Serdica
Bulgariacae mathematicae publicationes

Сердика
Българско математическо списание

НЕТОЧНОСТИ В МАТЕМАТИЧЕСКИХ ОСНОВАХ РЕЛЯЦИОННОЙ МОДЕЛИ БАЗЫ ДАННЫХ И ПОПЫТКА ИХ ИСПРАВЛЕНИЯ*

КАЗИМЕЖ СУБЕТА

В сообщении обсуждается широко известная реляционная модель базы данных. При попытке полной формализации этой модели наталкиваемся на две основные трудности: (1) в этой модели на одном уровне рассуждения выступают элементы языка теории отношений (напр., представления отношений, множеств, постоянных) и элементы метаязыка (названия отношений и множеств); (2) в реляционной модели вводится понятие „мультимножество", которое не является вполне формальным с математической точки зрения. В сообщении проводится корректировка реляционной модели, при которой в качестве математической основы вместо теории отношений используется формальная лингвистика.

1. Введение. Реляционная модель базы данных, основоположником которой является Кодд [1], привлекает широкое внимание многих специалистов в области баз данных. Главным достоинством реляционной модели является простота вводимых понятий, из-за которой модель доступна даже пользователю с небольшим опытом. При этой простоте она является вполне универсальной и в принципе может применяться при любых требованиях, касающихся информационного содержания базы данных.

В работах авторов, связанных с реляционной моделью (напр. [2]), часто подчеркивается прочность ее математических основ. В настоящей работе мы бы хотели показать, что, к сожалению, эта прочность является иногда сомнительной. По нашему мнению, теория отношений не является подходящим аппаратом для описания интуиций, предполагаемых в реляционной модели. Для этой цели мы предлагаем формализм математической лингвистики.

Другая цель, которую мы намерены достичь, касается освобождения реляционной модели от ненужного „мистицизма", в котором теория некоторой действительности считается сама действительностью. Этот „мистицизм" главным образом отражается в терминологии: реляционисты предлагают „n-орка" (tuple) вместо „запись", „отношение" вместо „файл", и т. д. Обычно в таких случаях употребляются фразы типа „математической моделью записи является „n-орка", или „файл является отношением", где четко различается действительность и теория. Это вызывает существенные методологические трудности: оказывается невозможным говорить о той же действительности в других математических терминах и тем самым ограничиваются возможности улучшения теории.

2. Неточности реляционной модели. 2.1. Недостатки различения между языком и метаязыком теории отношений. В любой математической теории, в том числе в теории отношений, предполагается

* Представлена на конференции по системам информационного обслуживания коллективов профессионально-связанных потребителей, 23—29 мая, 1977, Варна.

СЕРДИКА Българско математическо списание. Том 10, 1984, с. 142—147.

или явно вводится язык для описания объектов, которыми занимается теория. В теории отношений такой язык должен давать возможность записи множеств и отношений. Итак, множество можем записать перечисляя названия его элементов в фигурных скобках, напр. {желтый, зеленый, синий}, или {11, 12, 13, 14, 15}. Отношения, в свою очередь, можем записать как множество, элементы которого являются последовательностями, заключенными в круглые скобки, напр. {(11, желтый), (12, желтый), (13, зеленый)}. Однако такая запись является крайне неудобной, в случае более мощных множеств. Поэтому для того чтобы говорить о множествах, отношениях и т. д., не перечисляя их элементов, вводятся вспомогательные названия, напр. НОМЕР, ЦВЕТ, ВЕС. Эти названия являются элементами метаязыка, их определения тоже вводятся в метаязыке. Например: „ЦВЕТ={желтый, зеленый, синий}“, „ВЕС — множество положительных действительных чисел“. Таким образом мы можем (в метаязыке) ввести определение отношения ДЕТАЛЬ в виде

$$\text{ДЕТАЛЬ} \subseteq \text{НОМЕР} \times \text{ЦВЕТ} \times \text{ВЕС}.$$

Все названия в этом определении принадлежат метаязыку; в действительности любое название обозначает некоторое множество. Если после определения, отношение ДЕТАЛЬ сможем записать в языке, то в принципе можем забыть с названиях множеств, участвующих в определении. В реляционной модели файл является явно записанным отношением и в этой модели нет места для названий множеств. Поэтому мы никак не можем согласиться со способом употребления этих метаязыковых названий в так называемых „подъязыках“ базы данных. В качестве примера обсудим „алгебру отношений“ [3]. В этой алгебре вводится ряд операций. Одной из них является „проекция“ отношения на множество. Мы хотим определить эту операцию в обычном математическом смысле как функцию от двух аргументов. Первым аргументом этой функции является отношение, вторым — название множества — области определения отношения. Но если первый аргумент запишем на языке теории отношений в явном виде, тогда названия множеств, на которых это отношение определено, не будут выступать там вообще. В таком случае как мы сможем „связать“ второй аргумент с первым для получения правильного результата? Очевидно, что это невозможно без вспомогательных математических понятий, передвигающих названия множеств с метаязыкового на языковой уровень.

2.2. Недостатки различения между множеством и представлением множества. Согласно реляционной модели, над отношениями могут проводиться некоторые операции, в результате которых мы получаем новые отношения. Возьмем под внимание операцию проекции отношения на множество, пренебрегая неточностями в ее определении. Будем считать, что результат этой операции является для нас очевидным: получим некоторое одноместное отношение, т. е. множество элементов. К таким одноместным отношениям применяются так называемые „агрегированные операции“, напр., арифметическая сумма элементов множества, число элементов множества и т. д. Остановимся на операции СУММА и рассмотрим следующий пример: пусть файл некоторого предприятия имеет вид отношения РАБОЧИЕ:

РАБОЧИЕ НОМЕР	ФАМИЛИЯ	ЗАРПЛАТА
603	Браун	250
695	Смит	350
769	Джонс	250
861	Тейлор	200

Надо вычислить сумму зарплат всех рабочих. Мы должны принять следующий порядок операции: в первом шагу осуществим проекцию отношения РАБОЧИЕ на область ЗАРПЛАТА. Получим множество {250, 350, 250, 200}. Сейчас к этому множеству мы должны применить операцию СУММА, получая в результате величину 1050. Однако, к сожалению, в полученном множестве элемент 250 выступает дважды. По всем соглашениям, существующим в теории множеств {250, 350, 250, 200}={250, 350, 200} — множество содержит вместо четырех только три элемента. Применяя операцию СУММА к такому множеству, получим ошибочный результат 800. Вот и причина, для которой необходимо ввести понятие „мультимножества", т. е. такого множества, в котором „сохраняются повторяющиеся элементы".

В этом примере мы показали приблизительный ход рассуждений реляционистов, ведущий к понятию „мультимножества" [4, 5]. Однако в этих рассуждениях существуют некоторые недоразумения. Как известно, множество является неопределяемым понятием. Но представление множества в виде выражения некоторого языка — это совершенно другое дело. Для этой цели существует несколько методов. Любой из этих методов основан на том, что каждый элемент множества приобретает свое единственное название, которое употребляется в языковом представлении множества. Например, выражение {1, 3, 7} является представлением трехэлементного множества некоторых абстрактных существований (чисел); эти существования приобрели соответственно названия 1, 3, 7. Вполне естественно, что существует соглашение, устанавливающее эквивалентность представлений множества без учета количества повторяющихся названий элементов. Например, представление множества {7, 3, 7, 1, 1, 7} эквивалентно представлению {1, 3, 7}.

Подводя итог, нет смысла говорить, что в множестве элементы не могут повторяться потому, что неизвестно, что это значит „повторяться"— иметь те же названия в представлении, или что-нибудь другое. Так же не имеет смысла высказывание „элементы в множестве могут повторяться".

Мы можем только сказать, что не целесообразно в представлении множества употреблять несколько идентичных названий, поскольку существует более краткое представление.

Итак, мы выяснили главную причину недоразумения, ведущего к понятию „мультимножества": оказалось, что интуиция создателей реляционной модели была связана с представлением множества в виде выражения на некотором языке, а не с самим множеством.

3. Путь исправления реляционной модели. Что касается первой неточности, решение кажется очевидным: необходимо названия отношений и множеств как-то включить в язык модели. Это можо сделать многими способами, однако нет никаких причин, препятствующих воспользоваться приемом, существующим в теории структур данных [6]. Итак, мы можем предложить, чтобы любая запись (n-орка) была парой: (название записи,

содержание записи будет множеством пар вида (название элемента, значение элемента). То что мы назвали „названием записи" в реляционной модели выступало как название отношения; название элемента выступало как название множества — области определения отношения. Таким образом любое содержимое базы данных можем считать множеством (или последовательностью) записей.

Исправление другой неточности начнем с попытки точного определения „агрегированных операций". В качестве примера выберем операцию СУММА. Наиболее естественным является определение этой операции в виде отображения СУММА: $R^* \to R$, где R — множество действительных чисел, R^* — множество всех последовательностей, которые можем составить из элементов множества R.

Таким образом имеем возможность суммирования идентичных чисел.

Но, с другой стороны, аргумент операции СУММА может быть проекцией „отношения" на некоторую „область". Поскольку, по нашему мнению, необходимо, чтобы эта проекция была последовательностью, потому естественно считать, что целое „отношение" должно быть последовательностью. (Конечно, это уже не будет „отношение".)

Сейчас нетрудно заметить следующее правило: любое „отношение" оказалось последовательностью, любой элемент „отношения" также является последовательностью. В итоге, целое „отношение" можем рассматривать как последовательность некоторых символов. Мы считаем, что лучшим аппаратом, применяемым для определения и анализа свойств последовательностей символов, является математическая лингвистика

4. Лингвистическая формулировка реляционной модели. 4.1. Содержимое базы данных. Пусть даны два множества: N — множество элементарных значений. Пусть также будут даны два знака „(„ ")", не принадлежащие $N \cup V$.

Определение. *Элементарной данной будем называть последовательность $n(v)$, где $n \in N$, $v \in V$. Обозначим через D множество всех элементарных данных. Элементарная данная является аналогом того, что реляционисты называют „элемент n-орки".*

Примеры: НОМЕР (603), ФАМИЛИЯ (Смит), ЗАРПЛАТА (250).

Определение. *Записью будем называть любую последовательность вида $n(u)$, где $n \in N$, а u является непустой последовательностью элементарных данных, в которой названия данных не повторяются. Обозначим через R множество всех записей. Запись является аналогом того, что реляционисты назвали „n-орка".*

Пример: РАБОТНИК (НОМЕР (603) ФАМИЛИЯ (Смис) ЗАРПЛАТА (250)).

Определение. *Содержимое базы данных является последовательностью записей. Обозначим через C множество всех содержимых базы данных.*

4.3. Язык описания данных. В реляционной модели выражение языка описания данных названо „схема". Схема служит для записи некоторого общего взгляда, касающегося содержимого базы данных.

В лингвистической терминологии выражения языка описания данных являются грамматиками. Любая из этих грамматик определяет множество допустимых содержаний базы данных. Действительное содержание является

элементом этого множества. Обозначим через DC язык описания данных. С синтаксической точки зрения DC является множеством цепочек над алфавитом

$$N \cup V \cup P \cup \{\text{“(“, “)“, “\{“ , “\}“, “}\vee\text{“}\},$$

где N, V — имеют предыдущий смысл, а P — множество названий множеств элементарных значений, напр., $CHARACTER\ 5 \in P$, $NUMERIC\ 4 \in P$, и т. д.

Будем считать, что $(N \cup V) \cap P = \emptyset$. Любой из пяти специальных символов не принадлежит $N \cup V \cup P$.

С семантической точки зрения любой элемент DC определяет некоторое подмножество множества C. Примем следующее соглашение; если X является некоторой синтаксической конструкцией, то его семантику будем обозначать $|X|$. По определению, $x \in V \Rightarrow |x| = \{x\}$; $x \in P \Rightarrow |x| \subseteq V$.

Определение множества DV (описаний элементарных значений):

$$u \in V \cup P \Rightarrow u \in DV$$

$$w \in DV \wedge u \in V \cup P \Rightarrow \mathbf{w} \vee \mathbf{u} \in DV$$

$$|w \vee u| = |w| \cup |u|.$$

Жирным шрифтом мы будем выделять цепочки символов, чтобы обозначить, что речь идет об одном элементе, являющимся некоторой последовательностью.

Определение множества DD (описаний элементарных данных)

$$n \in N \wedge w \in DV \Rightarrow \mathbf{n(w)} \in DD$$

$$|n(w)| = \{\mathbf{n(u)}:\ u \in |w|\}.$$

Определение множества $\boldsymbol{DD}^+$. *$\boldsymbol{DD}^+$ обозначает множество всех непустых цепочек, составленных из элементов множества DD, при условии, что в любой из этих цепочек любое название $n \in N$ выступает не больше, чем один раз. Конечно, $DD \subseteq \boldsymbol{DD}^+$ и семантика общих элементов в этих множествах совпадает. Для элементов $\boldsymbol{DD}^+$ длины больше, чем 1, определение семантики следующее:*

$$a \in \boldsymbol{DD}^+ \vee b \in DD \Rightarrow |ab| = \{\boldsymbol{\alpha\beta}:\ \alpha \in |a| \wedge \beta \in |b|\}.$$

Определение множества DR (описаний записей)

$$n \in N \wedge d \in \boldsymbol{DD}^+ \Rightarrow \mathbf{n(d)} \in DR$$

$$|n(d)| = \{\mathbf{n(e)}:\ e \in |d|\}.$$

Определение множества DF (описаний файлов)

$$r \in DR \quad \{\boldsymbol{r}\} \in DF$$

$$|\{r\}| = |r|^*,$$

где, как выше, $|r|^*$ обозначает множество всех последовательностей, которые можно составить из элементов множества $|r|$.

Семантика элемента множества DF является аналогом того, что реляционисты назвали отношением.

Определение множества DC

$$DC = \boldsymbol{DF}^{+},$$

где, как выше, $\boldsymbol{DF}^{+}$ *обозначает множество всех непустых цепочек, составленных из элементов множества* DF*, при условии, что в этих цепочках названия записей не повторяются.*

Для элементов DC, описывающих более чем один файл, определение семантики следующее:

$$c \in DC \wedge f \in DF \Rightarrow |cf| = \{\boldsymbol{\gamma\psi} : \gamma \in |c| \wedge \psi \in |f|\}.$$

Пример выражения языка DC.

```
{РАБОЧИЙ(
   НОМЕР (NUMERIC 5)
   ФАМИЛИЯ (CHARACTER 15)
   ИМЯ (CHARACTER 15)
   ПОЛ (М∨Ж)
   ДЕВИЧЬЯ ФАМИЛИЯ (CHARACTER 15∨NULL)
   ЗАРПЛАТА (NUMERIC 3)
НОМЕР ОТДЕЛЕНИЯ (CHARACTER 6))}
  {ОТДЕЛЕНИЕ(
     НОМЕР (CHARACTER 6)
     НАЗВАНИЕ (CHARACTER 15)
     НОМЕР ЗАВЕДУЮЩЕГО (NUMERIC 5)
     БЮДЖЕТ (NUMERIC 6))}
```

ЛИТЕРАТУРА

1. E. F. Codd. A Relational Model of Data for Large Shared Data Banks. *Comm. ACM*, **13**, 1970, 377—387.
2. C. J. Date. An Introduction to Database Systems. Addison-Wesley, 1975.
3. E. F. Codd. Relational Completeness of Data Base Sublanguages. *Data Base Systems* (ed. R. Rustin), Prentice-Hall, 1972.
4. G. D. Held, M. Stonebraker, E. Wong. INGRES — A Relational Data Base System. AFIPS Conf. Proc., **44**, 1975, 409—416.
5. M. M. Zloof. Query by Example. *AFIPS Conf. Proc.* **44**, 1975, 431—437.
6. D. E. Knuth. The Art of Computer Programming, vol. 1, Addison-Wesley, 1968.

Институт основ вычислительной техники
Польская академия наук, Варшава, Польша

Поступила 11. 9. 1977