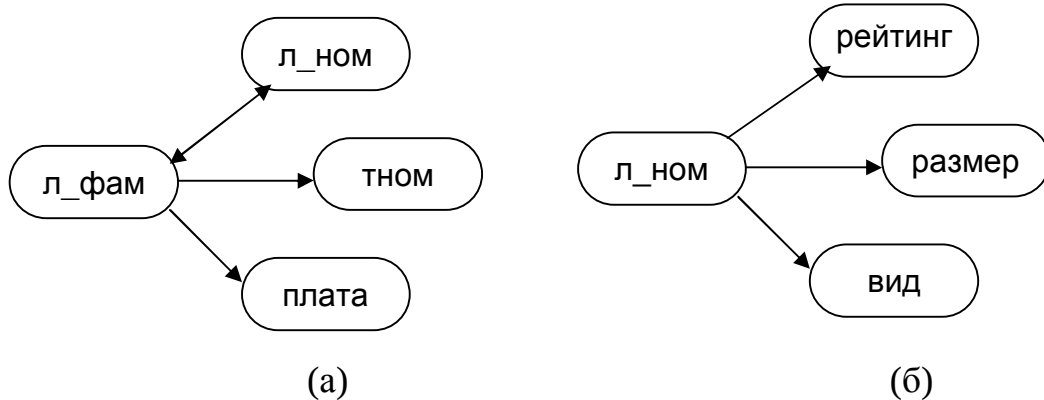


Другие атрибуты: тном, плата, размер, рейтинг, вид

(б)

Рис. 7.12. Диаграммы ER-экземпляров (а) и ER-типа (б) для первого примера ER-проектирования

На рис. 7.13 для обоих отношений приведены диаграммы ФЗ, позволяющие заключить, что каждое отношение находится в НФБК.



(а)

(б)

Рис. 7.13. Диаграммы ФЗ для отношений ЛАБОРАНТ (а) и ЛАБОРАТОРИЯ (б) для первого примера ER-проектирования

На рис. 7.14 приводятся типичные экземпляры отношений, используемых при создании БД.

ЛАБОРАНТ				ЛАБОРАТОРИЯ			
л_фам	тном	плата	л_ном	л_ном	рейтинг	размер	вид
Волков	9-16	400	416	403	средн	10	WEB-дизайн
Медведев	9-26	450	426	405	средн	10	Office
Лисов	9-28	450	428	416	низк	11	Языки
Кабанов	9-21	380	421	421	средн	15	СУБД
				426	высок	12	Visual-среды
				428	высок	12	Мультимедиа

Рис. 7.14. Типичные экземпляры отношений ЛАБОРАНТ и ЛАБОРАТОРИЯ

2.3.3. Предварительные отношения для бинарных связей степени 1:N (N:1)

Для случая бинарных связей степени $1:n$ ($n:1$) требуется только два правила. Фактором, определяющим выбор одного из этих правил, является класс принадлежности n -связной сущности; класс принадлежности 1-связной сущности не влияет на конечный результат в обоих случаях.

На рис. 7.15 (б) показан экземпляр отношения ПРЕПОДАВАТЕЛЬ, содержащего данные, приведенные на рис. 7.6. Это случай степени $n:1$ с необязательным классом принадлежности 1-связной сущности и обязательным – n -связной сущности. Отчетливо видны две проблемы, связанные с этим отношением. Пробелы обнаруживаются в тех полях атрибутов преподавателя, где курс не читается ни одним преподавателем; повторение полей данных о курсах наблюдается там, где курс читается несколькими преподавателями. Например, трижды появляется информация о курсе ИНФ. Если бы класс принадлежности 1-связной сущности был обязательным, то исчезли бы пробелы, но повторяющиеся группы данных в полях атрибутов курса сохранились.

Решить все эти проблемы, вне зависимости от класса принадлежности 1-связной сущности, можно согласно следующему правилу.

ПРАВИЛО 4. Если степень бинарной связи равна 1:n (n:1) и класс принадлежности n-связной сущности является обязательным, то достаточным является использование двух отношений, по одному на каждую сущность. При этом первичным ключом каждого отношения является ключ соответствующей сущности. Дополнительно для связи отношений ключ 1-связной сущности должен быть добавлен как атрибут в отношение, отводимое n-связной сущности.



(а)

ПРЕПОДАВАТЕЛЬ

нп	пфам	птел	нк	продкурс
п1	Иванов	3454	ГЕО	36
п4	Зайцев	7654	ГЕО	36
п2	Петров	2323	ИНФ	72
п5	Николаев	4114	ИНФ	72
п6	Сергеев	2345	ИНФ	72
п3	Сидоров	7777	МТА	72
п7	Климов	5432	ФИЗ	36
--	-----	-----	АЛГ	54
--	-----	-----	ПРГ	36

(б)

Рис. 7.15. Использование одного отношения для бинарной связи типа n:1 в том случае, когда класс принадлежности n-связной сущности является обязательным, а 1-связной – необязательным

На рис. 7.16 приведены примеры двух отношений, построенных с помощью этого правила и содержащие ту же информацию, которая представлена на рис. 7.15. Все пробелы и повторяющиеся группы данных исчезли.

На рис. 7.17 показан экземпляр отношения КУРС, содержащий данные из рис. 7.5. Это случай связи степени 1:n с необязательным классом принадлежности обеих сущностей.

ПРЕПОДАВАТЕЛЬ

нп	пфам	птел	нк
----	------	------	----

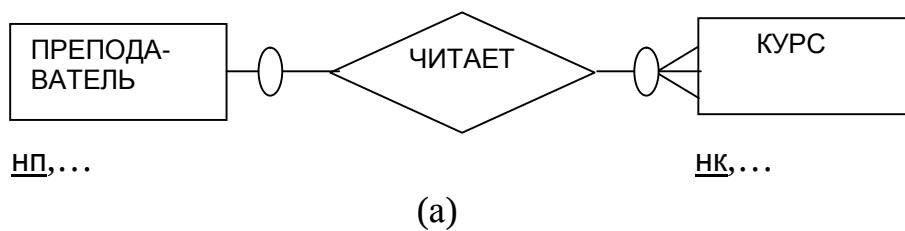
КУРС

нк	продкурс
----	----------

п1	Иванов	3454	ГЕО
п2	Петров	2323	ИНФ
п3	Сидоров	7777	МТА
п4	Зайцев	7654	ГЕО
п5	Николаев	4114	ИНФ
п6	Сергеев	2345	ИНФ
п7	Климов	5432	ФИЗ

ГЕО	36
ИНФ	72
МТА	72
ФИЗ	36
АЛГ	54
ПРГ	36

Рис. 7.16. Данные, приведенные на рис. 7.15, после их разнесения по двум отношениям с помощью ПРАВИЛА 4



КУРС

нк	продкурс	нп	пфам	птел
МТА	72	п3	Сидоров	7777
АЛГ	54	п2	Петров	2323
ГЕО	36	п2	Петров	2323
ИНФ	72	п5	Николаев	4114
ФИЗ	36	---	-----	-----
ЭВМ	54	п5	Николаев	4114
ПРГ	36	п5	Николаев	4114
-----	----	п1	Иванов	3454
-----	----	п4	Зайцев	7654
-----	----	п6	Сергеев	2345

(б)

Рис. 7.17. Использование одного отношения для бинарной связи типа 1:n в случае, когда класс принадлежности обеих сущностей является необязательным

В связи с этим отношением отчетливо видны три проблемы. Проблемы возникают в полях атрибутов курса там, где преподаватель не читает курс, и в тех полях атрибутов преподавателя, где курс не читается

нк	продкурс
МТА	72
АЛГ	54
ГЕО	36
ИНФ	72
ФИЗ	36
ЭВМ	54
ПРГ	36

нп	пфам	птел
п1	Иванов	3454
п2	Петров	2323
п3	Сидоров	7777
п4	Зайцев	7654
п5	Николаев	4114
п6	Сергеев	2345
п7	Климов	5432

ЧИТАЕТ

нп	нк
п2	АЛГ
п2	ГЕО
п3	МТА
п5	ИНФ
п5	ЭВМ
п5	ПРГ

нк является первичным ключом
в силу существующей связи n:1
между нк и нп

Рис. 7.19. Размещение данных, приведенных на рис. 7.17 с помощью ПРАВИЛА 5

2.3.4. Предварительные отношения для бинарных связей степени M:N

Если степень бинарной связи равна $m:n$, то для хранения данных требуются три отношения вне зависимости от класса принадлежности, как первой, так и второй сущностей. При использовании одного или двух отношений неизбежно возникновение пробелов и/или повторяющихся групп данных в экземплярах этих отношений; какая из двух проблем возникает при использовании двух отношений зависит от классов принадлежности двух сущностей.

ПРАВИЛО 6. Если степень бинарной связи равна $m:n$, то для хранения данных необходимо три отношения: по одному для каждой сущности, причем ключ каждой сущности используется в качестве первичного ключа соответствующего отношения и одного отношения для связи. Последнее отношение должно иметь в числе своих атрибутов ключ сущности каждой сущности.

На рис. 7.20 приведены экземпляры отношений, содержащих данные из рис. 7.7 (степень равна $m:n$, и ни один класс принадлежности не является обязательным). В этих экземплярах отсутствуют как пробелы, так и повторяющиеся группы данных.

КУРС		ПРЕПОДАВАТЕЛЬ		
нк	продкурс	нп	пфам	птел
МТА	72	п1	Иванов	3454
АЛГ	54	п2	Петров	2323
ГЕО	36	п3	Сидоров	7777
ИНФ	72	п4	Зайцев	7654
ФИЗ	36	п5	Николаев	4114
ЭВМ	54	п6	Сергеев	2345
ПРГ	36	п7	Климов	5432

ЧИТАЕТ	
нп	нк
п1	АЛГ
п3	МТА
п3	ГЕО
п3	ФИЗ
п5	ГЕО
п6	ЭВМ

Это отношение целиком
является ключом

Рис. 7.20. Размещение данных, приведенных на рис. 7.7 с помощью ПРАВИЛА 6

В случае отсутствия ФЗ, связывающих атрибуты отношения (как в случае отношения ЧИТАЕТ), первичный ключ этого отношения требует использования всех его атрибутов. Первичным ключом отношения ЧИТАЕТ является $\langle \text{нп}, \text{нк} \rangle$. Об отношении этого типа говорят, что оно целиком является ключом (оно всегда находится в НФБК).

2.3.5. Второй пример ER-проектирования

Рассмотрим решение задачи о лаборантах, определение которой было дано в 7.3.2. Проектируется БД, предназначенная для хранения информации о лаборантах и лабораториях информатики, которые ими обслуживаются. Руководство университета не препятствует тому, чтобы

сразу несколькими лаборантами обслуживалась одна лаборатория. Однако каждый лаборант должен обслуживать только одну лабораторию. Каждая лаборатория специализируется на изучении нескольких видов программного обеспечения. Администрацию интересует количество часов в неделю, отводимое для аудиторных занятий и самостоятельной работы по изучению каждого вида программного обеспечения.

Представляющими интерес атрибутами являются: фамилия лаборанта (л_фам), номер телефона (тном), оплата (плата), максимально допустимое число студентов в лаборатории (размер), номер лаборатории (л_ном), рейтинг лаборатории (рейтинг), основные виды программного обеспечения, изучаемые в лаборатории (вид), количество часов в неделю аудиторных (ауд) и самостоятельных (сам) занятий студентов по видам программного обеспечения (ПО)

Выделим сущности – ЛАБОРАНТ, ЛАБОРАТОРИЯ и ПО и связи – ОБСЛУЖИВАЕТ и ИЗУЧАЕТСЯ, как показано на рис. 7.21, где приводятся диаграммы типа и типичные диаграммы экземпляров.

Присутствуют две связи: одна степени $n:1$ и другая – $m:n$. Необходимо использовать правила 4 и 6 для получения четырех предварительных отношений: по одному для каждой сущности и одно для связи ИЗУЧАЕТСЯ. Кроме того, атрибут `л_ном` сущности ЛАБОРАТОРИЯ необходимо добавить в отношение ЛАБОРАНТ (согласно ПРАВИЛУ 4). В результате получаются следующие предварительные отношения:

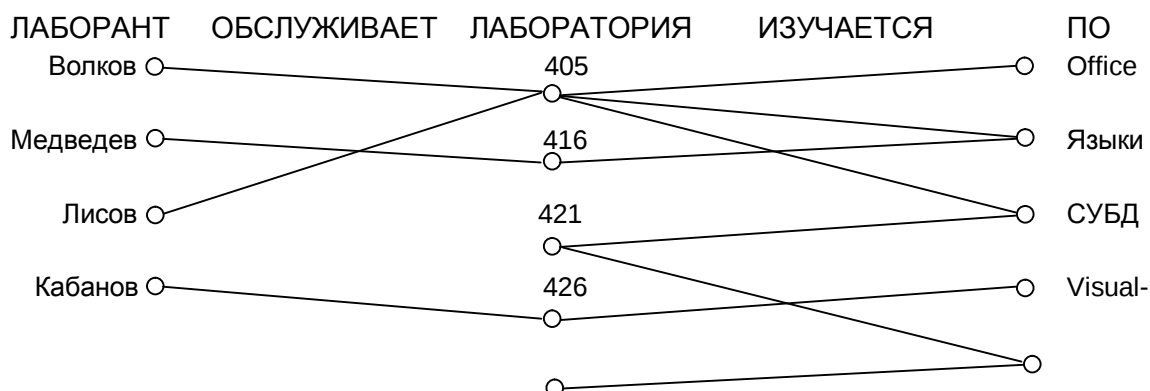
ЛАБОРАНТ (Л-фам,, Л-НОМ)

ЛАБОРАТОРИЯ (Л-НОМ,.....)

ЛАБ_ПО (Л-НОМ, ВИД,)

ПО (ВИД,)

Здесь предварительное отношение, полученное на основании связи ИЗУЧАЕТСЯ, названо ЛАБ ПО.



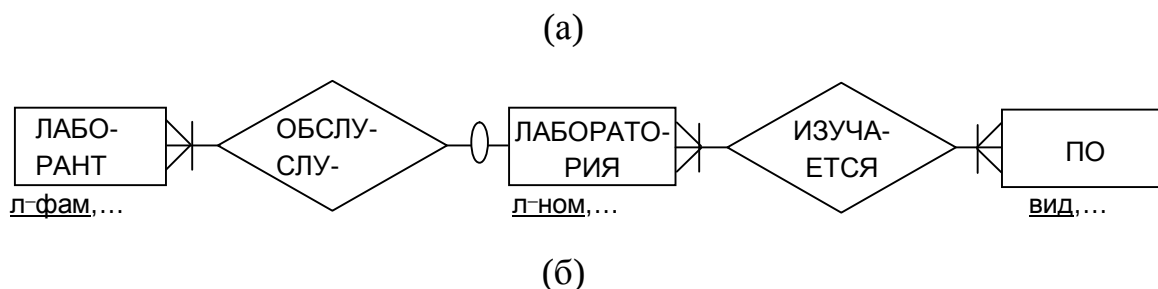


Рис. 7.21. Диаграммы ER-экземпляров (а) и ER-типа (б) для первого примера ER-проектирования

Как и в случае первого примера ER-проектирования, не сложно найти место атрибутам, не являющимся частью ключей сущностей. Размещение их завершает формирование окончательных отношений и проводится следующим образом: тном и плата – это свойства сущности ЛАБОРАНТ и помещаются в соответствующее отношение. Атрибуты рейтинг и размер являются свойствами сущности ЛАБОРАТОРИЯ, в то время как ауд и сам относятся к сущности ПО. Единственное предварительное отношение, не получающее дополнительных атрибутов, это ЛАБ_ПО. Как оказалось, оно целиком является ключом.

ЛАБОРАНТ (л_фам, тном, плата, л_ном)

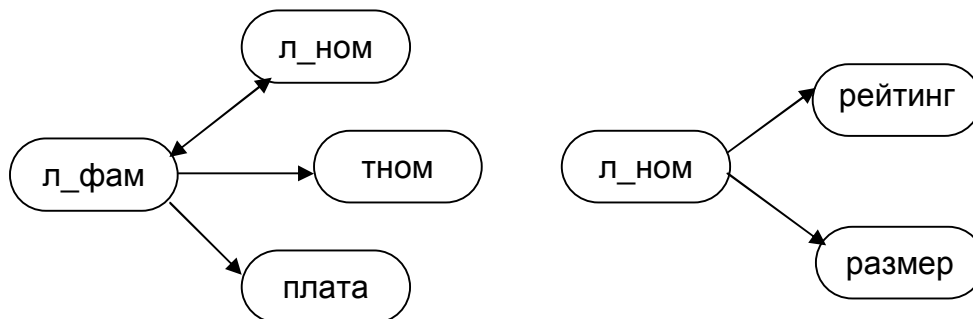
ЛАБОРАТОРИЯ (л_ном, рейтинг, размер)

ЛАБ_ПО (л_ном, вид)

ПО (вид, ауд, сам)

На рис. 7.22 показаны диаграммы ФЗ для всех отношений. Каждое из этих отношений уже находится в НФБК и никаких дополнительных преобразований не требуется.

Типичные примеры каждого из отношений приведены на рис. 7.23.



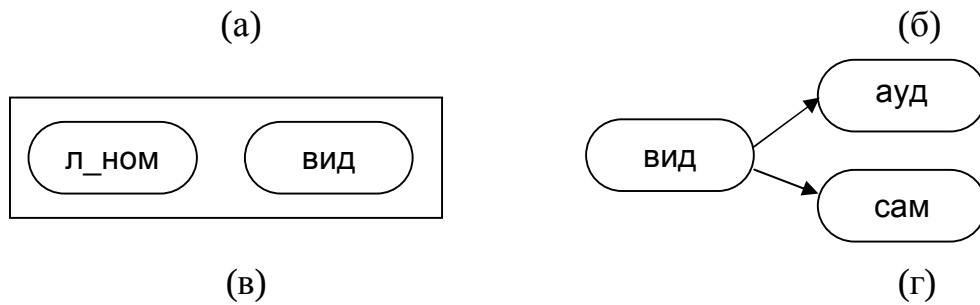


Рис. 7.22. Диаграммы ФЗ для отношений ЛАБОРАНТ (а), ЛАБОРАТОРИЯ (б), ЛАБ_ПО (в) и ПО (г), представленных на рис. 7.21

ЛАБОРАНТ			
л_фам	тном	плата	л_ном
Волков	9-05	400	405
Медведев	9-16	450	416
Лисов	9-05	450	405
Кабанов	9-26	380	426

ЛАБОРАТОРИЯ		
л_ном	рейтинг	размер
405	средн	10
416	низк	11
421	средн	15
426	высок	12
428	высок	12

ЛАБ_ПО	
л_ном	вид
405	Office
405	Языки
405	СУБД
416	Языки
421	СУБД
421	Мультимедиа
426	Visual-среды
428	Мультимедиа

ПО		
вид	ауд	сам
Office	16	10
Языки	36	24
СУБД	24	18
Visual-среды	36	24
Мультимедиа	24	12

Рис. 7.23. Типичные экземпляры отношений второго примера ER-проектирования

3. Дополнительные конструкции, используемые в ER-методе

В гл. 7 был представлен метод генерации отношений БД, основанный на использовании бинарных связей. Хотя с помощью бинарных связей могут быть описаны многие ситуации реального мира, тем не менее, неизбежно возникновение и таких ситуаций, в которых построение разумной модели организации невозможно без привлечения дополнительных конструкций.

3.1. Необходимость связей более высокого порядка

В гл. 7 на двух примерах изучались возможности применения ER-метода. В основе этих примеров лежит необходимость хранения информации о лаборантах, лабораториях, обслуживаемых лаборантами, видах программного обеспечения, изучаемых в этих лабораториях. Предположим, что администрация, нанимающая лаборанта, хочет знать, с каким видом программного обеспечения предпочитает работать лаборант. Если посмотреть на диаграмму экземпляров, изображенную на рис. 7.21 (а), то может показаться естественным заключение о том, что раз Волков обслуживает лабораторию 405 и в этой лаборатории изучаются офисные программы, языки программирования и СУБД, то, следовательно, Волков предпочитает именно это программное обеспечение. Это может соответствовать истине, а может и не соответствовать. Если связь между сущностями ЛАБОРАНТ и ПО существует, то эта связь, назовем ее ПРЕДПОЧИТАЕТ, должна быть представлена в обеих диаграммах – типа и экземпляров, как на рис. 8.1.

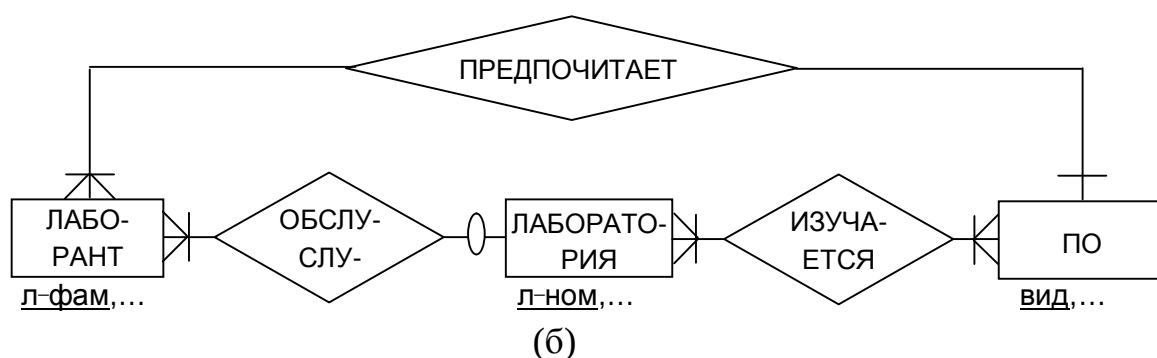
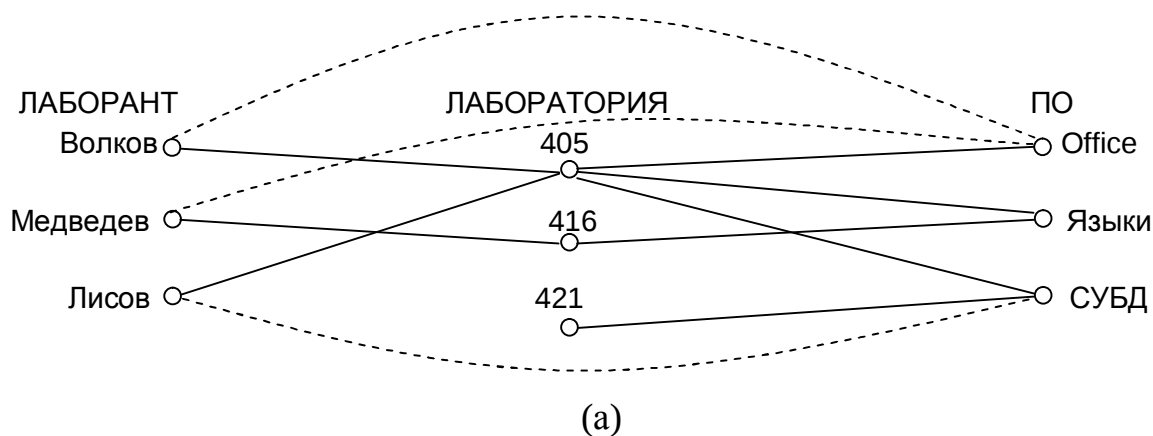


Рис. 8.1. Диаграммы ER-экземпляров (а) и ER-типа (б) в случае, когда лаборанты отдают предпочтение некоторым видам программного обеспечения

Пунктирные линии использованы с целью выделения экземпляров связи ПРЕДПОЧИТАЕТ. Кроме того, приведены только несколько экземпляров. Из диаграммы экземпляров следует, что Волков обслуживает лабораторию 405, что в лаборатории 405 изучается три вида программного обеспечения, а также то, что Волков предпочитает работать с офисными программами. В силу того, что связь между сущностями ЛАБОРАНТ и ПО имеет степень $n:1$, и, так как каждый лаборант обслуживает только одну лабораторию, не возникает проблемы выбора лаборанта для данной лаборатории, где изучается определенный вид программного обеспечения. Действительно, для построения отношений, отражающих ситуацию, приведенную на рис. 8.1, единственным изменением, которое требуется внести в отношения на рис. 7.23, является добавление в отношение ЛАБОРАНТ атрибута вида из отношения ПО. Этот атрибут указывает вид ПО, который предпочитает данный лаборант. Таким образом, в этом расширенном примере бинарные связи также позволяют получать адекватную ситуации модель.

Недостаточность бинарных связей для корректного моделирования можно показать на следующей ситуации. Предположим, что лаборанты обслуживают более чем по одной лаборатории, и каждый лаборант может предпочитать один вид ПО в одной лаборатории, а другой – в другой лаборатории. Волков, например, предпочитает работать с офисными программами в лаборатории 405 и с языками программирования в лаборатории 416. Медведев, в свою очередь, обслуживает только лабораторию 416 и предпочитает заниматься языками программирования. Собрав вместе эти данные, получим следующий перечень утверждений:

- Волков обслуживает обе лаборатории – 405 и 416.
- Медведев обслуживает только лабораторию 416.
- Волков предпочитает работать с офисным ПО в лаборатории 405.
- Волков предпочитает работать с языками программирования в 416 лаборатории.
- Медведев предпочитает работать с языками программирования в 416 лаборатории.

На рис. 8.2 приведена диаграмма ER-экземпляров, в которой использованы бинарные связи, включающие только указанные 5 единиц информации.

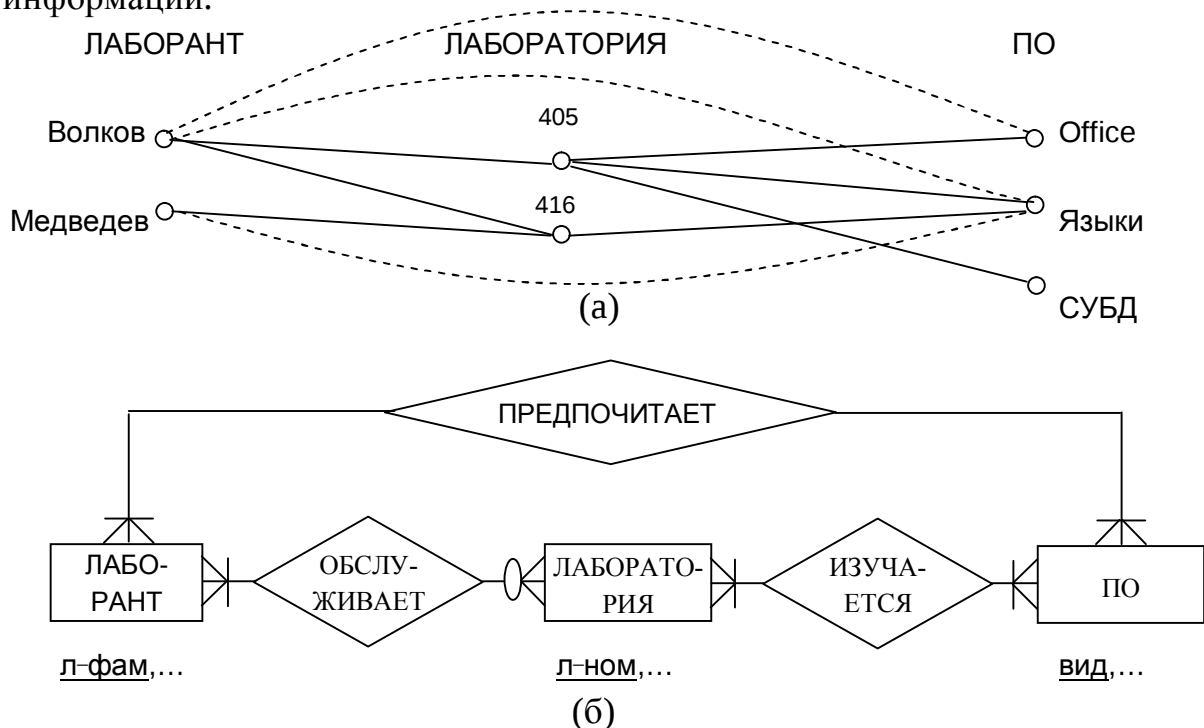


Рис. 8.2. Диаграммы ER-экземпляров (а) и ER-типа (б) в случае, когда все связи имеют степень m:n

Хотя из диаграммы можно вывести все характеризующие Волкова и Медведева утверждения, также можно заключить, что Волков предпочитает работать с языками программирования в лаборатории 405, а это неверно. Проблема возникает из того факта, что в данном случае все связи имеют степень $m:n$ и, следовательно, нет уникального пути, соединяющего вместе все три экземпляра сущности единственным образом. Причину возникновения подобной проблемы можно понять, обратив внимание на то, что каждое из утверждений:

- Волков предпочитает работать с офисным ПО в лаборатории 405;
- Волков предпочитает работать с языками программирования в 416 лаборатории;

связывает вместе три информационных единицы. Та же информация не может быть заключена в выражениях:

- Волков обслуживает лабораторию 405 и лабораторию 416;
- В лабораториях 405 и 416 изучаются офисные программы, языки программирования и СУБД;
- Волков предпочитает офисные программы и языки программирования.

Суть в том, что информационные триплеты нельзя представить в виде набора из трех бинарных связей. Правильная модель приводится на рис. 8.3 и требует использования трехсторонних связей.

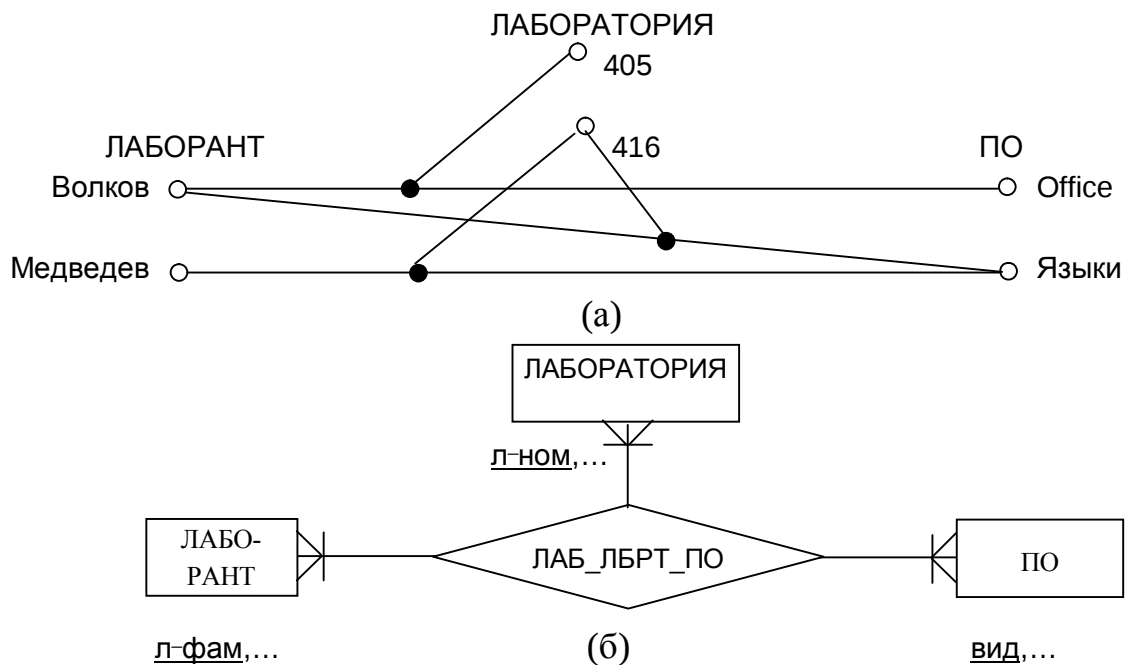


Рис. 8.3. Диаграммы ER-экземпляров (а) и ER-типа (б) в случае трехсторонней связи ЛАБ_ЛБРТ_ПО

3.2. Предварительные отношения для трехсторонних связей

В случае трехсторонних связей предварительные отношения генерируются на основании следующего правила.

ПРАВИЛО 7. В случае трехсторонней связи необходимо использовать четыре предварительных отношения, по одному для каждой сущности, причем ключ каждой сущности должен служить в качестве первичного ключа для соответствующего отношения, и одно для связи. Отношение, порожаемое связью, будет иметь среди своих атрибутов ключи сущности от каждой сущности. (Аналогично, когда связь n -сторонняя, требуется $n+1$ предварительное отношение).

Если применить это правило к данным, приведенным на рис. 8.3, то будут получены отношения:

ЛАБОРАНТ (л_фам,....)

ЛАБОРАТОРИЯ (л_ном,.....)

ПО (вид,....)

ЛАБ_ЛБРТ_ПО (л_фам, л_ном, вид,.....)

Первичный ключ для ЛАБ_ЛБРТ_ПО не может быть определен до тех пор, пока не будут распределены все другие атрибуты. Если воспользоваться всеми теми атрибутами, которые приведены на рис. 7.23, то атрибуты будут распределены следующим образом: отношению ЛАБОРАНТ назначаются атрибуты тном и плата; отношению ЛАБОРАТОРИЯ будут назначены атрибуты рейтинг и размер, отношению ПО назначаются ауд и сам. Отношение ЛАБ_ЛБРТ_ПО не получит никаких других атрибутов. Первичный ключ для ЛАБ_ЛБРТ_ПО будет составным <л_фам, л_ном> в том случае, если каждый лаборант предпочитает работать в лаборатории только с одним видом программного обеспечения. Если число предпочитаемых видов ПО равно двум или более для какой-либо лаборатории, тогда все три атрибута отношения ЛАБ_ЛБРТ_ПО будут составлять ключ. На рис. 8.4 приведены экземпляры четырех отношений в предположении, что каждый лаборант предпочитает работать с одним видом ПО в каждой лаборатории, которая им обслуживается. Нетрудно показать, что каждое из рассмотренных отношений находится в НФБК.

ЛАБОРАНТ

л_фам	тном	плата
Волков	9-05	400
Медведев	9-16	450
Лисов	9-05	450
Кабанов	9-26	380

ЛАБОРАТОРИЯ

л_ном	рейтинг	размер
405	средн	10
416	низк	11
421	средн	15
426	высок	12
428	высок	12

ЛАБ_ЛБРТ_ПО

л_фам	л_ном	вид
Медведев	416	Языки
Волков	416	Языки
Волков	405	Office
Лисов	405	СУБД
Лисов	416	Языка
Кабанов	426	Visual-среды

ПО

вид	ауд	сам
Office	16	10
Языки	36	24
СУБД	24	18
Visual-среды	36	24
Мультимедиа	24	12

Рис. 8.4. Экземпляры отношений, полученные на основании диаграмм, приведенных на рис. 8.3