лекция №5

Дополнительные операции реляционной алгебры предложены Дейтом

Дейтом были предложены следующие дополнительные операции реляционной алгебры: переименование данных, расширение, подведение итогов, присвоение, вставки, обновления данных, удаление данных, реляционного сравнения. Рассмотрим каждую из этих операций детально.

Операция переименования данных.

RENAME <исходное отношение> <старое имя атрибута> AS <новое имя атрибута>

Эта операция позволяет переименовать атрибуты. Исходное отношение задается именем отношения или выражением реляционной алгебры, который заключается в круглые скобки.

Операция множественного переименования атрибутов.

RENAME <отношение> <старое имя атрибута 1> AS <новое имя атрибута 1>, <старое имя атрибута 2> AS <новое имя атрибута 2> ...

Операция расширения.

EXTEND <исходное отношение> ADD <выражение> AS <новый атрибут>

Операция расширения порождает новое отношение похоже на исходное, которое отличается наличием добавленного атрибута, значение которого находят путем некоторых скалярных вычислений. Исходное отношение может быть задано именем отношения или с помощью выражения реляционной алгебры, заключается в круглые скобки. При этом имя нового атрибута не должно входить в заголовок исходного отношения и не может использоваться в «выражении». Кроме обычных арифметических операций сравнения в выражении используются итоговые функции, такие как: COUNT (количество) SUM (сумма) AVG (среднее арифметическое) MAX (максимальное значение) MIN (минимальное значение).

Пользуясь операцией расширения иногда выполняют переименования атрибутов. Для этого в выражении нужно указать имя атрибута, а в конструкции AS определить новое имя этого атрибута, а затем выполнить проекцию полученного отношения на множество атрибутов, включая старый атрибут.

Операция множественного расширения.

EXTEND <отношение> ADD <выражение 1> AS <новый атрибут 1>, <выражение 2> AS <новый атрибут 2> ...

Эта операция аналогична операции расширения, она позволяет в одной синтаксической конструкции вычислять несколько новых атрибутов.

Операция подведения итогов.

SUMMARIZE <исходное отношение> ADD <выражение> AS <новый атрибут>

Данная операция выполняет вертикальные или групповые вычисления. Исходное отношение задается именем отношения или выражением реляционной алгебры в круглых скобках. Результатом операции SUMMARIZE является отношение R с заголовком, состоящий из атрибутов списка расширенного новым атрибутом. Для получения тела отношения R сначала проводится проектирование исходного отношения на атрибуты, а затем каждый кортеж отношения расширяется новым N + 1 атрибутом. Поскольку проектирование приводит к сокращению количества кортежей, потому удаляются одинаковые кортежи, то считают, что происходит своеобразное группировки кортежей исходного отношения.

Значение N + 1 атрибута каждого кортежа отношения R формируется путем вычисления выражения над соответствующей этому кортежа группой кортежей исходного отношения. Функция COUNT определяет количество кортежей в каждой из групп начального отношения.

Операция множественного подведения итогов.

Операция множественного подведения итогов подобная соответствующим операциям переименования и расширения. Она выполняет одновременно несколько вертикальных вычислений и записывает результаты в отдельные новые атрибуты.

Операция присвоения.

<Выражение-цель> = <выражение-источник>

Операция присвоения является основной операцией, меняет тело существующего отношения, кроме нее меняют тело существующего отношения операции вставки, обновления и удаления. В операции присвоения слева указано имя отношения, а справа - некоторое выражение реляционной алгебры, при этом эти выражения должны быть совместимы по структурами и задавать совместимы по структуре отношения.

Выполнение операции присвоения сводится к замене предыдущего значения на новое. С помощью этой операции можно осуществлять добавление и удаление кортежей.

Операция вставки.

INSERT <выражение-источник> INTO <выражение-цель>

Оба выражения должны быть совместимы по структуре. Выполнение операции сводится к вычислению выражения источника и вставки полученных кортежей в заданное отношение выражение-цель.

Операция обновления.

UPDATE <выражение-цель> <список элементов>

Где список элементов является последовательностью разделенных запятыми операций присвоения, имеют следующий вид:

<Атрибут> = <скалярное выражение>

Результатом выполнения этой операции является отношение полученное после присвоения соответствующих значений атрибутам отношения, задаются целевым выражением.

Операция удаления.

DELETE <выражение-цель>

Здесь выражение-цель является реляционным выражением, описывающим кортежи, которые удаляются.

Операция реляционного сравнения.

<Выражение 1> 0 <выражение 2>

Здесь оба выражения задают совместимые по структуре отношения, а знак 0 - одну из следующих операций: =, , <>.

Операции реляционного исчисления

К операциям реляционного исчисления относятся операции, в которых запросы к существующей реляционной СУБД записываются с помощью свойств искомого отношения, без конкретизации процедуры его получения.

Принципиальное отличие между реляционной алгеброй и реляционным исчислением состоит в том, что в первом случае процесс получения результата описывается явным образом, путем указания набора операций, которые нужно выполнить для получения результата. Во втором же случае мы указываем, что мы хотим получить и откуда это взять, без конкретизации процедуры получения.

Внешне подходы сильно отличаются: один из них предписывающий (реляционная алгебра), а другой описательный (реляционное исчисление), но на более низком уровне подходы эквивалентны. Это обусловлено тем, что любые выражения реляционной алгебры могут быть преобразованы в семантический эквивалент выражений реляционного исчисления и наоборот. Для этих преобразований используют алгоритм редукции Кодда или алгоритмы других авторов.

Предположим, что мы работаем с БД, которая определяется схемам:

СОТРУДНИКИ (спивробитник\_номер (ключ), спивробитник\_имья,

                                спивробитник\_зарплата, спивробитник\_номер\_виддилу)

ОТДЕЛ (виддил\_номер (ключ), виддил\_килькисть, виддил\_начальник).

Реляционное исчисление является частью формального механизма исчисления предикатов первого порядка. Предикаты - это высказывания в виде функций.

Базовыми понятиями исчисления являются понятие переменной с определенной областью допустимых значений и понятие правильно построенной формулы, опирающейся на переменные, предикаты и кванторы. В зависимости от того, что является областью определения различают вычисления кортежей и исчисление доменов.

В исчислении кортежей областями определения переменных являются отношения БД, то есть допустимым значениям каждой переменной является кортежи некоторого отношения.

В исчислении доменов областями определения переменных являются домены, на которых определены атрибуты БД, то есть допустимым значениям каждой переменной является значение некоторого домена.

В исчислении кортежей для определения кортежных переменной используется оператор RANGE. Например, для того, чтобы определить переменную СОТРУДНИК, областью определения которой является отношение СОТРУДНИКИ, нужно использовать следующую конструкцию:

RANGE СОТРУДНИК IS СОТРУДНИКИ

Из этого определения следует, что в любой момент времени переменная СОТРУДНИК представляет некоторый кортеж отношения СОТРУДНИКИ. При использовании кортежных переменных в формулах можно ссылаться на значение атрибута переменной (аналогично структурам в языке СИ). Например, для того чтобы сослаться на значение атрибута СПИВРОБИТНИК\_ИМьЯ, переменной СОТРУДНИК, используется конструкция:

СОТРУДНИК. СПИВРОБИТНИК\_ИМьЯ

Правильно построенные формулы

(WFF - Well Formed Formula)

WFF используются для выражения условий, налагаемых на кортежных переменные. Основой WFF простые сравнения (comporison), которые представляют собой операции сравнения скалярных значений (значений атрибутов переменных или литерально заданных констант). Например, конструкция:

СПИВРОБИТНИК.СПИВРОБИТНИК\_НОМЕР = 140

является простым сравнением, то есть она и есть WFF. Более сложные варианты WFF строятся с помощью логических связей, с использованием логических операций: NOT, AND, OR, IF ... THEN.

Если Form - правильно построенная формула, а Comp - простое сравнение, то NOT Form; Comp AND Form; Comp OR Form и IF Comp THEN Form тоже являются WFF. Кроме этого допускается построение WFF с помощью кванторов. Если Form - правильно построенная формула, в которой участвует переменная VAR, то конструкции:

EXISTS VAR (Form) - квантор существования

FORRAL VAR (Form) - квантор всеобщности

являться тоже правильно построенными формулами.

Переменные, которые входят в WFF могут быть свободными или связанными. Все переменные, входящие в WFF, для построения которой не использовались кванторы свободны. Это означает, что если для какого-либо набора значений свободных кортежных переменных после вычисления WFF получено значение true, то эти значения кортежных переменных могут входить в отношения-результат. Если же имя переменной использовалось сразу после квантора, при построении WFF вида EXISTS VAR (Form) или FORRAL VAR (Form), то в этой WFF и во всех WFF построенных с ее участием, VAR является связующим переменной. Это значит, что такую ​​переменную не видно за пределами минимальной WFF, которая связала эту переменную.

При вычислении значения такой правильно построенной формулы, используется не одно значение связанной переменной, а вся ее область определения.

Пусть СПИВРОБ1 и СПИВРОБ2 - две кортежных переменные, определенные в отношении СОТРУДНИКИ.

Пример WFF 1:

EXISTS СПИВРОБ2 (СПИВРОБ1.СПИВРОБ\_ЗАРПЛАТА> СПИВРОБ2.

                                  СПИВРОБ\_ЗАРПЛАТА)

Для текущего кортежа переменной СПИВРОБ1цей выражение приобретает значение true только тогда, когда во всем отношении СОТРУДНИКИ находится такой кортеж (связанный с переменной СПИВРОБ2), что значение его атрибута СПИВРОБ\_ЗАРПЛАТА удовлетворит внутреннюю условие сравнения.

Пример WFF 2:

FORRAL СПИВРОБ2 (СПИВРОБ1.СПИВРОБ\_ЗАРПЛАТА> СПИВРОБ2.

                                     СПИВРОБ\_ЗАРПЛАТА)

Для текущего кортежа переменной СПИВРОБ1набувае значение true только в том случае, если для всех кортежей отношения СОТРУДНИКИ (связанных с переменной СПИВРОБ2) значения атрибута СПИВРОБ\_ЗАРПЛАТА удовлетворяет условию сравнения.

На самом деле говорят не о связанных и свободные переменные, а о свободных и связанных вхождения переменных.

Примером языка, которая основана на основе вычисления кортежей является язык SQL.

Целевые списки и выражения реляционного исчисления

WFF обеспечивают средства формирования условия выборки по отношению БД. Для того чтобы можно было использовать вычисления для реальной работы с БД нужен еще один компонент, определяющий набор и имена столбцов отношение-результата. Этот компонент называется целевым списком (target-list).

Целевой список строится из целевых элементов, каждый из которых может иметь следующий вид: \* var.attr, где var - имя свободной переменной соответствующей WFF, а attr - имя атрибута отношения, на котором определена переменная var.

Выражением реляционного исчисления кортежей называют конструкцию вида:

target\_list WHERE WFF

Значением выражения является отношение, тело которого определяется WFF, а набор атрибутов и их имена - целевым списком target\_list.

Реляционное исчисление доменов

В исчислении доменов областью определения переменных является не отношение, а домены. Относительно БД СОТРУДНИКИ - ОТДЕЛЫ можно говорить, например, о доменных переменных СПИВРОБИТИК\_ИМьЯ (значения - допустимые имена) или СПИВРОБИТИК\_НОМЕР (значения - допустимые номера сотрудников).

Основное отличие вычисления доменов от исчисления кортежей является наличие дополнительного набора предикатов, позволяющих выражать условия членства.

Если R отношение с атрибутами <a1, a2, ..., an>, то условие членства имеет вид: R (ai1: vi1, ai2: vi2 ..., aim: vim) (m <= n), где vij - это либо литеральная константа, задается, или имя доменной переменной.

Условие членства приобретает значение TRUE только тогда, когда в отношении R существует кортеж, содержащий заданные значения определенных атрибутов. Если vij - константа, то на атрибут aij накладывается жесткое условие, которое не зависит от текущих значений доменных переменных.

Если же vij - имя доменной переменной, то условие членства может принимать различные значения при различных значениях этой переменной.

Реляционное исчисление доменов является основой большинства языков запросов, основанных на использовании форм. На этом исчислении базируется язык запросов по образцу QUERY BY EXAMPLE.

Контрольные вопросы:

1. Назовите операции реляционной алгебры, которые были предложены Дейтом.

2. Укажите итоговые операции, используемые в выражении операции расширения, какие действия они выполняют?

3. Перечислите операции, которые могут использоваться в выражении реляционного сравнения.

4. Какие операции относятся к операциям реляционного исчисления?

5. Что такое предикат?

6. Что является областью определения переменных в исчислении кортежей и в исчислении доменов?

7. Что такое правильно построенная формула (WFF) и для чего она используется?

8. Дайте определение свободным и связанным переменным правильно построенной формулы.

9. Что собой представляет целевой список?

10. Назовите основное отличие вычисления доменов от исчисления кортежей.

Тестовые задания:

1. Какая функция определяет количество записей, возвращаемых запросом:

а) AVG;

б) MAX;

в) STDEV;

г) FIRST;

д) Нет правильного ответа.

2. Какие булевы операторы заcтоcовуютьcя в условиях выбора запроса?

 а) =,>, <;

 б) Sum, Avg, Min;

 в) Max, Min, Avg;

 г) AND, OR, NOT;

 д) Нет правильного ответа.

3. Какая функция вставляет новую строку в представление?

а) UPDATE;

б) UNION;

в) INSERT;

г) EXISTS;

д) Нет правильного ответа.

4. Какая функция обновляет любое поле в существующем строке?

а) FETCH;

б) UPDATE;

в) GROUP BY;

г) INSERT;

д) Нет правильного ответа.

5. Какая функция удаляет существующую строку по представлению?

а) INSERT;

б) DELETE;

в) UPDATE;

г) UNION;

д) Нет правильного ответа.

Рекомендуемая литература:

1. Бородаев В.А., Кустов В.Н. Банки и базы данных: Учебное пособие. Л .: Вики, 1989.

2. Основы современных компьютерных технологий: Учебное пособие / Под редакцией проф. Хомоненко А.Д. Авторы: Артамонов Б.Н., Брякалов Р.А., Гофман В.Э. и другие. СПб: КОРОНА принт, 1998.

3. Системы управления базами данных и знаний: Довид.вид. / Наумов А. М., Вендров А. М., Иванов В. К. и др; Под. ред. Наумова А. Н. - М.: Финансы и статистика, 1991.

4. Дейт К.Дж. Введение в системы баз данных .; Пер. с англ. 6-е изд. К .: Диалектика, 1998. - 784 с.

5. Зомуяин А.В. Системы программирования баз данных и знаний. Новосибирск .; Наука. Сиб. от-ния, 1990.

6. Мартин Дж. Организация баз данных в вычислительных системах. - М .: Мир, 1980, 260с.

7. Романов Б.А., Кушниренко А.С. dBase IV: Назначение, функции, применение. - М .: Радио и связь, 1991. - 384 с.

8. Ульман Дж. Основы систем баз данных. - М .: Финансы и статистика, 1983.

9. Хомоненко А.Д., Цыганков В.М., Мальцев М. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. - Издание второе, дополненное и переработанное - СПб .: КОРОНА принт, 2002. - 672с.

10. Гайдаржи В.И. Дацюк А.А. Основы проектирования и использования баз данных: Учебное пособие. Второе издание постирал. и полным. - М .: ИПЦ "Издательство Политехника", ООО "Фирма Периодика" 2004. - 256 с.