

Тема №5 Нормальные формы отношений

1. Понятие о логической модели реляционной базы данных
2. Нормализация отношений
3. Первая нормальная форма
4. Вторая нормальная форма
5. Третья нормальная форма
6. Нормальная форма Бойса-Кодда
7. Четвертая нормальная форма
8. Пятая нормальная форма

Понятие о логической модели реляционной базы данных

Определение 1: Под реляционной базой данных принято понимать совокупность экземпляров конечных отношений. Совокупность схем отношений образует схему реляционной базы данных.

Схема реляционной базы данных является *логической моделью реляционной базы данных*.

Понятие о логической модели реляционной базы данных

На основе *информационной модели* в процессе проектирования создаются *логическая* и *физическая модели* данных.

Информационная модель данных отражает потребности системы в данных и связи между данными с точки зрения их потребителей – пользователей.

Логическая модель данных является независимым логическим представлением данных.

Физическая модель данных содержит определения всех реализуемых объектов в конкретной базе данных для конкретной СУБД.

Понятие о логической модели реляционной базы данных

На практике часто рассматривают только две модели - логическую и физическую модели данных.

При этом информационная и логическая модели данных не различаются и считаются синонимами.

В рамках такого подхода некоторые специалисты в области баз данных считают, что информационная модель данных должна быть *нормализована*.

Это означает, что проектировщики баз данных должны требовать от аналитиков, чтобы они приводили информационную модель данных к *третьей нормальной форме!*

Понятие о логической модели реляционной базы данных

Возможны следующие проблемы:

1. Аналитики, являясь экспертами в предметной области, как правило, не представляют, что такое нормализация данных.
2. Информационная модель данных должна быть независимой от физической модели данных, в рамках которой она будет реализовываться.
3. Проектировщики базы данных должны иметь логическое представление данных, посредством которого, с одной стороны, общаться с аналитиками и пользователями в понятных для них терминах, и, с другой стороны, превращать полученные логические отношения в физические объекты базы данных.

Понятие о логической модели реляционной базы данных

При построении *логической модели реляционной базы данных* особое значение для решения задачи формирования отношений базы данных имеет понятие функциональной зависимости (ФЗ).

Установление ФЗ и получение наилучшего с точки зрения минимальности представления множества ФЗ позволят построить наиболее оптимальный вариант базы данных, обеспечивающий надежность хранения и обработки данных на основе методов эквивалентных преобразований схем отношений *реляционной базы данных*.

Процесс решения такой задачи называется *нормализацией отношений* информационной модели предметной области.

Нормализация отношений

Нормализация отношений информационной модели предметной области является механизмом создания логической модели реляционной базы данных.

С математической точки зрения задача построения как информационной модели предметной области, так и логической модели реляционной базы данных является результатом решения следующих комбинаторных задач:

- группировка атрибутов в отношении предметной области;
- распределение атрибутов по отношениям базы данных.

Нормализация отношений

Перечень наиболее важных требований при нормализации:

- Первичные ключи отношений должны быть минимальными (требование минимальности первичных ключей).
- Число отношений базы данных должно по возможности давать наименьшую избыточность данных (требование надежности данных).
- Число отношений базы данных не должно приводить к потере производительности системы (требование производительности системы).

Нормализация отношений

- Данные не должны быть противоречивыми, т.е. при выполнении операций включения, удаления и обновления данных их потенциальная противоречивость должна быть сведена к минимуму (требования непротиворечивости данных).
- Схема отношений базы данных должна быть устойчивой, способной адаптироваться к изменениям при ее расширении дополнительными атрибутами (требование гибкости структуры базы данных).

Нормализация отношений

- Разброс времени реакции на различные запросы к базе данных не должен быть большим (требование производительности системы).
- Данные должны правильно отражать состояние предметной области базы данных в каждый конкретный момент времени (требование актуальности данных).

Многие из требований находятся в противоречии друг к другу: Требование производительности находится в противоречии к требованию гибкости. Требование минимизировать число отношений в базе данных находится в противоречии к требованию надежности данных.

Нормализация отношений

В процессе эксплуатации *реляционной базы данных* происходит модификация данных и данные в базе данных с течением времени становятся *несогласованными* и *противоречивыми*.

Потенциальная противоречивость (или аномалии) при выполнении операций манипулирования данными в базе данных зависит как от типа ФЗ между атрибутами, так и от группировки этих атрибутов по отношениям.

Нормализация отношений

Пример: Аномалии операций над базой данных

ПОСТАВКИ (Поставщик, Адрес, Товар, Количество, Стоимость)

Обновление. Если Поставщик поставляет несколько видов товара, то значение атрибута Адрес повторяется для каждого кортежа и, следовательно, при изменении адреса поставщика необходимо изменить значение этого атрибута во всех этих кортежах. Иначе будет нарушена целостность данных базы данных.

Нормализация отношений

Удаление. Если Поставщик прекращает поставку товаров на некоторое время, то кортежи со всеми его поставками удаляются. При этом происходит потеря реквизитов поставщика - Адреса и Наименования.

Вставка. Если договор на поставку уже заключен, а поставка еще не произведена, то невозможно включить в рассматриваемое отношение значения атрибутов Поставщик и Адрес, поскольку нет сведений о поставках (проблема интерпретации нуль-значений).

Нормализация отношений

Теория функциональных зависимостей позволяет установить определенные требования к схемам отношений в реляционной базе данных.

Эти требования формулируются в терминах свойств отношений и называются *нормальными формами схем отношений*.

Каждая нормальная форма отношений связана с определенным классом ФЗ, которые представлены в отношениях.

Нормализация отношений

Теория ФЗ утверждает, что существует минимальное покрытие множества ФЗ базы данных, т.е. минимальный набор ФЗ, из которых можно вывести все остальные.

Каждой ФЗ из минимального покрытия можно отвести по самостоятельному отношению.

Однако,

во-первых, для заданного множества ФЗ может существовать несколько минимальных покрытий (выбор среди возможных альтернатив),

во-вторых, для практики важно иметь наглядный способ построения минимального покрытия.

Нормализация отношений

Процесс устранения потенциальной противоречивости и избыточности данных в отношениях реляционной базы данных называется нормализацией исходных схем отношений.

Нормализация отношений заключается в выполнении *декомпозиции* или *синтеза отношений*, назначении *ключей отношений* в соответствии с определенными правилами, гарантирующими целостность отношений базы данных.

Требование минимальности числа отношений, т.е. построения минимального покрытия ФЗ обычно является *опциональным*.

На минимальном покрытии, как правило, не может быть достигнута высокая производительность обработки запросов.

Нормализация отношений

Нормализация схем отношений дает наилучший результат при моделировании предметной области с использованием реляционной модели данных; при этом не вводится большого числа ограничений, не искажаются данные.

Таким образом, нормализация отношений является методом удаления из отношения ФЗ, которые приводят к аномалиям модификации данных.

Нормализация отношений помогает проектировать реляционную базу данных, которая не содержит избыточных данных и гарантирует их целостность.

Первая нормальная форма

Отношение находится в первой нормальной форме (1НФ), если все атрибуты отношения являются простыми (требование атомарности атрибутов в реляционной модели), т.е. не имеют компонентов.

Домен атрибута должен состоять из неделимых значений и не может включать в себя множество значений из более элементарных доменов.

Так, например, дата не может считаться простым атрибутом.

Первая нормальная форма

Пример: Приведение отношения к 1НФ

Отношение SHIPMENT (ОТГРУЗКА) является ненормализованным. Оно содержит повторяющиеся группы, представляющие массив значений, 1st Consignment, 2st Consignment, 3st Consignment (партии грузов).

Shipment
<u>Ship Registration Number</u>
<u>Departure Data</u>
Ship Name
Origin
Destination
1st Consignment
2st Consignment
3st Consignment
Customs Value
Ship Capacity
Customs Declaration

Для такого отношения следует ввести бизнес-правило, требующее, чтобы груз состоял не более чем из трех партий!

Первая нормальная форма

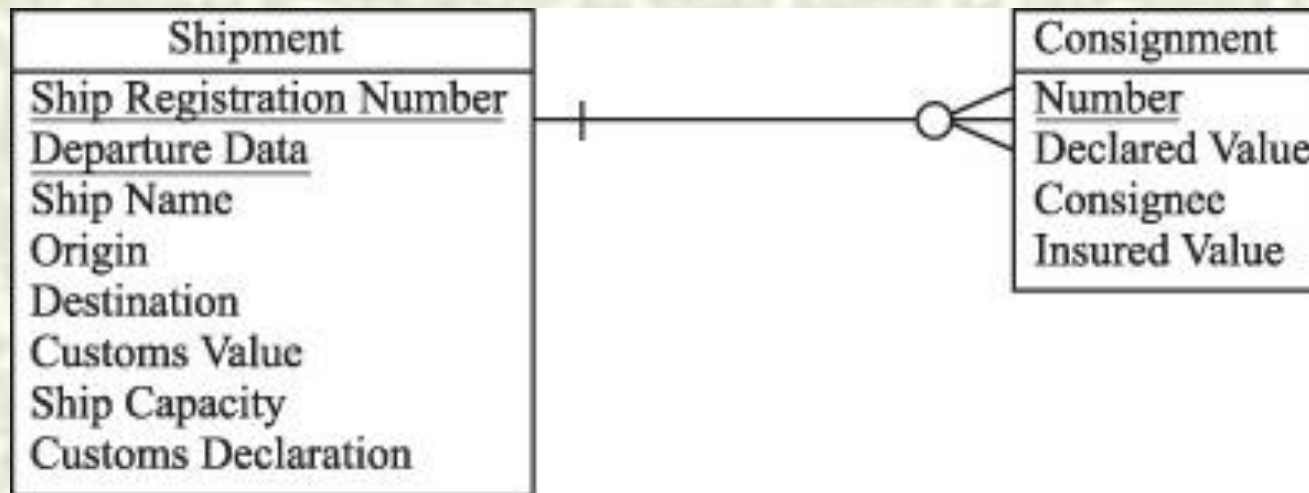
Использование отношения, представленного не в 1НФ, может породить следующие проблемы:

- если груз аннулируется и строка, связанная с грузом, удаляется из отношения, то вместе с ней удаляются все сведения о партиях груза на борту судна;
- если на склад прибывает новая партия груза, и она еще не включена в состав груза, подлежащего отправке, то сведения о партии заносить некуда;
- необходимо вводить ограничение: в грузе не может быть более трех партий.

Первая нормальная форма

Приведение отношения SHIPMENT к 1НФ заключается в изъятии данных о партиях груза из отношения SHIPMENT и создании для них связанного подчиненного отношения CONSIGNMENT (ПАРТИЯ_ГРУЗА).

Результат приведения отношения SHIPMENT к 1НФ :



Вторая нормальная форма

Атрибут отношения считается ключевым если он является элементом какого-либо ключа отношения.

В противном случае атрибут будет считаться *неключевым атрибутом*.

Так в отношении (Город, Адрес, Почтовый_индекс) все атрибуты являются ключевыми, поскольку при заданных ФЗ (*город, адрес*) → *почтовый_индекс* и *почтовый_индекс* → *город* ключами являются пары (*город, адрес*) и (*адрес, почтовый_индекс*).

Вторая нормальная форма

Отношение находится во второй нормальной форме (2НФ), если оно находится в 1НФ, и все неключевые атрибуты отношения функционально полно зависят от составного ключа отношения.

Иными словами, 2НФ требует, чтобы отношение не содержало частичных ФЗ.

Вторая нормальная форма

Пример: Приведение отношения ко 2НФ.

Отношение SHIPMENT содержит частичную ФЗ: неключевой атрибут Ship Capacity (грузоподъемность корабля) не зависит от ключевого атрибута Departure Date (даты убытия), а зависит от ключевого атрибута Ship Registration Number (регистрационный номер корабля).

Shipment
<u>Ship Registration Number</u>
<u>Departure Date</u>
Ship Name
Origin
Destination
1st Consignment
2st Consignment
3st Consignment
Customs Value
Ship Capacity
Customs Declaration

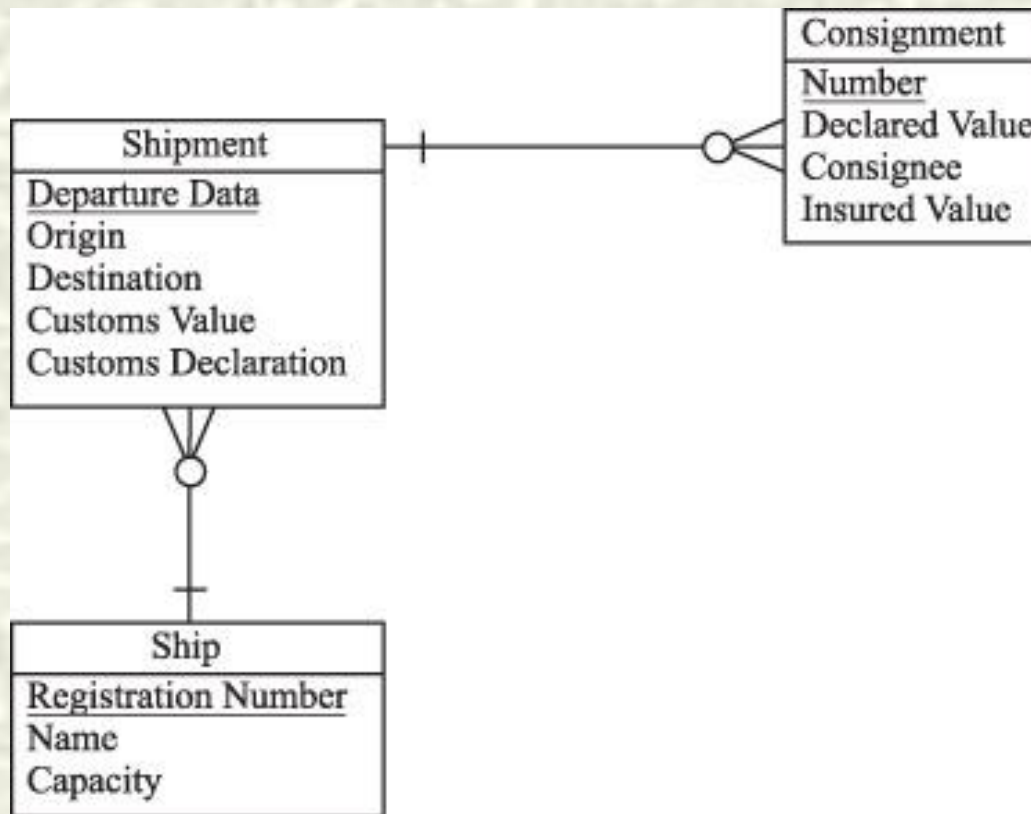
Вторая нормальная форма

Использование отношения, представленного не во 2НФ, может породить следующие проблемы:

- невозможно занести в базу данных название и грузоподъемность корабля, который не доставил еще ни одного груза, - можно только ввести для него фиктивный груз;
- если удалить кортеж из отношения Shipment после отправки груза, то потеряются все данные о кораблях, для которых в настоящее время нет груза;
- невозможно отразить факт переоборудования корабля и получения им новой грузоподъемности; если переписать все предыдущие кортежи об этом корабле, то получится, что он в прошлом плавал недогруженным или перегруженным.

Вторая нормальная форма

Приведение отношения SHIPMENT ко 2НФ заключается в изъятии атрибутов частичной ФЗ из отношения SHIPMENT и создании для нее связанного подчиненного отношения SHIP.



Третья нормальная форма

Отношение находится в третьей нормальной форме (3НФ), если оно находится во 2НФ, и все неключевые атрибуты отношения зависят только от первичного ключа.

Иными словами, 3НФ требует, чтобы отношение не содержало транзитивных ФЗ неключевых атрибутов от ключа.

Третья нормальная форма

Возможна следующая проблема:

Наличие транзитивной зависимости $X \rightarrow Y \rightarrow A$ не позволяет связать значения Y и X , если не существует значения A , связанного со значением Y . Это затрудняет вставку и обновление данных, которые необходимо выполнить сразу для пары связей, а в случае удаления данных приводит к потере связи.

Третья нормальная форма

Пример: Приведение отношения к 3НФ

Отношение SHIPMENT содержит транзитивную ФЗ: атрибут Customs Declaration (таможенная декларация) является по своей сути свойством атрибутов Origin (пункт отправления) и Destination (пункт назначения).

