

Раздел 3. Проектирование баз данных.

Основные этапы проектирования БД. Жизненный цикл БД. Концептуальное и логическое проектирование реляционной БД. Модель "Сущность - Связь". Методология построения ER-диаграмм. Логическое проектирование реляционных баз данных. Проектирование реляционных баз данных на основе нормализации. Нормальные формы (НФ). Понятие 1НФ, 2НФ, 3НФ.

Реляционная база данных представляет собой множество взаимосвязанных двумерных таблиц – реляционных таблиц, называемых также отношениями, в каждой из которых содержатся сведения об одной сущности автоматизируемой предметной области.

Логическую структуру реляционной базы данных образует совокупность реляционных таблиц, между которыми установлены связи.

В таблицах базы должны сохраняться все данные, необходимые для решения задач предметной области. Причем каждый элемент данных должен храниться только в одном экземпляре. Для создания таблиц, соответствующих реляционной модели данных, используется процесс, называемый нормализацией данных.

Нормализация – это удаление из таблиц повторяющихся данных путем их переноса в новые таблицы, записи которых не содержат повторяющихся значений.

Будем считать, что проблема логического проектирования реляционной базы данных состоит в обоснованном принятии решений о том:

- из каких отношений должна состоять база данных и
- какие атрибуты должны быть у этих отношений.

Правила целостности реляционной базы данных. Системы управления реляционными базами данных автоматически применяют правила целостности, но гораздо безопаснее убедиться, что проект БД соответствует правилам сущности и ссылочной целостности. Эти правила приведены в таблице

Таблиц

Правила целостности		
Требование	Цель	Пример
<i>Целостность сущности</i>		
Все записи первичного ключа должны являться уникальными, никакая часть первичного ключа не может быть нулевой.	Каждая строка будет иметь уникальный идентификатор, и значения внешнего ключа должны правильно ссылаться на значения первичного ключа.	У каждого студента уникальный номер, и при этом он не может быть нулевым; следовательно, все студенты однозначно идентифицируются по номеру.
<i>Ссылочная целостность</i>		

Внешний ключ может иметь либо пустое значение, если он не является частью первичного ключа своей таблицы, либо значение, которое соответствует первичному ключу в таблице, с которой он связан. Каждое ненулевое значение внешнего ключа должно иметь ссылку на существующее значение первичного ключа.	Атрибут может не иметь соответствующего значения, но значение не пустое, оно должно совпадать с соответствующим первичным ключом. Применение правила ссылочной целостности делает невозможным удаление строки в одной таблице, первичный ключ которой имеет обязательные совпадающие значения внешнего ключа в другой таблице.	У студента еще может не быть назначенного куратора (номер), но нельзя поставить номер несуществующего куратора.
---	---	---

Пример реляционной БД

Таблица: СТУДЕНТ

Основной ключ: СТ_КОД

Внешний ключ: ПР_КОД

База данных: НВГУ

СТ_КОД	СТ_ИМЯ	СТ_ФАМ	СТ_ОТЦ	СТ_ДР	СТ_ГРУП	СТ_КУРС	СТ_ПЕР	ПР_КОД
6-190701	Анатолий	Канаев	Арсениевич	11.03.2003	3902	2	0	9015
6-190702	Павел	Матвиенко	Потапович	15.05.2003	3902	2	0	9015
6-190703	Карл	Сальников	Трофимович	25.01.2003	3902	2	0	9015
6-190704	Владимир	Клинских	Богданович	30.12.2002	3902	2	0	9015
6-190706	Василиса	Сотова	Владимировна	06.11.2002	3902	2	0	9015
6-190707	Оксана	Журавлева	Павловна	31.10.2002	3902	2	0	9015
6-190708	Дарья	Кудрина	Филипповна	10.09.2003	3902	2	0	9015
6-190709	Виктория	Никитина	Леонидовна	22.05.2003	3902	2	0	9015
6-190710	Матвей	Шипеев	Егорович	21.03.2003	3902	2	1	9015
6-190745	Святослав	Балин	Матвеевич	28.06.2003	3902	2	1	9015

Таблица: ПРЕПОДАВАТЕЛЬ

Основной ключ: ПР_КОД

Внешние ключи: ЗВ_КОД, ДЛЖ_КОД, КАФ_КОД

ПР_КОД	ПР_ИМЯ	ПР_ФАМ	ПР_ОТЦ	ЗВ_КОД	ДЛЖ_КОД	ПР_ТЕЛ	КАФ_КОД
4995	Виталий	Аглеев	Денисович	2	2	73466756646	3
5685	Наталия	Шубина	Юрьевна	2	1	73466374030	3
7451	Марианна	Кокорина	Родионовна	3	2	73466481750	4
8256	Денис	Каретников	Федорович	1	2	73466087024	3
8954	Марина	Медникова	Пахомовна	1	3	73466874437	4
9524	Константин	Стрельцов	Маркович	1	2	73466563452	5
9562	Василий	Праздников	Всеволодович	1	2	73466368287	4

Проектирование реляционных баз данных с использованием аппарата нормализации

Отношения реляционной базы данных содержат как структурную, так и семантическую (смысловую) информацию. Структурная информация задается схемой отношения, а семантическая выражается функциональными связями между атрибутами схемы. Группировка атрибутов должна быть рациональной и удовлетворять следующим требованиям:

- выбранные для отношения первичные ключи должны быть минимальными;
- выбранный состав отношений должен отличаться минимальной избыточностью атрибутов;

- между атрибутами не должно быть нежелательных функциональных зависимостей и они должны обеспечивать минимальное дублирование данных;
- не должно быть трудностей при выполнении операций включения, удаления и модификации (аномалии);
- перестройка набора отношений при введении новых типов должна быть минимальной.

Пример.

Отношение: Поставка (Название фирмы, Адрес, Товар, Кол-во, Цена)

Избыточность: кортежи отношения многократно дублируют название и адрес фирмы, если она поставляет несколько видов товара, а тем более плохо, если имеется несколько поставок одного вида товара.

Аномалии модификации: вследствие избыточности при обновлении необходимо просматривать все отношение для нахождения и изменения всех подходящих строк; изменение адреса фирмы, выполненное не во всех кортежах, относящихся к некоторой конкретной фирме, ведет к нарушению целостности.

Аномалии удаления: удаление всех кортежей с поставками от некоторого поставщика приведет к потере адреса и других реквизитов фирмы.

Аномалии включения: предположим, что заключен договор, но еще нет поставок от некоторой фирмы: следует ли включать кортежи с пустым (NULL) значением количества? А не забудем ли мы впоследствии удалить строку с неопределенным значением?

Таким образом, основная цель логического проектирования базы данных - сокращение избыточности хранимых данных и устранение возможных потенциальных аномалий работы с базами данных.

Для удовлетворения вышеотмеченных требований Э.Коддом предложен аппарат нормализации отношений.

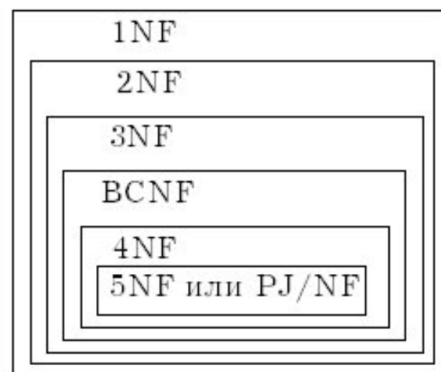
Нормализация отношений - это пошаговый обратимый процесс композиции или декомпозиции исходных отношений в отношения, обладающие лучшими свойствами при включении, изменении и удалении данных, назначение им ключей по определенным правилам нормализации и выявление всех возможных функциональных зависимостей.

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- первая нормальная форма (1NF);
- вторая нормальная форма (2NF);
- третья нормальная форма (3NF);
- нормальная форма Бойса-Кодда (BCNF);
- четвертая нормальная форма (4NF);
- пятая нормальная форма, или нормальная форма проекции-соединения (5NF или PJ/NF).

Основные свойства нормальных форм:

- каждая следующая нормальная форма в некотором смысле является более ограниченной, но более лучшей, чем предыдущая;
- при переходе к следующей нормальной форме положительные свойства предыдущих нормальных свойств сохраняются.



Нормальные формы

Нормальные формы отношений основываются на фундаментальных в теории реляционных баз данных понятиях функциональной и многозначной зависимости.

Определение 1. Функциональная зависимость.

В отношении R атрибут Y функционально зависит от атрибута X (X и Y могут быть составными) в том и только в том случае, если каждому значению X соответствует в точности одно значение Y: $R.X \rightarrow R.Y$.

Табельный номер -> Фамилия; Должность -> Зарплата.

Определение 2. Полная функциональная зависимость.

Функциональная зависимость $R.X \rightarrow$ называется полной, если атрибут Y не зависит функционально от любого точного подмножества X.

Определение 3. Неключевой атрибут.

Неключевым атрибутом называется любой атрибут отношения, не входящий в состав ключа (в частности, первичного).

Определение 4. Функционально полная и частичная зависимость неключевого атрибута от составного ключа.

Неключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части ключа, в противном случае имеет место частичная зависимость.

Отношение:

Чтение лекций (Таб_номер, Название_курса, Кол-во_часов)

Название_курса -> Кол-во_часов

Зависимость неключевого атрибута Кол-во_часов от части составного ключа говорит о частичной зависимости.

Определение 5. Транзитивная функциональная зависимость.

Функциональная зависимость $R.X \rightarrow R.Y$ называется транзитивной, если существует такой атрибут Z, что имеются функциональные зависимости $R.X \rightarrow R.Z$ и $R.Z \rightarrow R.Y$ и отсутствует функциональная зависимость $R.Z \rightarrow R.X$.

Фамилия -> Офис -> Телефон.

Определение 6. Взаимно независимые атрибуты.

Два или более атрибута взаимно независимы, если ни один из этих атрибутов не является функционально зависимым от других.

В отношении Чтение лекций: Кол-во_часов -> Таб_номер;

Таб_номер -> Кол-во_часов.

Определение 7. Отношение находится в 1NF тогда и только тогда, когда все входящие в него атрибуты являются атомарными (неделимыми).

Отношение R1

Таб_номер	ФИО	Оклад	Офис	Телефон	Д е т и	
					Имя	Возраст
211	Иванов	200	12	6-16	С	10
					Ж	7
					В	3
358	Петров	300	12	6-16	Т	5
360	Сидоров	250	5	3-06	Ж	8
					В	6

Отношение R2 (1NF)

Таб_номер	Имя ребенка	Возраст ребенка	ФИО	Оклад	Офис	Телефон
211	С	10	Иванов	200	12	6-16
211	Ж	7	Иванов	200	12	6-16
211	В	3	Иванов	200	12	6-16
358	Т	5	Петров	300	12	6-16
360	Ж	8	Сидоров	250	5	3-06
360	В	6	Сидоров	250	5	3-06

Первичный ключ: Таб_номер, Имя ребенка;

Функциональные зависимости:

$\text{Таб_номер} \rightarrow \text{ФИО};$
 $\text{Таб_номер} \rightarrow \text{Оклад};$
 $\text{Таб_номер} \rightarrow \text{Офис};$
 $\text{Офис} \rightarrow \text{Телефон};$
 $\text{Таб_номер, Имя_ребенка} \rightarrow \text{Возраст_ребенка}.$

Атрибуты ФИО, Оклад, Офис не находятся в полной функциональной зависимости от ключа, поскольку функционально зависят от части ключа (Таб_номер). Следствием это является:

- дублирование информации;
- нет возможности занести кортеж с сотрудником без детей (ключ не может содержать неопределенного значения);
- при удалении кортежа теряем не только информацию о ребенке сотрудника, но, возможно, о месте работы сотрудника, телефоне офиса и т.д.);
- при переводе сотрудника в другой офис вынуждены модифицировать все кортежи, описывающие этого сотрудника, иначе получим несогласованный результат.

Определение 8. Отношение находится в 2NF, если оно находится в 1NF и каждый неключевой атрибут функционально полно зависит от первичного ключа (см. определение 4).

Для приведения отношения во 2NF необходимо:

1. построить его проекцию, исключив атрибуты, которые не находятся в полной функциональной зависимости от составного ключа;

2. построить дополнительно одну или несколько проекций на часть составного ключа и атрибуты, функционально зависящие от этой части.

Отношение
R3

Таб_номер	Имя ребенка	Возраст ребенка
211	С	10
211	Ж	7
211	В	3
358	Т	5
360	Ж	8
360	В	6

Отношение
R4

Таб_номер	ФИО	Оклад	Офис	Телефон
211	Иванов	200	12	6-16
358	Петров	300	12	6-16
360	Сидоров	250	5	3-06

Отношения R3 и R4 во 2NF.
Отмеченные выше аномалии устранены

Если допустить наличие нескольких ключей, то определение 8 примет вид:

Определение 8~. Отношение находится в 2NF, если оно находится в 1NF и каждый неключевой атрибут функционально полно зависит от каждого ключа отношения.

Наличие транзитивной зависимости

Таб_номер → Офис;

Офис → Телефон

порождает аномалии следующего характера:

- имеет место дублирование информации о телефоне для сотрудников одного офиса;
- существует проблема избыточности, поскольку изменение телефона офиса влечет за собой необходимость поиска и изменения номеров всех сотрудников этого офиса;
- нельзя включить информацию о новом офисе, если в данный момент отсутствуют сотрудники этого офиса.

Определение 9 (в предположении существования единственного ключа). Отношение находится в 3NF в том и только в том случае, если оно находится в 2NF и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Для исключения транзитивной зависимости нужно произвести декомпозицию отношения R4 в два отношения R5 и R6.

Отношение
R5

Таб_номер	ФИО	Оклад	Офис
211	Иванов	200	12
358	Петров	300	12
360	Сидоров	250	5

Отношение
R6

Офис	Телефон
12	6-16
5	3-06

В результате преобразований имеем три отношения в 3NF, свободные от отмеченных аномалий:

Если отказаться от того ограничения, что отношение обладает единственным ключом, то определение 3NF примет следующую форму:

Определение 9~. Отношение находится в 3NF в том и только в том случае, если оно находится во 2NF, и каждый неключевой атрибут не является транзитивно зависимым от какого-либо ключа отношения.

На практике третья нормальная форма схем отношений достаточна в большинстве случаев, и приведением к третьей нормальной форме процесс проектирования реляционной базы данных обычно заканчивается.

При отсутствии многозначной зависимости, но наличии других зависимостей атрибутов, кроме зависимости от ключа, 3NF не гарантирует отсутствия аномалий операций включения, обновления и удаления.

В этом случае применяют усиленную 3NF Бойса-Кодда (BCNF).

Рассмотрим пример отношения:

Курсовой проект (Преподаватель, Предмет, Тема, Студент)

Курсовые проекты ведут несколько преподавателей по различным дисциплинам и каждый студент закреплен за одним из них. Студент выполняет только один проект, а одну и ту же тему проекта могут выполнять несколько студентов, но у разных преподавателей.

Преподаватель	Предмет	Тема	Студент
Иванов	экономика	Тема_1	Смирнов
Зайцев	физика	Тема_2	Федоров
Хабаров	математика	Тема_2	Антонов
Иванов	экономика	Тема_2	Егоров
Григорьев	статистика	Тема_1	Круглов
Григорьев	статистика	Тема_3	Фомин

Возможные ключи: *Преподаватель, Тема;*
Предмет, Тема.

Функциональные зависимости:

Преподаватель, Тема → Студент (зависимость от ключа);
Предмет, Тема → Студент (зависимость от ключа);
Студент → Тема.

Приведенное отношение находится в 3NF, так как в нем отсутствуют частичные и транзитивные зависимости неключевых атрибутов от ключа. Однако *имеется зависимость части составного ключа Тема от неключевого атрибута Студент, что порождает следующие аномалии:*

1. существует проблема контроля непротиворечивости данных, так как изменение преподавателя по дисциплине требует просмотра всего отношения с целью поиска и изменения кортежей, содержащих данные о преподавателе этой дисциплины;

2. данные о студенте и его проекте не могут быть занесены в БД до тех пор, пока не назначен руководитель проекта; и наоборот, если необходимо удалить преподавателя, то будут удалены данные о руководимом им студенте.

Устранение этих аномалий достигается устранением функциональной зависимости части составного ключа от неключевого атрибута (Студент -> Тема).

Определение 10. Детерминант.

Детерминант – любой атрибут, от которого полностью функционально зависит некоторый другой атрибут.

Определение 11.

Отношение находится в нормальной форме Бойса-Кодда (BCNF), если оно находится в 3NF и каждый детерминант является возможным ключом.

Можно дать и другое определение BCNF.

Определение 11~.

Отношение находится в нормальной форме Бойса-Кодда (BCNF), если оно находится в 3NF и в нем отсутствуют зависимости ключей или их частей от неключевых атрибутов.

Очевидно, что это требование не выполнено для отношения Курсовой проект.

Можно произвести его декомпозицию к двум отношениям Руководство(Преподаватель, Предмет) и Выполнение (Студент, Предмет, Тема).

Соединение полученных отношений Руководство и Выполнение по атрибуту Предмет дает исходное отношение Курсовой проект.

Отношение	Первичный ключ: <u>Преподаватель</u>
Руководство	Функциональные зависимости: $\text{Преподаватель} \rightarrow \text{Предмет}$
Отношение	Первичный ключ: <u>Студент</u>
Выполнение	Функциональные зависимости: $\text{Студент} \rightarrow \text{Предмет}$ $\text{Студент} \rightarrow \text{Тема}$

Если в отношении присутствуют многозначные зависимости, то устранение возможных аномалий выполняется приведением к 4NF.

Рассмотрим пример следующей схемы отношения:

Проекты (Проект_номер, Таб_номер, Проект_задание)

Отношение Проекты содержит номера проектов, для каждого проекта список сотрудников, которые могут выполнять проект, и список заданий, предусматриваемых проектом. Сотрудники могут участвовать в нескольких проектах, и разные проекты могут включать одинаковые задания.

Каждый кортеж отношения связывает некоторый проект с сотрудником, который может участвовать в этом проекте, и заданием, которое сотрудник выполняет в рамках данного проекта. Предполагается, что любой сотрудник, участвующий в проекте, выполняет все задания, предусмотренные проектом.

Проект_номер (A)	Таб_номер (B)	Проект_задание (C)
П1	C1	1
П1	C1	2
П1	C2	1
П1	C2	2
П2	C1	1
П2	C1	3
П2	C1	4
П2	C3	1
П2	C3	3
П2	C3	4

По причине сформулированных выше условий единственным возможным ключом отношения является составной атрибут Проект_Номер, Таб_номер, Проект_задание и нет никаких других детерминантов. Следовательно, отношение Проекты находится в BCNF.

Но при этом оно обладает недостатками: если, например, некоторый сотрудник присоединяется к данному проекту, необходимо вставить в отношение Проекты столько кортежей, сколько заданий в нем сотрудник будет выполнять. Аналогичная ситуация возникает при появлении нового проекта.

Определение 12. Многозначные зависимости.

В отношении $R(A, B, C)$ существует многозначная зависимость $R.A \twoheadrightarrow R.B$ в том и только в том случае, если множество значений B , соответствующее паре значений A и C , зависит только от A и не зависит от C (то есть если для каждого значения атрибута $R.A$ существует хорошо определенное множество соответствующих значения атрибута $R.B$).

В отношении *Проекты* существуют следующие две многозначные зависимости:

Проект_номер \twoheadrightarrow *Таб_номер*;

Проект_номер \twoheadrightarrow *Проект_задание*

Легко показать, что в общем случае в отношении $R(A, B, C)$ существует многозначная зависимость $R.A \twoheadrightarrow R.B$ в том и только в том случае, когда существует многозначная зависимость $R.A \twoheadrightarrow R.C$.

Определение 13. Полная декомпозиция и проецирование без потерь.

Полной декомпозицией отношения называют такую совокупность произвольного числа ее проекций, соединение которых полностью совпадает с исходным отношением. Под проецированием без потерь понимается такой способ декомпозиции отношения, при котором исходное отношение полностью и без избыточности восстанавливается путем естественного соединения полученных отношений.

Теорема Фейджина.

Отношение $R(A, B, C)$ можно спроецировать без потерь в отношения $R_1(A, B)$ и $R_2(A, C)$ в том и только в том случае, когда многозначные зависимости $A \twoheadrightarrow B$ и $A \twoheadrightarrow C$ могут быть разнесены в отношения R_1 и R_2 .

Определение 14. Отношение находится в 4NF, если оно находится в BCNF и в нем отсутствуют многозначные зависимости, последнее возможно тогда и только тогда, когда полная декомпозиция двух проекций, в которые разнесены многозначные зависимости, содержат возможный ключ.

В нашем примере можно произвести декомпозицию отношения *Проекты* в два отношения *Проекты-Сотрудники* и *ПроектыЗадания*:

Отношение
Проекты-Сотрудники
(Проект_номер, Таб_номер)

Проект_номер (A)	Таб_номер (B)
П1	С1
П1	С2
П2	С1
П2	С3

Отношение
Проекты-Задания
(Проект_номер, Таб_номер)

Проект_номер (A)	Проект_задание (C)
П1	1
П1	2
П2	1
П2	3
П2	4

Оба эти отношения находятся в 4NF и свободны от отмеченных аномалий. Соединение отношений Проекты-Сотрудники и ПроектыЗадания дает отношение Проекты.

Однако не всегда декомпозиция схем отношений гарантирует обратимость. Отношение может быть восстановлено без потерь соединением его проекций, если оно удовлетворяет зависимости по соединению.

Определение 15. Зависимость соединения.

Отношение $R(X, Y, \dots, Z)$ удовлетворяет зависимости соединения $*(X, Y, \dots, Z)$ в том и только в том случае, когда R восстанавливается без потерь путем соединения своих проекций на X, Y, \dots, Z .

В качестве примера рассмотрим отношение Сотрудники-Отделы-Проекты (Таб_номер, Отд_номер, Проект_номер)

Предположим, что один и тот же сотрудник может работать в нескольких отделах и работать в каждом отделе над несколькими проектами. Первичным ключом этого отношения является полная совокупность его атрибутов, отсутствуют функциональные и многозначные зависимости.

Поэтому отношение находится в 4NF. Однако в нем могут существовать аномалии, которые можно устранить путем декомпозиции в три отношения.

Определение 16. Отношение находится в нормальной форме проекции соединения PJ/NF в том и только в том случае, когда в каждой ее полной декомпозиции все проекции содержат возможный ключ.

Введем следующие имена составных атрибутов:

СО = {Таб_номер, Отд_номер}; СП = {Таб_номер, Проект_номер};

ОП = {Отд_номер, Проект_номер}.

Предположим, что в отношении Сотрудники-Отделы-Проекты существует зависимость соединения:

$*(СО, СП, ОП)$

Возможные аномалии при работе с отношением Сотрудники-ОтделыПроекты можно устранить путем декомпозиции исходного отношения в три новых отношения:

Сотрудники-Отделы (Таб_номер, Отд_номер);

Сотрудники- Проекты (Таб_номер, Проект_номер);

Отделы-Проекты (Отд_номер, Проект_номер).

Пятая нормальная форма - это последняя нормальная форма, которую можно получить путем декомпозиции. Ее условия достаточно нетривиальны, и на практике 5NF не используется.

Заметим, что зависимость соединения является обобщением как многозначной, так и функциональной зависимостей.

Об ограничениях целостности

Целостность (от англ. integrity - нетронутость, неприкосновенность, сохранность, целостность) понимается как правильность данных в любой момент времени. Поддержание целостности базы данных может рассматриваться как защита данных от неверных изменений или разрушений.

Выделяют три группы правил целостности:

- **Целостность по сущностям.**
- **Целостность по ссылкам.**
- **Целостность, определяемая пользователем.**

Общая мотивировка первых двух правил целостности общих для любых реляционных баз данных, состоит в следующем:

1. Не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе, принимал неопределенное значение.

2. Для каждого внешнего ключа в проекте проектировщик базы данных должен специфицировать не только атрибут или комбинацию атрибутов, составляющих этот внешний ключ, и целевое отношение, которое идентифицируется этим ключом, но также и ответы на три вопроса (три ограничения, которые относятся к этому внешнему ключу).

Вопрос 1. Может ли внешний ключ принимать неопределенное значение (NULL-значение)?

Значение внешнего ключа должно:

- либо быть равным значению первичного ключа отношения, с которым он связан;
- либо быть полностью неопределенным, при этом каждое значение атрибута, участвующего во внешнем ключе, должно быть неопределенным.

Например, в отношении Поставка, очевидно, поставка, осуществляемая неизвестным поставщиком, или поставка неизвестного продукта, не может иметь NULL-значения, в то время как атрибут Отдел в отношении Сотрудник может иметь NULL-значение, если сотрудник пока еще не зачислен ни в какой отдел.

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

Существует три возможности:

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

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

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

3. Для любой конкретной базы данных существует ряд дополнительных специфических правил, которые относятся к ней одной и определяются разработчиком. Чаще всего контролируется:

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

Получение реляционной схемы из ER-модели

Шаг 1. Каждая простая сущность превращается в отношение. Простая сущность - сущность, не являющаяся подтипом и не имеющая подтипов. Имя сущности становится именем отношения.

Шаг 2. Каждый атрибут становится возможным столбцом с тем же именем; может выбираться более точный формат. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, - не могут.

Шаг 3. Компоненты уникального идентификатора сущности превращаются в первичный ключ отношения. Если имеется несколько возможных уникальных идентификаторов, выбирается наиболее используемый.

Шаг 4. Связи "многие к одному" (и "один к одному") становятся внешними ключами. Для этого делается копия уникального идентификатора с конца связи "один", и соответствующие столбцы составляют внешний ключ. Необязательные связи соответствуют столбцам, допускающим неопределенные значения; обязательные связи - столбцам, не допускающим неопределенные значения.

Шаг 5. В таблицах, построенных на основе ассоциаций, внешние ключи используются для идентификации участников ассоциации, а в таблицах, построенных на основе характеристик и обозначений, внешние ключи используются для идентификации сущностей, описываемых этими характеристиками и обозначениями.

Специфицировать ограничения, связанные с каждым из этих внешних ключей.

Шаг 6. Если в концептуальной схеме присутствовали подтипы, то возможны два способа:

- все подтипы размещаются в одной таблице (а);
- для каждого подтипа строится отдельная таблица (б).

При применении способа (а) таблица создается для наиболее внешнего супертипа. В таблицу добавляется по крайней мере один столбец, содержащий код ТИПА и он становится частью первичного ключа. Для работы с подтипами могут создаваться представления.

При использовании метода (б) супертип воссоздается с помощью конструкции UNION.

Все в одной таблице	Таблица - на подтип
Преимущества	
Все хранится вместе	Более ясны правила подтипов
Легкий доступ к супертипу и подтипам	Программы работают только с нужными таблицами
Требуется меньше таблиц	
Недостатки	
Слишком общее решение	Слишком много таблиц
Требуется дополнительная логика работы с разными наборами столбцов и разными ограничениями	Смущающие столбцы в представлении UNION
Потенциальное узкое место (в связи с блокировками)	Потенциальная потеря производительности при работе через UNION
Для хранения неопределенных значений требуется дополнительная память	Над супертипом невозможны модификации

Шаг 7. Выполнить шаги по нормализации полученных отношений, приведя их к желаемой нормальной форме.

Шаг 8. Указать ограничения целостности проектируемой базы данных и дать (если это необходимо) краткое описание полученных таблиц и их полей.

Шаг 9. Создать индексы для первичного ключа (уникальный индекс), внешних ключей и тех атрибутов, на которых предполагается в основном базировать запросы и выполнять соединения.