Теория экономических информационных систем

Лекция 3.

Модели хранения данных.

Синтаксические модели.

Критерий качества создания базы данных

- Минимальная избыточность хранимой информации, выражаемая принципом: *каждое сообщение хранится в БД один раз*. Соблюдение данного принципа дает следующие преимущества:
  - Сокращается объем памяти ЭВМ, для хранения БД;
  - Сокращается трудоемкость ввода данных и упрощаются проблемы контроля достоверности информации;
  - Упрощаются алгоритмы корректировки данных;
- Использование экономических показателей позволяет построить структуру БД с минимальной избыточностью, если сначала расчленить все сведения в ЭИС на показатели, а потом объединить атрибуты родственных показателей по принципу:
  - *в один файл включается группа показателей с одинаковым составом атрибутов-признаков*.

Модель данных

- При рассмотрении структуры базы данных одной из основных составляющих является модель данных. Под моделью данных понимается совокупность из трех составляющих:
  - Множество информационных конструкций, допускаемых этой моделью;
  - Множество допустимых операций над данными;
  - Множество ограничений, наложенных на информационные конструкции.

Модели данных

- Наиболее распространенными моделями данных являются следующие:
  - реляционная;
  - сетевая;
  - иерархическая.
- **Модель данных** – представляет собой инструмент для представления данных в БД.

Реляционные базы данных

- Реляционная база данных характеризуется следующими компонентами:
  - Информационной конструкцией – отношение с двухуровневой структурой;
  - Допустимые операции – проекция, выборка, соединения и др.
  - Ограничения – функциональные зависимости между атрибутами отношений.

Структура реляционной БД

- Каждому объекту P ставится в соответствие некоторое множество атрибутов, (A1, …, An). Отдельный объект класса P описывается строкой величин (a1, …, an).
- Строка (a1, …, an) называется *кортежем*. Всему классу объектов соответствует множество кортежей, называемое *отношением*. Выражение P (A1, …, An) называется *схемой отношения* P.
- Каждое отношение представляет собой состояние класса объектов в некоторый момент времени.

Представление реляционной БД

- Множество значений отношения можно представить в виде таблицы, в которой соблюдаются соответствия:
  - Название таблицы и перечень названий граф (столбцов) соответствует схеме отношений;
  - Строке таблицы соответствует кортеж отношения;
  - Все строки таблицы различны;
  - Порядок строк и столбцов произвольный.

Схема реляционной БД

- Схема реляционной БД содержит компоненты:
  - S(rel) = <A, R, Dom, Rel, V(s)>
  - Здесь
  - A – множество имен атрибутов
  - R – множество имен отношений
  - Dom – вхождения атрибутов в домены
  - Rel – вхождения атрибутов в отношения
  - V(s) – множество ограничений
- Описание процессов обработки отношений выполняется двумя способами:
  - Указание перечня операций, приводящих к необходимому результату (процедурный подход)
  - Указание свойств, которым должно удовлетворять результирующее отношение (декларативный подход)

Операции над отношением реляционной БД

- **Проекция** – операция, переносящая в результирующее отношение атрибуты, которые указаны в условии операции;
- **Выборка** – операция, переносящая в результирующее отношение строки из исходного отношения, которые удовлетворяют условию выборки
- Операции **объединения, пересечения** и **вычитания** отношений – выполняются над отношениями с одинаковой структурой. Соответствуют операциям над множествами.
- **Операция соединения** выполняется над двумя исходными отношениями и создает одно результирующее. Каждая строка первого сопоставляется по очереди со всеми строками второго отношения, и если для этой пары выполняется условие соединения, то они сцепляются и образуют очередную строку в результирующем отношении.

Нормализация отношений

- Центральная задача проектирования базы данных ЭИС – определение количества отношений и их атрибутный состав.
- Рациональный способ формирования отношений включает следующие требования:
  - Множество отношений должно обеспечивать минимальную избыточность представления информации;
  - Корректировка отношений не должна приводить к двусмысленности или потери информации;
  - Перестройка отношений при добавлении в БД новых атрибутов должна быть минимальной.
- **Нормализация** реляционной БД – метод преобразования отношений в соответствии с описанными требованиями.

Функциональные зависимости и ключи

- Функциональные зависимости определяются для атрибутов, находящихся в одном и том же отношении, удовлетворяющем 1НФ (первой нормальной форме).
- Атрибут А *функционально определяет* атрибут В, если в любой момент времени каждому значению А соответствует значение В (А -> В).
- **Вероятным ключом** называется множество атрибутов, каждое сочетание которых встречается только в одной строке отношения, и никакое подмножество атрибутов не обладает этим свойством.
- **Первичный ключ** – вероятный ключ, по значениям которого выполняется контроль достоверности информации. Каждое значение первичного ключа встречается только в одной строке отношения.
- *Набор атрибутов первичного ключа функционального определяет любой атрибут отношения.*

Нормальные отношения

- **Нормальные отношения** – это отношения с дополнительно соблюдаемыми ограничениями. С увеличением номера нормальной формы должно соблюдаться большее число ограничений.
- Первая нормальная форма – отношение в нормализованном файле, имеющее вероятный ключ.

Вторая нормальная форма отношений

- Отношение имеет **вторую нормальную форму,** если оно соответствует 1НФ и не содержит неполных функциональных зависимостей.
- **Неполная функциональная зависимость – это** зависимость удовлетворяющая условиям:
  - Вероятный ключ отношения функционально определяет некоторый неключевой атрибут;
  - Часть вероятного ключа функционально определяет этот же неключевой атрибут.
- Отношение, не соответствующее 2НФ, характеризуется избыточностью хранимых данных.

Третья нормальная форма отношений

- Отношение соответствую **третьей нормальной форме,** если оно соответствует 2НФ и среди его атрибутов отсутствуют транзитивные функциональные зависимости.
- Транзитивная функциональная зависимость включает в себя:
  - Вероятный ключ отношения функционально определяет неключевой атрибут;
  - Этот атрибут функционально определяет другой неключевой атрибут.

Алгоритм нормализации к 3НФ

- Исходные данные – множество всех реквизитов базы данных.
- Метод – создание отношений, в которых соблюдается одна функциональная зависимость либо ни одной.
- **Шаг 1**. Получить исходное множество функциональных зависимостей для реквизитов рассматриваемой базы данных.
  - Если исходные функциональные зависимости не удается определить путем анализа содержательных ролей реквизитов, приходится использовать перечисление и отбраковку допустимых вариантов функциональных зависимостей.
  - Для этого рассматриваются все сочетания по два реквизита и в каждом случае доказывается или отвергается функциональная зависимость. Затем рассматриваются сочетания:
    - по три реквизита, где два первых функционально определяют третий;
    - по четыре реквизита, где первые три функционально определяют четвертый и т.д.

Алгоритм нормализации к 3НФ (продолжение)

- **Шаг 2**. Получить минимальное покрытие множества функциональных зависимостей. В частности, требуется объединить функциональные зависимости с одинаковой левой частью в одну зависимость. Обозначим полученное множество зависимостей через $F=\{f_1, \ldots, f_k\}$.
- **Шаг 3**. Определить первичный ключ отношения.
- **Шаг 4**. Для каждой функциональной зависимости f_i создать проекцию исходного отношения $R_i=R[X_i]$, где X_i – объединение реквизитов левой и правой частей f_i.

Алгоритм нормализации к 3НФ (продолжение)

- **Шаг 5**. Если первичный ключ исходного отношения не вошел полностью ни в одну проекцию, полученную на шаге 4, необходимо создать отдельное отношение из реквизитов ключа.
- Для практического применения алгоритма нормализации необходимо решить вопросы:
  - **Как учитывается наличие взаимно-однозначных соответствий?**
  - **Как сократить объем перебора при первоначальном определении множества функциональных зависимостей?**
- Для взаимно-однозначных соответствий принято выделение старшего реквизита, который затем представляет все реквизиты взаимно-однозначного соответствия.

Ациклические базы данных

- Некоторые ограничения в предметной области и БД не может быть описаны с помощью функциональных зависимостей, что приводит к необходимости рассмотрения новых типов зависимостей – многозначных.
- Специальный класс реляционных БД – **ациклические БД** – характеризуется однозначной декомпозицией на основе многозначных зависимостей и иных свойств.
- Для описания ациклических БД используют графы соединений на множестве отношений {S1, S2, …}. Вершины графа – имена отношений. Дуги графа <Si,Sj> существует, если в структуре отношений есть общие реквизиты. В графе соединений требуется чтобы для каждой пары отношений Si и Sj с общим реквизитом A(i,j) существовал A – путь между Si и Sj. Если граф можно превратить в дерево с помощью исключения некоторых дуг при сохранении названного требования, то база данных называется **ациклической**.

Алгоритм проверки структуры БД на ацикличность

- Исходные данные – список отношений с указанием реквизитного состава каждого отношения.
- **Шаг 1**. Если реквизит встречается только в одном отношении, то необходимо вычеркнуть данный реквизит из этого отношения.
- **Шаг 2**. Если все реквизиты некоторого отношения находятся среди реквизитов другого отношения, то первое отношение вычеркивается из списка.
- Если в результате будут вычеркнуты все отношения, то база данных является ациклической.
- База данных с циклическим графом соединений может давать некорректные ответы на запросы из-за наличия неравноценных путей доступа при реализации запроса.

Сетевая модель данных

- Сетевая модель данных представляется как множество отношений и веерных отношений.
- **Веерное отношение** W(R,S) – пара отношений, включающая одно основное R и одно зависимое S и связь между ними, при условии, что каждое значение зависимого отношения связано с единственным значение основного отношения.
- Сетевые базы данных разделяются на **двухуровневые** и **многоуровневые** сети.
- Ограничения двухуровневых сетей – каждое отношение может существовать в одной из следующих ролей:
  - Вне каких-либо веерных отношений;
  - В качестве основного отношения в любом числе веерных отношений;
  - В качестве зависимого отношения в любом числе веерных отношений.
- Запрещается существования в одном контексте в качестве зависимого и одновременно в качестве основного в другом контексте.
- Современные сетевые СУБД поддерживают, как правило, двухуровневую сеть. Операция связывания отношений в реляционной СУБД приводит к двухуровневым системам отношений.

Схема сетевой БД

- Схема сетевой БД включает следующие компоненты:
  - S(net) = <A, R, WW, Dom, Rel, Net, V(s)>
  - Здесь
    - WW – множество веерных отношений
    - Net – вхождение отношений в веерные отношения.
- Способы включения в веерные отношения:
  - **Автоматический** – при появлении нового значения основного отношения оно сразу ставится в соответствие некоторому значению зависимого отношения и образует новый элемент веерного отношения
  - **Неавтоматический** – несоблюдение вышеизложенного правила
- Способ исключения из веерного отношения:
  - **Обязательный** – после того, как значение включено в основное отношение, оно является его постоянным членом. Его можно обновить, но нельзя удалить
  - **Необязательный** – означает, что любое значение основного отношения может быть удалено.

Схема сетевой БД

- Схемой сетевой БД называется описание всех отношений с указанием атрибутного состава и ключей каждого отношения, а также веерных отношений.
- В схеме сетевой БД отношения и веерные отношения часто рассматриваются как файлы и связи. При таком подходе сетевая структура может быть рассмотрена как множество файлов:
  - F={F1(X1), F2(X2), …, Fi(Xi), …}
    - где Xi – атрибут ключа в файле Fi
- Дополнительно вводится граф сетевой структуры B с вершинами {X1, X2, …, Xi, …}. Дуга графа <Xi, Xj> существует, если Xi является частью Xj и Fi(Xj) является подмножеством Fi. Здесь предполагается, что ключ основного файла содержится в зависимом файле.
- База данных называется **ациклической ,** если между любыми двумя вершинами графа B существует не более одного пути.
- Двухуровневые сети всегда обладают свойством ацикличности.

Двухуровневые сети

- Для двухуровневых сетевых СУБД вводится еще два ограничения:
  - первичный ключ основного отношения может быть только однореквизитным;
  - веерное отношение устанавливается, если первичный ключ основного отношения является частью первичного ключа зависимого отношения.
- В структуру основного и зависимого отношения вводится дополнительный реквизит, называемый *адрес связи*. Значения адресов связи обеспечивают в веерном отношении соответствие каждого значение зависимого отношения S с единственным значением основного отношения R.
- Связь значений зависимого отношения с единственным значением основного отношения обеспечивается следующим образом:
  - Адрес связи некоторой записи основного отношения указывает на одну из записей зависимого отношения (значением адреса связи основного отношения является начальный адрес этой записи зависимого отношения);
  - Адрес связи указанной записи зависимого отношения – на следующую запись зависимого отношения, связанную с той же записью основного отношения.

Алгоритм приведения к двухуровневой структуры сети

1. Для каждой функциональной зависимости вида A -> B создается файл Fi(A,B). Каждый блок взаимно-однозначных соответствий порождает файл с ключом, равным старшему по объему атрибуту.
2. У всех файлов, полученных на шаге 1, проверяется условие для ключей (Ki – часть Kj). Если оно соблюдается, то из соответствия файлов создается веерное отношение Wij(Fi,Fj).
3. Если на шаге 2 будут получены два веерных отношения Wij и Wjk, то все атрибуты файла Fi передаются в файл Fj и Fi вместе с Wij уничтожается.
4. Атрибуты, не вошедшие в состав веерных отношений на шаге 2, добавляются в те файлы Fn,где они будут неключевыми. При наличии нескольких подходящих файлов предпочтение отдается основным файлам. Если требуемый файл Fn отсутствуют, то создается новый файл из атрибутов первичного ключа, и повторяются шаги 2,3 и 4.

Иерархическая модель данных

- Иерархическая модель данных имеет много общих черт с сетевой моделью.
- Допустимыми конструкциями иерархической модели являются:
  - **Отношение**;
  - **Веерное отношение;**
  - **Иерархическая база данных.**
- В отличие от сетевой и реляционной моделей, в иерархической модели допускается отображение одной предметной области в несколько иерархических баз данных.

Понятие иерархической базы данных

- **Иерархической базой данных** называется множество отношений и веерных отношений, для которых выполняются ограничения:
  1. Существует единственное отношение, называемое **корневым**, которое не является зависимым ни в одном веерном отношении.
  2. Все остальные отношения являются зависимыми отношениями только в одном веерном отношении.

Иерархическая база данных

- **Записью иерархической базы данных** называется множество значений, содержащих одно значение корневого отношения и все вееры, доступные для него в соответствии со структурой иерархической базы данных.
- Для веерных отношений справедливо утверждение: *если существует веерное отношение, то ключ зависимого отношения функционально определяет ключ основного отношения, и наоборот, если ключ одного отношения функционально определяет ключ второго отношения, то первое отношение может быть зависимым, а второе – основным в некотором веерном отношении*.

Алгоритм получения структуры иерархической базы данных

- Для каждой функциональной зависимости вида A -> B создается отношение Si(A,B). Каждый блок взаимно-однозначных соответствий порождает отношение с ключом, равным старшему по объему понятия атрибуту.
- Разделить отношения на группы по признаку: два отношения находятся в одной группе, если их ключи функционально определяют хотя бы один общий атрибут.
- У всех пар отношений группы проверяется условие для ключей Kj -> Ki. Если оно соблюдается, то из соответствующих отношений создается веерное отношение Wij(Si,Sj).
- Найти в группе цепи веерных отношений и сцепить их в дерево. Элемент цепи образуется по условию Wij – Wjk.
- Атрибуты, оставшиеся вне цепей на предыдущих шагах, добавить в структуру тех отношений, где они будут неключевыми, либо в структуру отношений, соответствующих висячим вершинам дерева.
- Если группы содержат общие отношения, то решить вопрос о целесообразности установления логических связей между иерархическими БД.
- Сократить список атрибутов в сегментах за счет удаления атрибутов зависимого отношения, общих в паре «основной – зависимый»

Сравнение моделей данных

- **Достоинства** реляционной модели:
  - **Простота.** В реляционной модели присутствует одна информационная конструкция, которая формализует табличное представление.
  - **Теоретическое обоснование**. Существуют теоретически обоснованные методы нормализации отношений.
  - **Независимость данных.** Изменение в структуре реляционной БД приводит, как правило, к минимальным изменениям в прикладных программах.
- **Недостатки** реляционной модели:
  - Низкая скорость при выполнении операций соединения.
  - Относительно большой расход памяти при представлении реляционной БД.

Сравнение моделей данных

- **Достоинства** иерархической модели:
  - **Простота**. Хотя модель использует три информационные конструкции, иерархический принцип соподчиненности понятий является естественным при описании предметной области.
  - **Минимальный расход памяти**. Иерархическая модель позволяет получить представление с минимально требуемой памятью.
- **Недостатки** иерархической модели:
  - **Неуниверсальность.** Многие варианты взаимоотношений невозможно реализовать средствами иерархической модели.
  - **Допустимость только навигационного принципа доступа к данным.**
  - **Доступ к данным осуществляется только через корневое отношение.**

Сравнение моделей данных

- **Достоинства** сетевой модели:
  - **Универсальность**. Возможности сетевой модели являются наиболее обширными в сравнении с другими моделями.
  - **Возможность доступа к данным через несколько отношений**. Ускорение доступа к хранимой информации через любые основные отношения.
- **Недостатки** сетевой модели:
  - **Сложность.** Наличие различных понятий в модели, вариантов их взаимосвязей и особенностей реализации.
  - **Допустимость только навигационного принципа доступа к данным.**

Модель инвертированных файлов

- Модель инвертированных файлов – частный случай сетевой двухуровневой модели данных.
- Основные информационные конструкции в данной модели:
  - Основной файл (аналог отношения)
  - Инвертированный файл
  - Список связи
- Обозначим через {F1, F2, …, Fn} множество основных файлов. Все записи файлов имеют в пределах базы данных единую нумерацию.

Модель инвертированных файлов

- В основном файле Fi выделяется один или несколько атрибутов, по значениям которых формируются инвертированные файлы и списки связи.
- Выделенный атрибут – A – может принимать в Fi несколько различных значений - {a(1), a(2), …, a(k)}.
- Поставим в соответствие каждому значению a(j) множество записей файла Fi, в которых это значение связано с именем атрибута A. Определенная таким образом последовательность значений атрибута A и номеров записей основного файла Fi называется инвертированным файлом, который обозначим как A(Fi).
- Рассмотрим два файла Fi и Fj, в структуре которых имеется общий атрибут A. Существует два списка связи (Fi,Fj) и (Fj,Fi). В списке (Fi,Fj) для каждого номера записи из файла Fi указываются номера записей из файла Fj, имеющих то же самое значение атрибута A.

Информационно-поисковые системы

- Преимущество модели инвертированных файлов проявляется при реализации выборки с большим количеством условий. Каждое условие выборки соответствует множеству записей, и комбинация условий означает манипулирование ранее полученными из инвертированных файлов множествами номеров записей.
- Модель инвертированных файлов служит основой для построения информационно-поисковых систем.

Информационно-поисковые системы

- Модели инвертированных файлов соответствуют дескрипторные языки (вид информационно-поисковых языков).
- **Дескриптором** (ключевым словом) называется отдельное слово или словосочетание, используемое для краткого обозначения темы документа, хранящегося в БД информационно-поисковой системы.
- Процесс создания БД информационно-поисковой системы завершается формированием главного инвертированного файла, в котором для каждого значения дескриптора, указываются номера записей, среди атрибутов которых есть слова или словосочетания, совпадающие с этим дескриптором.

Литература

- А.И. Мишенин. Теория экономических информационных систем.