Содержание:

Введение

Эдгар Франк «Тед» Кодд — британский учёный, работы которого заложили основы теории реляционных баз данных.

Работая в компании IBM, он создал реляционную модель данных. Он также внёс существенный вклад в другие области информатики.

Краткая биография Эдгара Франка

Родился в Портланде (Дорсет) в Англии. Его отец был производителем кожи, а его мама была учительницей. Обучался математике и химии в Оксфордском университете (Exeter College).

Во время Второй мировой войны служил пилотом в военно-воздушных силах.

В 1948 переехал в Нью-Йорк, чтобы работать в IBM как математик-программист.

В 1953, из-за преследований со стороны сенатора Джозефа Маккарти (Joseph McCarthy), Кодд переехал в Оттаву (Канада).

В 60-х — 70-х годах он работал над своими теориями хранения данных. В 1970 издал работу «A Relational Model of Data for Large Shared Data Banks», которая считается первой работой по реляционной модели данных.

В начале 80-х реляционная модель начала входить в моду. Борясь с недобросовестными поставщиками СУБД, которые утверждали, что их устаревшие продукты поддерживают реляционную технологию, Кодд опубликовал «12 правил Кодда», описывающие, что должна содержать реляционная СУБД.

Его борьба коснулась языка SQL, который Кодд считал неправильной реализацией теории. Это делало его положение в IBM достаточно тяжелым, так как та

поставляла продукты, основанные на SQL. Он покинул IBM и организовал вместе с Кристофером Дейтом и несколькими другими людьми собственную консалтинговую компанию.

12 правил Кодда - 13 правил (в данном случае исчисление начинается с 0), которым должна удовлетворять каждая система управления реляционными базами данных.

Предложены английским математиком Эдгаром Коддом (Edgar Codd) в 1985 году в статьях в журнале *ComputerWorld*.

В действительности правила столь строги, что все популярные так называемые «реляционные» СУБД не соответствуют многим критериям.

12 правил Кодда

Правило 0: Основное правило (*Foundation Rule*):

Система, которая рекламируется или позиционируется как реляционная система управления базами данных, должна быть способна управлять базами данных, используя исключительно свои реляционные возможности.

История

В 1974 году компания IBM начала исследовательский проект по разработке РСУБД, получивший название System R. Её первым коммерческий продуктом был IBM SQL/DS, выпущенный в 1982 году.

Однако первой коммерчески успешной РСУБД стала Oracle, выпущенная в 1979 году компанией Relational Software, которая впоследствии была переименована в Oracle Corporation.

В 1970-е годы, когда уже были получены почти все основные теоретические результаты и даже существовали первые прототипы реляционных СУБД, многие авторитетные специалисты отрицали возможность добиться эффективной реализации таких систем. Однако преимущества реляционного подхода и развитие методов и алгоритмов организации и управления реляционными базами данных привели к тому, что к концу 1980-х годов реляционные системы заняли на мировом рынке СУБД доминирующее положение.

В связи с резким ростом популярности РСУБД в 1980-х годах многие компании стали позиционировать свои СУБД как «реляционные» в рекламных целях, иногда не имея для этого достаточных оснований, вследствие чего автор реляционной модели данных Эдгар Кодд в 1985 году опубликовал свои знаменитые «12 правил Кодда», которым должна удовлетворять каждая РСУБД.

Правило 1: Информационное правило (*The Information Rule*):

Вся информация в реляционной базе данных на логическом уровне должна быть явно представлена единственным способом: значениями в таблицах.

Отношение — фундаментальное понятие реляционной модели данных. По этой причине модель и называется *реляционной*.

Пусть дана совокупность типов данных T_1, T_2, ..., T_n, называемых также доменами, не обязательно различных. Тогда n-арным отношением R, или отношением R степени n называют подмножество декартова произведения множеств T_1, T_2, ..., T_n.

Отношение R состоит из *заголовка* (*схемы*) и *тела*. Заголовок представляет собой множество *атрибутов* (именованных вхождений домена в заголовок отношения), а тело — множество *кортежей*, соответствующих заголовку. Более строго:

- Заголовок (или схема) H отношения R — конечное множество упорядоченных пар вида (A_i, T_i), где A_i — *имя атрибута*, а T_i — *имя типа* (домена), i=1,..., n. По определению требуется, чтобы все имена атрибутов в заголовке отношения были различными (уникальными).
- Тело B отношения R — множество кортежей t. Кортеж t, соответствующий заголовку H, — множество упорядоченных триплетов (троек) вида <A_i, T_i, v_i>, по одному такому триплету для каждого атрибута в H, где v_i — допустимое значение типа (домена) T_i. Так как имена атрибутов уникальны, то указание домена в кортеже обычно излишне. Поэтому кортеж t, соответствующий заголовку H, нередко определяют как множество пар (A_i, v_i).

Правило 2: Гарантированный доступ к данным (*Guaranteed Access Rule*):

В реляционной базе данных каждое отдельное (атомарное) значение данных должно быть логически доступно с помощью комбинации имени таблицы, значения первичного ключа и имени столбца.

Правило 3: Систематическая поддержка отсутствующих значений (*Systematic Treatment of Null Values*):

Неизвестные, или отсутствующие значения NULL, отличные от любого известного значения, должны поддерживаться для всех типов данных при выполнении любых операций. Например, для числовых данных неизвестные значения не должны рассматриваться как нули, а для символьных данных — как пустые строки.

NULL в Системах управления базами данных (СУБД) — специальное значение (псевдозначение), которое может быть записано в поле таблицы базы данных (БД). NULL соответствует понятию «пустое поле», то есть «поле, не содержащее никакого значения». Введено для того, чтобы различать в полях БД пустые (визуально не отображаемые) значения (например, строку нулевой длины) и отсутствующие значения (когда в поле не записано вообще никакого значения, даже пустого).

Правило 4: Доступ к словарю данных в терминах реляционной модели (*Active On-Line Catalog Based on the Relational Model*):

Словарь данных должен сохраняться в форме реляционных таблиц, и СУБД должна поддерживать доступ к нему при помощи стандартных языковых средств, тех же самых, которые используются для работы с реляционными таблицами, содержащими пользовательские данные.

Словарь данных, описанный в *Словаре вычислений от IBM* (*IBM Dictionary of Computing*) как «центральное хранилище информации о данных, такой как значение, взаимосвязи с другими данными, их источник, применение и формат.» Термин может иметь одно из близких по смыслу значений, относясь к базам данных и СУБД:

Правило 5: Полнота подмножества языка (*Comprehensive Data Sublanguage Rule*):

Система управления реляционными базами данных должна поддерживать хотя бы один реляционный язык, который

(а) имеет линейный синтаксис,

(б) может использоваться как интерактивно, так и в прикладных программах,

(в) поддерживает операции определения данных, определения представлений, манипулирования данными (интерактивные и программные), ограничители целостности, управления доступом и операции управления транзакциями (begin, commit и rollback).

Правило 6: Возможность изменения представлений (*View Updating Rule*):

Каждое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции выборки, вставки, изменения и удаления данных.

Правило 7: Наличие высокоуровневых операций управления данными (*High-Level Insert, Update, and Delete*):

Операции вставки, изменения и удаления данных должны поддерживаться не только по отношению к одной строке реляционной таблицы, но и по отношению к любому множеству строк.

Правило 8: Физическая независимость данных (*Physical Data Independence*):

Приложения не должны зависеть от используемых способов хранения данных на носителях, от аппаратного обеспечения компьютеров, на которых находится реляционная база данных.

Правило 9: Логическая независимость данных (*Logical Data Independence*):

Представление данных в приложении не должно зависеть от структуры реляционных таблиц. Если в процессе нормализации одна реляционная таблица разделяется на две, представление должно обеспечить объединение этих данных, чтобы изменение структуры реляционных таблиц не сказывалось на работе приложений.

Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Правило 10: Независимость контроля целостности (*Integrity Independence*):

Вся информация, необходимая для поддержания целостности, должна находиться в словаре данных. Язык для работы с данными должен выполнять проверку входных данных и автоматически поддерживать целостность данных.

Правило 11: Независимость от расположения (*Distribution Independence*):

База данных может быть распределённой, может находиться на нескольких компьютерах, и это не должно оказывать влияния на приложения. Перенос базы данных на другой компьютер не должен оказывать влияния на приложения.

Правило 12: Согласование языковых уровней (*The Nonsubversion Rule*):

Если используется низкоуровневый язык доступа к данным, он не должен игнорировать правила безопасности и правила целостности, которые поддерживаются языком более высокого уровня.

Список литературы

https://habr.com/company/towave/blog/158755/

http://bourabai.ru/dbt/dbms/12rules.htm

http://db4.sbras.ru/elbib/data/show_page.phtml?77+1317