Реляционное исчисление и реляционная алгебра

Ранее утверждалось, что реляционная алгебра и реляционное исчисление в своей основе эквивалентны. Обсудим это утверждение более подробно. Вначале Кодд показал в своей статье, что алгебра, по крайней мере, мощнее исчисления. (Термин "исчисление" будет использоваться для обозначения исчисления кортежей.) Он сделал это, придумав алгоритм, называемый "алгоритмом редукции Кодда", с помощью которого любое выражение исчисления можно преобразовать в семантически эквивалентное выражение алгебры. Мы не станем приводить здесь полностью этот алгоритм, а ограничимся примером, иллюстрирующим в общих чертах, как этот алгоритм функционирует.

В качестве основы для нашего примера используется база данных поставщиков, деталей и проектов. Для удобства приводится набор примерных значений для этой базы данных.

Таблица Поставщики (S)

Sno	Same	Status	City
1	Алмаз	20	Смоленск
2	Циклон	10	Владимир
3	Дельта	30	Владимир
4	Орион	20	Смоленск
5	Аргон	30	Ярославль

Таблица Детали (P) Pno Pname

Pno	Pname	Color	Weight	City
1	Гайка	Красный	12	Смоленск
2	Болт	Зеленый	17	Владимир
3	Винт	Синий	17	Рязань
4	Винт	Красный	14	Смоленск
5	Шайба	Синий	12	Владимир
6	Шпунт	Зеленый	19	Смоленск

Таблица Проекты (J)

Jno	Jname	City
1	Ангара	Владимир
2	Алтай	Рязань
3	Енисей	Ярославль
4	Амур	Ярославль

Jno	Jname	City
5	Памир	Смоленск
6	Чегет	Тверь
7	Эльбрус	Смоленск

Таблица Поставки (SPJ)

Sno	Pno	Jno	Qty
1	1	1	200
1	1	4	700
2	3	1	400
2	3	2	200
2	3	3	200
2	3	4	500
2	3	5	600
2	3	6	400
2	3	7	800
2	5	2	100
3	3	1	200
3	4	2	500
4	6	3	300
4	6	7	300
5	2	2	200
5	2	4	100
5	5	5	500
5	5	7	100
5	6	2	200
5	1	4	100
5	3	4	200
5	4	4	800
5	5	4	400
5	6	4	500

Рассмотрим запрос: "Получить имена и города поставщиков, обеспечивающих по крайней мере один проект в городе Ярославль с поставкой по крайней мере 50 штук каждой детали". Выражение исчисления для этого запроса следующее:

(SX.Name, SX.City) WHERE EXISTS JX FORALL PX EXISTS SPJX (JX.City = 'Ярославль' AND JX.Jno = SPJX.Jno AND PX.Pno = SPJX.Pno AND SX.Sno = SPJX.Sno AND SPJX.Qty >= 50)

Здесь SX, PX, JX и SPJX - переменные кортежей, берущие свои значения из отношений S, P, J и SPJ соответственно. Теперь покажем, как вычислить это выражение, чтобы добиться необходимого результата.

Шаг 1. Для каждой переменной кортежа выбираем ее область значений (т.е. набор всех значений для этой переменной) если это возможно. Выражение «выбираем, если возможно» подразумевает, что существует условие выборки, встроенное в фразу WHEPE, которую можно использовать, чтобы сразу исключить из рассмотрения некоторые кортежи. В нашем случае выбираются следующие наборы кортежей:

SX : Все кортежи отношения S 5 кортежей
PX : Все кортежи отношения P 6 кортежей
JX : Кортежи отношения J, в которых City = 'Ярославль' 2 кортежа SPJX : Кортежи отношения SPJ, в которых Qty >= 50 24 кортежа

Шаг 2. Строим декартово произведение диапазонов, выбранных на первом шаге. Получим ... Для экономии места таблица не приводится. Полное произведение содержит 5*6*2*24 = 1440 кортежей.

Шаг 3. Осуществляем выборку из произведения, построенного на втором шаге в соответствии с частью «условие соединения» фразы WHERE. В нашем примере эта часть следующая:

JX.Jno = SPJX.Jno AND PX.Pno = SPJX.Pno AND SX.Sno = SPJX.Sno AND

Поэтому из произведения исключаются кортежи, для которых значение Sno поставщика не равно значению Sno поставки, значение Pno детали не равно значению Pno поставки, значение Jno проекта не равно значению Jno поставки, после чего получаем подмножество декартова произведения, состоящее только из 10 кортежей.

Шаг 4. Применяем кванторы справа налево следующим образом:
- Для квантора «EXISTS RX» (где RX - переменная кортежа, принимающая значение на не котором отношении R) проецируем текущий промежуточный результат, чтобы исключить все атрибуты отношения R.
- Для квантора «FORALL RX» делим текущий промежуточный результат на отношение «выбранной области значений», соответствующее RX, которое было получено выше. При выполнении этой операции также будут исключены все атрибуты отношения R.

В нашем примере имеем следующие кванторы: EXISTS JX FORALL PX EXISTS SPJX Выполняем соответствующие операции.

- EXISTS SPJX. Проецируем, исключая атрибуты отношения SPJ (SPJ.Sno, SPJ.Pno, SPJ.Jno и SPJ.Qty). В результате получаем:

S no	Same	Status	City	P no	Pname	Color	Weight	City	Jno	Jname	City
1	Алмаз	20	Смоленск	1	Гайка	Красный	12	Смоленск	4	Амур	Ярославль
2	Циклон	10	Владимир	3	Винт	Синий	17	Рязань	3	Енисей	Ярославль
2	Циклон	10	Владимир	3	Винт	Синий	17	Рязань	4	Амур	Ярославль
4	Орион	20	Смоленск	6	Шпон	Красный	19	Смоленск	3	Енисей	Ярославль
5	Аргон	30	Ярославль	2	Болт	Зеленый	17	Владимир			Ярославль
5	Аргон	30	Ярославль	1	Гайка	Красный	12	Смоленск	4	Амур	Ярославль
5	Аргон	30	Ярославль	3	Винт	Синий	17	Рязань	4	Амур	Ярославль
5	Аргон	30	Ярославль	4	Винт	Красный	14	Смоленск	4	Амур	Ярославль
5	Аргон	30	Ярославль	5	Шайба	Синий	12	Владимир	4	Амур	Ярославль
5	Аргон	30	Ярославль	6	Шпунт	Красный	19	Смоленск	4	Амур	Ярославль

- FORALL PX. Делим на отношение P. В результате получаем: 17

Sno	Same	Status	City	Jno	Jname	City
5	Аргон	30	Ярославль	4	Амур	Ярославль

- EXISTS JX. Проецируем, исключая атрибуты отношения J (J.Jno, J.name и J.City). В результате получаем: Sname Status City

Sno	Same	Status	City
5	Аргон	30	Ярославль

Шаг 5. Проецируем результат шага 4 в соответствии со спецификациями в целевом списке элементов. В нашем примере целевым элементом списка будет: SX.Sname, SX.City. Следовательно, конечный результат таков:

Same	City
Аргон	Ярославль

Из сказанного выше следует, что начальное выражение исчисления семантически эквивалентно определенному вложенному алгебраическому выражению, а если быть более точным, то проекции от проекции деления проекции выборки из произведения четырех выборок (!). Этим завершаем пример. Конечно, можно намного улучшить алгоритм, хотя многие подробности скрыты в наших пояснениях; но вместе с тем, необходим адекватный пример, предлагающий общую идею.

Теперь можно объяснить одну из причин (и не только одну) определения Коддом ровно восьми алгебраических операторов. Эти восемь операторов обеспечивают соглашение целевого языка, как средства возможной реализации исчисления. Другими словами, для данного языка, такого как QUEL, который основывается на исчислении, одно из возможных применений заключается в том, чтобы можно было брать запрос в том виде, в каком он предоставляется пользователем (являющийся, по существу, просто выражением исчисления), и применять к нему алгоритм получения эквивалентного алгебраического выражения. Это алгебраическое выражение, конечно, содержит множество алгебраических операций, которые согласно определению по своей природе выполнимы. (Следующий шаг состоит в продолжении оптимизации этого алгебраического выражения.)

Также необходимо отметить, что восемь алгебраических операторов Кодда являются мерой оценки выразительной силы любого языка баз данных (существующего или предлагаемого). Обсудим этот вопрос подробнее.

Во-первых, язык называется реляционно полным, если он по своим возможностям, по крайней мере, не уступает реляционному исчислению, т.е. любое отношение, которое можно определить с помощью реляционного исчисления, также можно определить и с помощью некоторого выражения рассматриваемого языка. «Реляционно полный» - значит, не уступающий по возможностям алгебре, а не исчислению, но это то же самое. По сути, из существования алгоритма преобразования Кодда немедленно следует, что реляционная алгебра обладает реляционной полнотой.)

Реляционную полноту можно рассматривать как основную меру возможностей выборки и выразительной силы языков баз данных вообще. В частности, так как исчисление и алгебра - реляционно полные, они могут служить базисом для проектирования языков, не уступающих им по выразительности, без необходимости пересортировки для использования циклов - особенно важное замечание, если язык предназначается конечным пользователям, хотя оно также применимо, если язык предназначается прикладным программистам.

Далее, поскольку алгебра реляционно полная, то, чтобы доказать, что некоторый язык L обладает реляционной полнотой, достаточно показать, что в языке L есть аналоги всех восьми алгебраических операций (на самом деле достаточно по казать, что в нем есть аналоги пяти примитивных операций) и что операндами любой операции языка L могут быть любые выражения этого языка. Язык SQL - это пример языка, реляционную замкнутость которого можно показать таким способом. Язык QUEL – еще один такой пример. В действительности на практике часто проще показать, что в данном языке есть эквиваленты алгебраических операций, чем найти в нем эквиваленты выражений исчисления. Именно поэтому реляционная полнота обычно определяется в терминах алгебраических выражений, а не выражений исчисления.