

Вопросы и задания к параграфу 4.2

49. Как формулируется задача проектирования реляционной базы данных? Какие цели при этом преследуются?

Задача: создание реляционной схемы БД, включающей определения базовых отношений и ограничений целостности, отражающих семантику предметной области.

Цели:

1. Возможность хранения всех необходимых данных в БД
2. Исключение избыточного дублирования данных
3. Сведение числа хранимых в БД отношений к минимуму
4. Нормализация отношений для упрощения решения проблем, связанных с вставкой, обновлением и удалением данных.

(**Нормализация** - включает в себя создание таблиц и установку **отношений** между этими таблицами в соответствии с правилами, предназначенными для защиты **данных** и обеспечения большей гибкости **базы данных** за счет исключения избыточности (излишнего дублирования) и правильной зависимости)

50. Что такое универсальное отношение?

Универсальное отношение – это отношение, в которое включаются все представляющие интерес атрибуты ПрО и которое может таким образом содержать все данные, предполагающиеся к хранению в БД. Для малых БД (включающих не более 15-20 атрибутов) универсальное отношение может использоваться в качестве отправной точки при проектировании РБД.

51. Какие аномалии могут возникать при использовании некачественных отношений?

Аномалия вставки, удаления, обновления.

(В идеале эти команды должны выполняться единственной соответствующей командой INSERT, DELETE, UPDATE)

52. Укажите условие первой нормальной формы отношений.

Каждый элемент отношения имеет и всегда будет иметь атомарное значение.

Отношение должно быть в 1НФ даже прежде постановки вопроса о его разбиении на два или более отношения. По большому счету реляционная модель и теория имеют дело только с такими отношениями

53. Что такое декомпозиция отношения? Для чего она используется?

Процесс разбиения отношения с целью уменьшения вероятности возникновения аномалий.

Ключевой для осуществления декомпозиции логическим методическим путем является концепция функциональных зависимостей между атрибутами в рассматриваемом отношении (при анализе отношения с точки зрения повышенных нормальных форм в расчет берутся другие виды зависимостей в данных).

54. С помощью какой операции над отношениями она осуществляется?

С помощью операции проекции.

Декомпозиция осуществляется с помощью функциональной зависимости между атрибутами в рассматриваемом отношении.

55. Каких правил следует придерживаться при выборе ФЗ для очередной декомпозиции?

Простым правилом выбора ФЗ для декомпозиции может служить поиск цепочек ФЗ вида $A \rightarrow B \rightarrow C$ с последующим использованием крайней правой зависимости. (Читается: В функционально зависит от А)

Более **обобщенное правило** можно сформулировать следующим образом: всеми средствами следует избегать выбора ФЗ, зависящая часть которой сама – целиком или частично – является детерминантом другой ФЗ.

56. Какая операция является обратной декомпозиции?

Если оператором декомпозиции в процедуре нормализации является операция проекции, то обратной операцией служит **операция соединения**. Обратимость означает, что исходное отношение равно соединению его проекций, полученных в ходе декомпозиции.

57. Как в теории реляционных БД определяется функциональная зависимость (ФЗ)? Какое отображение стоит за этим понятием? Что является источником информации о ФЗ?

Функциональная зависимость определяется следующим образом: если даны два атрибута А и В, то говорят, что В функционально зависит (ФЗ) от А, если для каждого значения А в любой момент времени существует не более одного связанного с ним значения В. А и В могут быть составными. Говорят еще, что А функционально определяет В.

В понятии функциональная зависимость скрыто **функциональное отображение** между атрибутами А и В.

Единственный способ определения функциональных зависимостей для схемы отношения заключается в том, чтобы внимательно проанализировать семантику атрибутов. В этом смысле зависимости являются фактически **высказываниями о закономерностях реального мира**.

58. Что такое детерминант атрибута?

Если $A \rightarrow B$ (А функционально определяет В) и В не зависит функционально от любого подмножества А, то говорят, что А представляет собой детерминант В

59. Что такое возможный ключ отношения?

Минимальный набор атрибутов, таких что, если известны их значения, то можно однозначно определить и значения остальных атрибутов отношения

- Атрибут или группа атрибутов UNIQUE (уникальных)

(т.е. выполняются требования к первичному ключу - значение атрибута/ГА являются уникальными для каждой записи в таблице. Таких возможных ключей может быть несколько).

Определения:

1. Если атрибут какого-либо отношения (или группа атрибутов) функционально определяет другие атрибуты этого же отношения, то эти атрибуты являются возможным ключом.
2. Если атрибут какого-либо отношения (или группа атрибутов) функционально определяет отношение, этот атрибут является возможным ключом. [Другими словами, возможный ключ уникально идентифицирует кортежи отношения.]
3. Если атрибут отношения (или группа атрибутов) не имеет значений-дубликатов в кортежах отношения, этот атрибут (или группа) является возможным ключом.

60. Определите условие нормальной формы Бойса-Кодда (НФБК).

Отношение находится в нормальной форме Бойса-Кодда (НФБК), если и только если каждый детерминант атрибутов отношения является возможным ключом отношения.

61. Приведите первоначальный алгоритм нормализации отношений до НФБК.

Декомпозиционный алгоритм проектирования:

1. Разработка универсального отношения для БД и преобразование его в 1НФ. Оно пока является единственным в очереди еще не проверенных на НФБК отношений.
2. Определение всех ФЗ очередного отношения.
3. Определение того, находится ли очередное отношение в НФБК. Если да, то проектирование для него завершается, если нет – это отношение декомпозируется на два новых, которые помещаются в очередь еще не проверенных на НФБК отношений.
4. Повторение шагов 2 и 3 для каждого очередного отношения. Проектирование завершается, когда очередь еще не проверенных на НФБК отношений опустеет, а, значит, все полученные отношения находятся в НФБК.

62. Укажите желательные свойства декомпозиции. Дайте им определения.

Первое – это свойство соединения без потерь (англ. lossless-join), которое позволяет восстановить любой кортеж исходного отношения,

используя соответствующие кортежи меньших отношений, полученных в результате декомпозиции.

Второе – свойство сохранения зависимостей (англ. dependency preservation), которое позволяет сохранить все ограничения, наложенные на исходное 345 отношение, посредством наложения некоторых ограничений на каждое из меньших отношений, полученных после декомпозиции.

Определение 4.2.5.8. Декомпозиция ρ обладает **свойством соединимости без потерь информации** по отношению к множеству функциональных зависимостей F , если для каждой реализации (экстенционала) r отношения R , удовлетворяющей F ,

$$r = \bigstar_{i=1}^k \pi_{R_i}(r), \text{ где «}\bigstar\text{» – операция естественного соединения, а } R_i \text{ в данном случае}$$

представляет атрибуты отношения R_i . Свойство соединимости без потерь гарантирует

Определение 4.2.5.9. Декомпозиция называется **сохраняющей зависимости**, если зависимости R сохранены в новой схеме отношений ρ . Формально это свойство можно выразить следующим образом. Рассмотрим подмножество ФЗ F_i , содержащее все такие

ФЗ $X \rightarrow Y$, что $XY \subseteq R_i$. Декомпозиция ρ сохраняет множество ФЗ F , если $\left(\bigcup_{i=1}^k F_i \right)^+ = F^+$.

Операция «+» образует множество всех ФЗ, выводимых из исходного множества ФЗ (т.е. являющихся их следствиями).

63. В чем заключается метод синтеза? Приведите пример.

Метод синтеза: необходимо все ФЗ с одинаковыми детерминантами выделить в группы и для каждой группы построить свое отношение, куда включить все атрибуты всех ФЗ этой группы.

В последнем примере были две зависимости с различными детерминантами. Согласно методу синтеза каждой ФЗ следует выделить ее собственное отношение – $R_1(A, B)$ и $R_2(C, B)$.

3. Другим случаем возможной потери ФЗ в процессе проектирования является ситуация, в которой один атрибут зависит от двух различных детерминантов.

$R(\underline{A}, \underline{C}, B)$

$A \rightarrow B$

$C \rightarrow B$

Отношение R не находится в НФБК. Если выбрать одну из ФЗ для декомпозиции, вторая теряется.

Если выбрать $A \rightarrow B$, имеем:

$R1(\underline{A}, \underline{C})$ и $R2(\underline{A}, B) \Rightarrow C \rightarrow B$ утеряна

Если выбрать $C \rightarrow B$, имеем:

$R1(\underline{A}, \underline{C})$ и $R2(\underline{C}, B) \Rightarrow A \rightarrow B$ утеряна

Остается разбить R на $R1(\underline{A}, B)$ и $R2(\underline{C}, B)$. Они находятся в НФБК, и исходные ФЗ сохранены.

Метод синтеза: необходимо все ФЗ с одинаковыми детерминантами выделить в группы и для каждой группы построить свое отношение, куда включить все атрибуты всех ФЗ этой группы.

64. Что такое избыточная ФЗ?

Зависимость, не заключающая в себе такой информации, которая не могла бы быть получена на основе других зависимостей из числа использованных при проектировании БД.

65. Перечислите правила вывода ФЗ.

Правила вывода функциональных зависимостей:

A1 (РЕФЛЕКТИВНОСТЬ): $A, X \rightarrow X$

A2 (ПОПОЛНЕНИЕ): $A \rightarrow B \Rightarrow A, Z \rightarrow B, Z$ и $A, Z \rightarrow B$

A3 (ТРАНЗИТИВНОСТЬ): $A \rightarrow B$ и $B \rightarrow C \Rightarrow A \rightarrow C$

П4 (ОБЪЕДИНЕНИЕ): $A \rightarrow B$ и $A \rightarrow C \Rightarrow A \rightarrow B, C$

П5 (ДЕКОМПОЗИЦИЯ): $A \rightarrow B, C \Rightarrow A \rightarrow B$ и $A \rightarrow C$

П6 (ПСЕВДОТРАНЗИТИВНОСТЬ): $A \rightarrow B$ и $B, Z \rightarrow C \Rightarrow A, Z \rightarrow C$

Первые три правила – аксиомы Армстронга.

Правило 1 (рефлексивность) указывает, что множество атрибутов всегда функционально определяет любое из своих подмножеств.

Правило 2 (пополнение) указывает, что добавление одного и того же множества атрибутов и к левой, и к правой частям функциональной

зависимости (или только к левой) приводит к получению еще одной действительной зависимости.

Правило 3 (транзитивность) указывает, что функциональные зависимости являются транзитивными.

Правило 5 (декомпозиция) определяет, что можно удалять атрибуты из правой части зависимости. Повторное применение этого правила позволяет разложить функциональную зависимость $A \rightarrow B, C, D$ на ряд функциональных зависимостей $A \rightarrow B$, $A \rightarrow C$ и $A \rightarrow D$. 356

Правило 4 (объединение) указывает, что в процессе проектирования может быть выполнена обратная операция, при которой ряд зависимостей с одинаковыми левыми частями $A \rightarrow B$, $A \rightarrow C$ и $A \rightarrow D$ объединяется в одну функциональную зависимость $A \rightarrow B, C, D$.

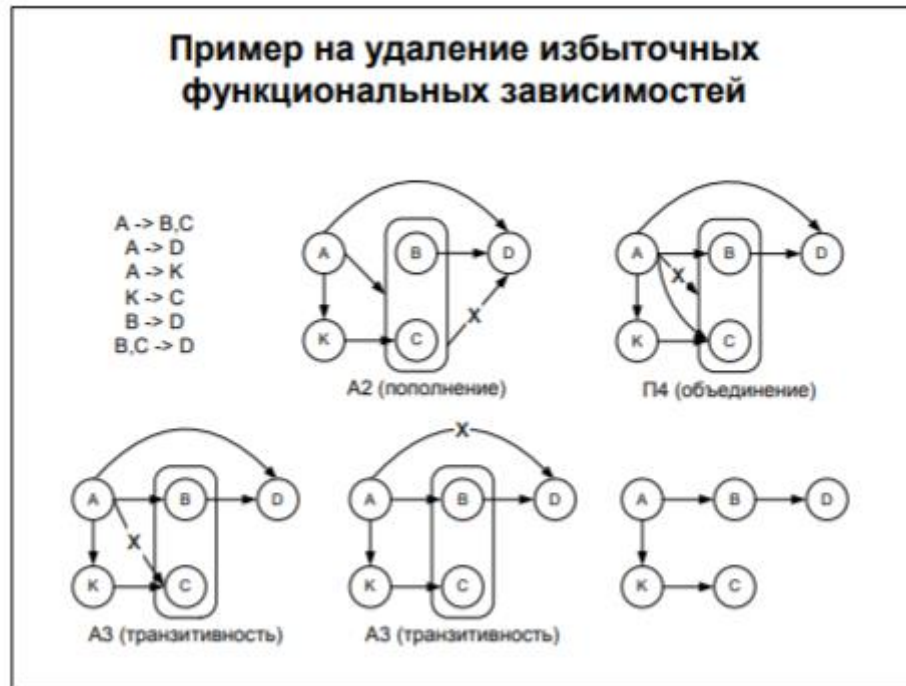
66. Какими свойствами обладают аксиомы Армстронга?

Надежность: если зависимость $X \rightarrow Y$ выведена из F с помощью этих аксиом, то она справедлива в любом отношении, в котором справедливы зависимости из F .

Полнота: всякий раз, когда зависимость $X \rightarrow Y$ не может быть выведена из F с помощью этих аксиом, она логически не следует из F .

67. Как определять избыточные ФЗ с использованием правил вывода ФЗ? Приведите пример.

Если ФЗ может быть выведена из других ФЗ с помощью правил вывода, она избыточна.



В качестве примера на удаление избыточных ФЗ рассмотрим набор ФЗ, показанный на слайде. Первый рисунок представляет собой графическую диаграмму этого исходного набора ФЗ.

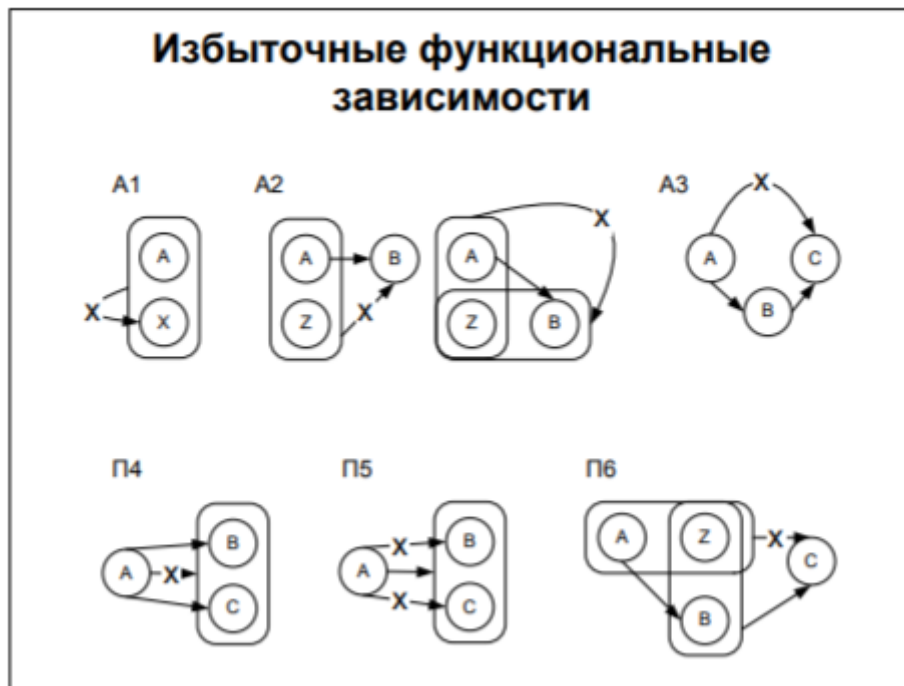
На первом шаге избавляемся от ФЗ $B, C \rightarrow D$ так как по аксиоме A2 она является следствием $B \rightarrow D$.

На следующем шаге заменяем ФЗ $A \rightarrow B, C$ на пару эквивалентных ей ФЗ $A \rightarrow B$ и $A \rightarrow C$ (эту эквивалентность обеспечивают правила П4 и П5).

Вторая из вновь образованных ФЗ ($A \rightarrow C$) удаляется, поскольку она является следствием ФЗ $A \rightarrow K$ и $K \rightarrow C$ (аксиома A3).

Благодаря применению этой же аксиомы, на последнем шаге удалена ФЗ $A \rightarrow D$.

В результате образовался следующий набор ФЗ, неизбыточно характеризующий все зависимости, представленные в исходном наборе: $A \rightarrow B$, $B \rightarrow D$, $A \rightarrow K$, $K \rightarrow C$.



68. Что такое минимальное покрытие ФЗ отношения?

Набор неизбыточных ФЗ, полученный путем удаления всех избыточных ФЗ из исходного набора с помощью 6 правил вывода, называется минимальным покрытием ФЗ отношения.

69. Как окончательно выглядит декомпозиционный алгоритм проектирования реляционных схем БД?

Модификация декомпозиционного алгоритма проектирования:

1. Построение универсального отношения. Оно пока является единственным в очереди еще не проверенных на НФБК отношений.
2. Определение всех ФЗ, существующих между атрибутами этого 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.
 - 4.1. Определение ФЗ минимального покрытия для очередного отношения.
 - 4.2. Определение того, находится ли очередное отношение в НФБК. Если да, то проектирование для него завершается, если нет –

это отношение декомпозируется на два новых, которые помещаются в очередь еще не проверенных на НФБК отношений.

4.3. Повторение шагов 4.1 и 4.2 для каждого очередного отношения. Проектирование завершается, когда очередь еще не проверенных на НФБК отношений опустеет, а, значит, все полученные отношения находятся в НФБК.

Если может быть построено более чем одно минимальное покрытие, осуществляется сравнение результатов, полученных на основе этих минимальных покрытий, с целью определения варианта, лучше других отвечающего требованиям ПрО.

70. Какие проверки отношений следует провести на завершающей фазе проектирования?

Проверка отношений на завершающей фазе проектирования:

1. Составляются списки ФЗ для каждого отношения. Эти списки ФЗ проверяются по двум направлениям:

1.1. Одна и та же ФЗ не должна появляться более чем в одном отношении.

1.2. Набор ФЗ, полученный в результате проектирования, должен в точности совпадать с набором, присутствующем в минимальном покрытии, полученном перед началом проектирования. В противном случае, необходимо показать возможность получения итогового набора ФЗ из минимального покрытия с помощью правил вывода и наоборот.

2. Осуществляется проверка на присутствие избыточных отношений. Если устанавливается избыточность отношения, его следует исключить из проектного набора. Отношение является избыточным, если:

2.1) все атрибуты этого отношения могут быть найдены в одном другом отношении проектного набора;

2.2) все атрибуты этого отношения могут быть найдены в отношении, которое может быть получено из других отношений проектного набора с помощью серии операций соединения над этими отношениями.

3. Отношения рассматриваются с практической точки зрения. Изучается характер использования отношений в конструируемой БД и

определяется, будут ли они обеспечивать те типы запросов и операций обновления, которые предполагается выполнять.

Если хотя бы одна из этих проверок окажется недостоверной, следует проанализировать процесс проектирования для выявления ошибок и/или рассмотреть другие варианты проектирования.

Вопросы и задания к параграфу 5.1

71. Каковы основные недостатки классической методики проектирования реляционных БД?

- нетрадиционный для людей способ восприятия и формализации ПрО
- практическая неприменимость для анализа сложных ПрО
- неоднозначность решения проблемы проектирования, граничащая с прямым перебором многочисленных вариантов схемы в поисках наиболее подходящего

72. Как выглядит основная схема любой семантической методики проектирования БД?

- проектирование семантической схемы ПрО с использованием той или иной модели
- перевод схемы в реляционную модель с применением подходящего набора правил трансформации и получение множества предварительных отношений
- проверка полученных отношений на удовлетворение требований нормальных форм и дальнейшая нормализация методом декомпозиции

73. Укажите этапы расширенной семантической методики проектирования БД?

- функциональное моделирование бизнес-процессов предприятия, для информационного обеспечения которых создается БД (деловая модель)
- семантическое моделирование данных в рамках той или иной семантической модели (ER-модель)
- получение логической схемы БД в понятиях логической модели выбранной СУБД (реляционная модель)
- настройка БД на языке физической модели СУБД

74. Какие доводы можно привести в пользу необходимости внесения изменений синхронно в артефакты всех этапов проектирования, включая самые ранние?

Внесение синхронных изменений во все артефакты, включая самые ранние, необходимо для обеспечения целостности восприятия проектных документов и для использования интегрированных CASE-инструментов с автоматическими преобразователями представления и генераторами логической и физической схем БД и кода приложения.

75. Каковы цели этапа анализа потребностей задач ПрО? Каким образом они достигаются?

- Изучение области деятельности моделируемой организации. (достигается определением целей организации и стратегии их достижения)
- Определение информационных потребностей организации; при этом единственное существенное ограничение, которое должно учитываться - адекватность процессам функционирования организации. (сбор метаданных и формирование метаописаний процессов функционирования организаций)
- Обеспечение представления указанных потребностей с помощью техники формального моделирования. (использование формальной модели)

76. Для чего предназначены различные функциональные модели ПрО?

SADT-модель (structured analysis and design technique) - моделирование сложных организационных бизнес-процессов и их реинжиниринг.

DFD-модель (data flow diagrams) - моделирование функционирования проектируемых информационных систем.

Деловая модель - простейшее средство функционального моделирования, достаточное для задачи проектирования схемы БД.

77. В каких понятиях описывается функционирование организации в деловой модели? Что стоит за этими понятиями?

Функции (бизнес-процессы) организации и классы данных, используемые для обеспечения выполнения этих функций.

Бизнес-процесс - совокупность различных видов деятельности компании, в рамках которой “на входе” используются один или более видов ресурсов, и в результате этой деятельности “на выходе” создается продукт, представляющий ценность для потребителя.

78. Каковы основные принципы построения деловой модели?

Определяются функции и классы данных, затем производится проверка непротиворечивости, избыточности и понятности определений.

Сама деловая модель представляет собой матрицу, строки которой соответствуют функциям, а столбцы – классам данных. На пересечении столбца и строки ставится метка, означающая, что этот класс данных участвует в выполнении соответствующей функции.

79. Как в дальнейшем будет использоваться деловая модель ПрО на последующих этапах семантической методики?

При информационном семантическом моделировании функции будут определять независимые этапы моделирования, а классы данных, упомянутые в строке, – границы рассмотрения на этих этапах. На последнем этапе семантического моделирования независимые информационные модели для каждой отдельной функции будут интегрированы в общую модель всей системы.

Вопросы и задания к параграфу 5.2

80. Какова главная стратегия процесса семантического моделирования с использованием деловой модели, как исходного артефакта, и ER-модели, как целевого формализма для представления семантической схемы БД?

Сведение глобальной задачи к подзадачам, решение этих подзадач и синтез решения глобальной задачи из решений подзадач

81. Какие этапы выделяются в этом процессе? Какие задачи решаются в ходе этих этапов?

1. Построение подсхем для каждой функции в отдельности:

- А. Определение множеств сущностей
- Б. Определение множеств связей
- В. Определение ограничений целостности

2. Интеграция подсхем в общую ER-схему предметной области

Формируется ряд взглядов на схему (подсхем) в зависимости от сфер деятельности. Взгляды затем интегрируются в описание ПрО, соответствующее схеме в целом.

82. На какие вопросы необходимо дать ответ при определении множеств сущностей?

1. Каким множествам сущностей соответствует каждый класс данных?
2. Каковы значение (семантика) и имя каждого множества сущностей?
3. Какие атрибуты каждого множества сущностей представляют интерес?
4. Каковы значение (семантика) и имя каждого атрибута?

83. На какие вопросы необходимо дать ответ при определении множеств связей?

1. Какие взаимосвязи (множества связей) между множествами сущностей ассоциируются с каждой функцией?
2. Каковы значение (семантика), имя и степень каждого множества связей?
3. Существуют ли у множеств связей собственные атрибуты, представляющие интерес с точки зрения функции? Если да, то каковы их семантика и имена?

84. На какие вопросы необходимо дать ответ при определении ограничений целостности?

1. Какова область значений каждого атрибута? Есть ли среди них многозначные?
2. Каковы известные функциональные зависимости между атрибутами каждого множества сущностей и множества связей?
3. Каковы ключи (если таковые существуют) каждого множества сущностей? Как можно еще идентифицировать сущности и связи каждого типа?
4. Какой тип отношений соответствует каждому бинарному множеству связей?
5. Какие типы отображений соответствуют каждому множеству связей?
6. Какие ограничения, выражаемые в логике предикатов первого порядка, накладываются на данные?

85. По каким правилам осуществляется интеграция подсхем в общую ER-схему ПрО?

Интеграция множеств сущностей методом их семантического объединения

1. Интеграция атрибутов каждого множества сущностей методом их семантического объединения
2. Интеграция ограничений целостности, ассоциированных с каждым множеством сущностей, методом их объединения
3. Интеграция множеств связей методом их семантического объединения; возможна генерализация множеств связей
4. Интеграция атрибутов каждого множества связей методом их семантического объединения
5. Повторная тщательная проверка типов отображений, соответствующих каждому множеству связей
6. Интеграция других ограничений целостности, которые напрямую не отражаются в ER-модели, методом их объединения; сформулировать их на естественном языке или языке логики предикатов первого порядка.

86. Что представляет собой генерализация множеств связей?

Замена нескольких специализированных множеств связей одним обобщенным множеством связей.

Вопросы и задания к параграфу 5.3

87. Какая задача решается на этапе логического проектирования данных?

Подготовка схемы БД в терминах и на языке логической модели СУБД.

88. Какие действия предусмотрены на этапе логического проектирования данных для реляционной модели? В каких случаях они выполняются?

1. Трансформация ER-схемы (ERM-схемы) в реляционную схему с помощью соответствующих правил.

Выполняется при любом логическом проектировании с использованием семантической методики.

Если можно улучшить качество и/или эффективность полученной реляционной схемы:

2. Проверка отношений полученной реляционной схемы на выполнение условий требуемых нормальных форм и их дальнейшая нормализация.

Выполняется, если в реляционной схеме остались аномалии вставки, обновления и удаления.

Второй этап заключается в выполнении алгоритма декомпозиции для каждого из предварительных отношений, полученных на первом этапе логического проектирования.

3. В необходимых случаях (неэффективность выполнения запросов к БД) введение контролируемой избыточности данных (денормализация схемы), которой сопутствуют меры, исключающие возникновение аномалий.

89. Какие факторы в основном влияют на успех применения семантической методики?

Успех семантической методики проектирования в основном определяется двумя факторами:

- мощностью применяемой семантической модели данных;
- способностью используемого набора правил преобразования семантической схемы в СУБД-ориентированную схему порождать в каждом конкретном случае идеальные логические схемы.

90. Вспомните простейшие правила перехода от ER-схемы к реляционной схеме БД.

1. Каждое множество сущностей представляется самостоятельным отношением, однозначные атрибуты множества сущностей становятся атрибутами отношения, ключи множества сущностей являются возможными ключами отношения; при необходимости в качестве первичного ключа отношения используется суррогатный атрибут.
2. Бинарные множества связей типа 1:M без атрибутов представляются дублированием первичного ключа 1-отношения в M-отношение.
3. Бинарные множества связей типа M:N без атрибутов представляются самостоятельными отношениями, куда дублируются первичные ключи отношений, построенных для множеств сущностей.
4. Множества связей с атрибутами представляются самостоятельными отношениями, куда дублируются первичные

ключи отношений, построенных для множеств сущностей.

Однозначные атрибуты множества связей становятся атрибутами этого отношения. Простейшие правила перехода от ER-схемы к реляционной схеме БД (продолжение)

5. Множества связей степени больше 2-х представляются самостоятельными отношениями, куда дублируются первичные ключи отношений, построенных для множеств сущностей. Однозначные атрибуты множества связей становятся атрибутами этого отношения.
6. Каждый многозначный атрибут множества сущностей представляется отдельным отношением, куда дублируется первичный ключ отношения, построенного для множества сущностей; второй атрибут этого отношения предназначен собственно для значения.
7. Каждый многозначный атрибут множества связей представляется отдельным отношением, куда дублируется первичный ключ отношения, построенного для множества связей; второй атрибут этого отношения предназначен собственно для значения.

91. За счет чего повышается качество схемы при использовании усовершенствованных правила перехода от ER-схемы к реляционной схеме БД?

За счёт анализа таких особенностей семантической схемы, как:

- бинарных множеств связей типа 1:1;
- значений минимальных кардинальных чисел отображений, определяемых бинарными множествами связей;
- специализаций и категоризаций.

92. Какие решения предлагают усовершенствованные правила для множеств связей типа 1:1? Какими рассуждениями следует сопровождать применение этих правил?

1. Каждое бинарное множество связей (МСв) типа $R(S1(1;1):S2(1;1))$ и множества сущностей (МСу) $S1$ и $S2$, в нем участвующие, заменяются в ER-схеме одним агрегированным МСу, с которым соединяются все другие МСв, имевшиеся у двух исходных МСу. Атрибуты нового МСу представляют собой объединение атрибутов обоих исходных МСу и МСв. Ключами полученного МСу являются ключи исходных МСу и МСв.

Целесообразность применения этого правила не вызывает никаких сомнений, поскольку, благодаря ему, будет получена явно более эффективная схема. Но могут возникнуть проблемы с именованием вновь образованного агрегата

2. Если МСу не вступает в связи типа $R(S1(1;1):S2(1;1))$, оно образует одно отношение, построенное на однозначных атрибутах этого МСу. Возможными ключами этого отношения являются ключи МСу.

Второе правило практически без изменений перешло в этот набор из ранее рассмотренного. В соответствии с ним **для каждого из оставшихся после применения первого правила множеств сущностей будет образовано свое отношение реляционной схемы**

3. Бинарное МСв типа $R(S1(1;1):S2(0;1))$ представляется дублированием первичного ключа отношения, построенного для МСу S1 (множества регулярных сущностей), в отношение, построенное для МСу S2 (множества слабых сущностей). Этот внешний ключ является возможным ключом второго отношения и имеет описатель NOT NULL.

Третье правило предлагает **представлять множество связей дублированием первичного ключа одного отношения в другое**. Образованный при этом внешний ключ является, с одной стороны, возможным ключом отношения, а, с другой стороны, является обязательным атрибутом.

Третье правило, как и многие другие правила этого набора, приводит к появлению дополнительных отношений, которые на первый взгляд являются лишними. С практической точки зрения это чревато дополнительными операциями соединения.

4. Бинарное МСв типа $R(S1(0;1):S2(0;1))$ представляется самостоятельным отношением, куда дублируются первичные ключи отношений, построенных для МСу S1 и S2. Каждый из этих внешних ключей является возможным ключом нового отношения и имеет описатель NOT NULL.

Четвертое правило применяется, когда бинарное множество связей типа 1:1 определяет **два частичных функциональных отображения**. В этом случае **множество связей представляется самостоятельным отношением с внешними ключами, идентифицирующими сущности обоих типов**.

В ситуации четвёртого правила для восстановления полной информации о связях двух сущностей потребуется уже две операции соединения отношений.

Первое и второе правила относились к случаю, когда оба определяемые ими функциональные отображения были полными, в третьем правиле одно из них полное, второе – частичное, а в четвертом – частичны оба.

93. Какие решения предлагают усовершенствованные правила для множеств связей типа $1:M$? Какими рассуждениями следует сопровождать применение этих правил?

1. Бинарное МСв типа $R(S1(1;1):S2(_;\infty))$ ($_$ означает, что минимальное кардинальное число отображения не имеет значения) представляется дублированием первичного ключа отношения, построенного для МСу $S1$, в отношение, построенное для МСу $S2$. Этот внешний ключ имеет описатель NOT NULL.

В этом случае без всяких сомнений надо **представлять** такое **множество связей с помощью внешнего ключа**. Поскольку отображение полно, он к тому же будет обязательным атрибутом.

2. Бинарное МСв типа $R(S1(0;1):S2(_;\infty))$ представляется самостоятельным отношением, куда дублируются первичные ключи отношений, построенных для МСу $S1$ и $S2$. Внешний ключ, являющийся дубликатом первичного ключа второго отношения (для $S2$), представляет собой возможный ключ нового отношения. Оба атрибута этого отношения имеют описатель NOT NULL.

Если функциональное отображение частично, усовершенствованный набор правил предлагает построить для множества связей типа $1:M$ самостоятельное отношение.

Один из внешних ключей этого отношения является его возможным ключом, а второй – нет.

Целесообразность применения этого правила также следует внимательно проанализировать. Возможно, лучше следует воспользоваться пятым правилом и в этом случае.

Два правила относятся к бинарным множествам связей типа 1:M. Первое правило регламентирует ситуацию, когда функциональное отображение является к тому же полным.

94. Как изменились решения в случае однозначных атрибутов множеств связей?

Если МСв имеет свои собственные однозначные атрибуты, они добавляются в то отношение, куда осуществляется дублирование первичного(-ых) ключа(-ей) для представления этого множества связей.

Правило определяет действия в случае, когда множество связей имеет свои собственные однозначные атрибуты. Понимать его надо следующим образом. Во всех предыдущих правилах использовались два способа представления множеств связей – самостоятельным отношением и через внешний ключ. И в том, и в другом случае создавались дубликаты первичных ключей множеств сущностей – внешние ключи, только в первом – в новом отношении для множества связей, а во втором – в ранее определенном отношении для множества сущностей. Именно в этих отношениях и следует разместить однозначные атрибуты множеств связей.

95. Какими методами могут быть представлены в реляционной модели специализации и категоризации?

Методы представления специализаций в реляционной схеме:

Метод 1: 1 отношение. Создается одно отношение, которое включает однозначные атрибуты суперкласса и всех подклассов, т.е. каждый объект в рамках специализации описывается кортежем только одного отношения.

Метод 2: n+1 отношение. Для суперкласса и каждого из подклассов создается по одному отдельному отношению. Один объект может описываться кортежами нескольких отношений: от 1 до n+1.

Метод 3: n отношений. Создается n отношений – по одному для каждого подкласса, однозначные атрибуты суперкласса включаются в каждое из этих отношений. Один объект может быть представлен кортежами нескольких отношений: от 1 до n.

Метод 4 (гибрид 2 и 3 методов): n+1 отношение. Для суперкласса и каждого из подклассов создается по одному отдельному отношению. Кроме этого, однозначные атрибуты суперкласса добавляются в

каждое отношение, построенное для подклассов. Один объект может быть представлен кортежами нескольких отношений: от 1 до n .

Метод 5: $n+2$ отношения. Помимо n отношений для подклассов и одного для суперкласса создается связующее отношение, которое имеет следующие атрибуты: имя родительского отношения; первичный ключ объекта в родительском отношении; имя дочернего отношения; первичный ключ объекта в дочернем отношении. Таким образом, один объект может описываться кортежами нескольких отношений: от 1 до $n+2$. Причем в связующем отношении к этому объекту может относиться несколько кортежей.

Метод 6: 2^n отношений. Для любого возможного поддерева в иерархии специализаций, включающего корневой суперкласс, создается одно отношение, которое содержит в качестве атрибутов все однозначные атрибуты множеств сущностей, принадлежащих поддереву. В конечном счете, каждый объект ПрО относится к одному из этих 2^n классов, и, значит, его кортеж надо создавать в соответствующем отношении. Таким образом, каждый объект в рамках рассматриваемого метода описывается кортежем одного и только одного отношения.

Методы представления категоризаций в реляционной схеме:

Метод 1: n отношений. Данный метод предполагает, что все однозначные атрибуты категории становятся атрибутами в отношениях, представляющих суперклассы. Отношение для категории не создается.

Метод 2: $n+1$ отношение. Для подкласса (категории) и каждого из суперклассов создается отдельное отношение. Варианты способов организации связей между кортежами отношений суперклассов и кортежами отношения категории аналогичны предлагавшимся для соответствующего метода специализаций.

Метод 3: $n+2$ отношения. Аналогичен методу 5 для специализаций.

96. Какие критерии могут направлять процесс выбора метода представления специализаций и категоризаций?

1. Простота реализации.

Помимо собственно времени, затрачиваемого на реализацию метода, критерий может включать и специфические требования, такие как учет потенциальной необходимости вносить изменения в реляционную схему (в частности, возможность появления дополнительных

подклассов). Другим примером специфических требований является желание упростить реализацию изменений в отношении принадлежности объекта к тем или иным подклассам специализации. Речь идет, например, о ситуации, когда врач получает или лишается лицензии по специальности. Одни методы могут упростить удовлетворение этих требований, в то время как другие – усложнить.

2. Эффективность хранения.

Имеется возможность за счет выбора метода снизить влияние некоторых факторов, отрицательно сказывающихся на эффективности хранения:

- А. дублирование значений атрибутов: - дублирование значений ключей; - дублирование значений остальных атрибутов;
- В. представление отсутствующих значений (NULL).

Если у вас на счету каждый байт внешней памяти, этот критерий является для вас основным. Особенно это касается СУБД, не обеспечивающих эффективные методы хранения данных на диске, в частности, неопределенных значений.

3. Эффективность обработки:

- а) обновления;
- б) проверки ограничений целостности (совокупность ограничений целостности, заданных как декларативно, так и процедурно, неизменна, однако, сложность проверки одного и того же ограничения будет различной в зависимости от выбранного метода);
- с) выборки.

Последний критерий предлагает оценивать методы с точки зрения эффективности процедур обработки данных. При этом следует проанализировать:

- какие команды будут выполняться при изменениях объектов ПрО, насколько они трудоемки;
- будут ли ограничения целостности, связанные со специализациями и категоризациями, представлены декларативно или потребуются написание специальных программ, какова их трудоемкость;
- обеспечит ли метод представления специализаций и категоризаций требуемую эффективность выборки

97. Какие факторы необходимо учесть при выборе метода?

При выборе метода для специализаций необходимо учесть следующие факторы(**факторы специализаций**):

1. Количество атрибутов и множеств связей, ассоциированных с суперклассом и подклассами.
2. Свойства специализации (полное или частичное участие членов суперкласса в подклассах, пересечение или непересечение подклассов).
3. Степень пересечения подклассов в случае пересекающейся специализации.
4. Степень неполноты в случае специализации с частичным участием.
5. Количество подклассов специализации, их динамизм.
6. Наличие множеств связей или многозначных атрибутов, для представления которых правила преобразования в реляционную схему создают для некоторого отношения внешние ключи, ссылающиеся на ключ суперкласса, но не входящие в состав первичного ключа этого отношения.
7. Выполнение условия, аналогичного условию пункта 6, для одного или нескольких подклассов.
8. Наличие множеств связей или многозначных атрибутов, порождающих в процессе трансформации в реляционную схему отношения, внешние ключи которых ссылаются на ключ суперкласса, а также входят в состав первичного ключа данного отношения.
9. Выполнение условия, аналогичного указанному в пункте 8, для одного или нескольких подклассов.
10. Использование несуррогатных (естественных) ключей для суперкласса или подклассов.

При выборе метода для категоризаций необходимо учесть следующие факторы(**факторы категоризаций**):

1. Количество атрибутов и множеств связей, ассоциированных с категорией и суперклассами.
2. Свойство полноты/неполноты участия для категоризации.
3. Степень неполноты в случае категоризации с неполным участием.
4. Количество суперклассов, их динамизм.
5. Наличие множеств связей или многозначных атрибутов, порождающих в процессе трансформации в реляционную схему отношения, внешние ключи которых ссылаются на ключ категории, но не входят в состав первичного ключа данного отношения.

6. Наличие множеств связей или многозначных атрибутов, порождающих в процессе трансформации в реляционную схему отношения, внешние ключи которых ссылаются на ключ категории, а также входят в состав первичного ключа данного отношения.
7. Использование естественных ключей для суперклассов и подкласса.

98. Какой принцип используется для денормализации отношений на завершающем этапе логического проектирования?

Цель этапа - определение необходимости ввода контролируемой избыточности за счет ослабления условий нормализации для повышения производительности системы.

К денормализации следует прибегать лишь тогда, когда нормализованная база не удовлетворяет требованиям, предъявляемым к производительности системы.

Кроме того, необходимо учесть и следующие **особенности**:

1. денормализация может усложнить физическую реализацию системы;
2. денормализация часто приводит к снижению гибкости;
3. денормализация может ускорить чтение данных, но при этом замедлить обновление записей.

Процесс денормализации должен проводиться по определенным **правилам**:

1. Создаваемые в результате денормализации избыточные структуры (столбцы и таблицы) должны добавляться в схему без удаления уже созданных при нормализации структур.
2. Клиентские приложения, осуществляющие изменения данных, должны воздействовать напрямую только на нормализованные структуры.
3. Данные избыточных структур должны модифицироваться исключительно специализированным программным кодом, реализованным либо в виде триггеров обновления нормализованных структур, либо в виде пакетных программ, периодически запускаемых в моменты низкой загрузки системы БД.

99. Опишите типичные случаи денормализации.

1. Объединение таблиц со связями типа один к одному (1:1)
2. Дублирование неключевых атрибутов родительской таблицы в связях один ко многим (1:M)
3. Создание агрегирующих столбцов родительской таблицы в связях один ко многим (1:M) Дублирование атрибутов внешних ключей в иерархии связей один ко многим (1:M)
4. Дублирование атрибутов внешних ключей в иерархии связей один ко многим (1:M)
5. Дублирование атрибутов в связях многие ко многим (M:N)
6. Введение повторяющихся групп полей
7. Объединение справочных таблиц с базовыми таблицами
8. Создание таблиц из данных, содержащихся в других таблицах

Вопросы и задания к параграфу 5.4

100. Какие действия осуществляются на этапе физического проектирования данных?

1. Анализ транзакций
2. Выбор файловой структуры
3. Определение индексов
4. Определение требований к дисковой памяти

101. Какие виды сегментов предоставляет СУБД Oracle для хранения таблиц?

1. Таблица
2. Кластер
3. Индекс-таблица

(Все сегменты: Таблица, Секция таблицы, Кластер, Индекс, Индекс-таблица, Индексная секция, Сегмент отката, Временный сегмент, Сегмент LOB, Индекс LOB)

102. Каких правил следует придерживаться при построении индексов?

1. Не создавать индекс на небольших отношениях
2. Следует создавать индекс на первичном ключе отношения
3. Ввести дополнительный индекс на внешнем ключе

4. Ввести дополнительные индексы на всех атрибутах, которые часто применяются в качестве дополнительного поискового ключа
5. Ввести дополнительные индексы на атрибутах, которые часто применяются в конструкциях WHERE, ORDER BY, GROUP BY;
6. Не индексировать атрибут или отношение, которые часто обновляются
7. Не индексировать атрибут, если в запросах с использованием этого атрибута обычно происходит выборка значительной части (например, 25%) кортежей в отношении
8. Не индексировать атрибуты, которые состоят из длинных символьных строк

103. Какими параметрами команды CREATE TABLE определяются требования к дисковой памяти?

[TABLESPACE <имя табличного пространства>] – табличное пространство, в котором будет создана таблица

[PCTFREE <целое>] – объем пространства, резервируемого в каждом блоке (процентное отношение от пространства данных, т.е. общего размера блока минус заголовков) для потенциального роста длины строк

[PCTUSED <целое>] – нижний предел используемости пространства блока (после заполнения до PCTFREE), ниже которого необходимо опуститься, чтобы блок вновь стал доступным для вставки

[INITTRANS <целое>] – начальное количество транзакций, которые могут одновременно обращаться к строкам блока данных (по умолчанию – 1)

[MAXTRANS <целое>] – максимальное количество транзакций, которые могут одновременно обращаться к строкам блока данных (по умолчанию – 255)

[<фраза STORAGE>] – фраза, определяющая правила выделения экстендов таблице

[LOGGING | NOLOGGING] – ключевые слова, определяющие будет ли информация сохраняться в журнальном файле информации о создании и последующих операциях изменения таблицы