

Применение лямбда-исчисления в реляционных базах данных

Чалая Л.Э, Попаденко П.Ю.

Харьковский национальный университет радиоэлектроники
(61166, Харьков, пр. Ленина,14, каф. ИИ, тел. (057) 702-13-37),
E-mail : kovalivnich@yahoo.com, ppropadenko@gmail.com

This research is dedicated to the study of lambda-calculus in connection with relational databases. In this work overviewed main disadvantages of traditional database type model. It is suggested to use untyped form of view on the data in our traditional databases. Also discussed advantages and disadvantages of untyped data stored in databases.

ТИПИЗАЦИЯ ДАННЫХ В РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ

В современных компьютерных системах для хранения информации широко используются реляционные базы данных (РБД). К РБД, как правило, предъявляется требование однородности данных, в соответствии с которым данные, находящиеся в одном столбце, обязательно должны иметь одинаковый тип. Однако при работе с большими РБД часто возникают проблемы с типизацией данных, поступающих из различных источников.

Строгая типизация данных всегда обеспечивает системе определенную защиту, гарантируя их корректность. Это, несомненно, является преимуществом СУБД с типизированными данными. Следует, однако, отметить, что данные, которые характеризуются одинаковой смысловой наполненностью, но по каким-то причинам имеющие различный тип, будут конфликтовать друг с другом. Необходимость постоянного устранения таких конфликтов вызывает идею отхода от строгой типизации данных в РБД.

Такая идея не является новой – она успешно используется в том или ином виде практически во всех функциональных языках программирования.

Переход к такой схеме в РБД возможен, однако при этом возникают трудности, связанные с приведением реляционных СУБД к виду, в котором возможно их описание с применением лямбда-исчислений.

Рассмотрим возможность такого описания в терминах функциональных языков программирования. В современных реляционных базах данных данные в таблицах связаны с помощью определенных функциональных зависимостей. Другими словами, значения кортежа на некотором множестве атрибутов единственным образом определяют значения на другом множестве атрибутов:

$$\text{ЕСЛИ}(x \cap y) \text{ТО}(z). \quad (1)$$

Используя форму записи (1), функциональные зависимости можно представить в форме, легко применимой для функциональных языков программирования.

Лямбда-исчисление и типизация

Лямбда-исчисление представляет собой модель абстрактного вычисления, которая рассматривает функции как правила. Она позволяет перейти от традиционной задачи «КАК сделать что-либо» до интуитивного «ЧТО делать». При применении лямбда-исчисления к РБД нет необходимости постоянно проверять данные на соответствие какому-либо типу. Задав определенные правила, здесь можно манипулировать разнородными данными и при этом получать корректные результаты.

Рассмотрим, например, поведение базы данных при вычитании одного значения из другого:

23/02/2012-"22-02-2011". (2)

Очевидно, что: вторая часть выражения (2) является строкой. Для обычной базы с типизированными данными реализация такого выражения неразрешима. И даже если бы обе части выражения (2) имели одинаковый тип, то возникали бы неизбежные проблемы, связанные различным форматом и недопустимостью операции вычитания дат в некоторых СУБД.

ВВЕДЕНИЕ ЛЯМБДА-ИСЧИСЛЕНИЯ В КОНЦЕПЦИЮ БАЗ ДАННЫХ

Лямбда-исчисление в общем случае может рассматриваться как семейство прототипных языков программирования. Их основная особенность состоит в том, что они являются языками высших порядков. Тем самым обеспечивается систематический подход к исследованию операторов, аргументами которых могут быть другие операторы, а значением также может быть оператор. Языки в этом семействе являются функциональными, поскольку они основаны на представлении о функции или операторе, включая функциональную аппликацию и функциональную абстракцию.

Лямбда-выражения операций над данными будут иметь такой вид:

$$z = \lambda x. \lambda y. xy \quad (3)$$

где x , y – аргументы; z – полученное значение. Знак λ указывает на операцию, определяющую характер преобразования x и y . При этом неизбежно возникают нетривиальные вопросы выбора класса λ -функции и возможности их применения в ситуациях, когда данные поступают в РБД в разнородном виде. Компромиссным вариантом использования λ -функций в СУБД является типизированное λ -исчисление, это типовый формализм, использующий символ абстракции « λ » для записи выражений, обозначающих безымянные функции. Типовые λ -исчисления являются фундаментальными примитивными языками программирования, которые обеспечивают основу типовым языкам функционального программирования — аппликативным языкам, — среди которых ML и Haskell, а также типовым императивным языкам программирования. λ -исчисление с типами служит основой для разработки новых систем типизации для языков программирования, поскольку именно средствами типов и зависимостей между ними выражаются желаемые свойства программ.

В данном докладе рассмотрены самостоятельные вычислительные блоки (функции, процедуры, методы) для разработки РБД со специализированными типовыми λ -выражениями применительно к базам данных социальных сетей.

Выводы

Типизацию данных в реляционных СУБД можно существенно упростить, применяя теорию лямбда-исчислений. Теоретически можно хранить полностью разнородные данные, не нарушая целостности системы.

Перспективным является преобразование данных СУБД под руководством администратора БД с целью создания утилиты, которая будет воспринимать данные системы без типов и применять свои методы для решения специальных задач пользователя.

ЛИТЕРАТУРА

- [1] Бенджамин Пирс «Типы в языках программирования»: - Россия: Добросвет, 2012 – 655с.
Achim Jung “A short introduction to the Lambda Calculus” - March 18, 2004
Lambda Calculus on Safalra’s Website <http://safalra.com/science/lambda-calculus/>