Содержание:

Введение:

Реляционные системы не сразу стали широко известными. В то время, как основные теоретические результаты, были получены еще в 70-х годах и тогда же появились первые прототипы реляционных СУБД - долгое время считалось невозможным добиться реализации подобных систем. Однако, постепенное накопление методов и алгоритмов организации реляционных баз данных и управления ими привели к тому, что примерно в середине 80-х годов реляционные системы практически вытеснили с мирового рынка ранние СУБД.

Реляционная модель данных основывается на математических принципах, вытекающих непосредственно из теории множеств и логики предикатов. Эти принципы впервые были применены в области моделирования данных в конце 60-х годов, доктором Е.Ф. Коддом, который в то время работал в IBM (впервые опубликованы - в 1970 году).

Техническая статья "Реляционная модель данных для больших разделяемых банков данных" доктора Е.Ф. Кодда - является родоначальницей современной теории реляционных БД. Доктор Кодд определил 12 правил реляционной модели, которые называют 12 правилами Кодда.

12 правил Кодда:

Реляционная СУБД должна полностью управлять БД через ее реляционные возможности.

1.

Правило информации.

Вся информация в базе данных должна быть представлена исключительно на логическом уровне и только одним способом - в виде значений, содержащихся в таблицах. Фактически это неформальное определение реляционной БД.

1.

Правило гарантированного доступа.

Логический доступ ко всем и каждому элементу данных в реляционной базе данных должен обеспечиваться путём использования комбинации имени таблицы, первичного ключа и имени столбца. Имя таблицы позволяет найти требуемую таблицу. Имя столбца позволяет найти требуемый столбец. А первичный ключ позволяет найти строку, содержащую искомый элемент данных. Данное правило указывает на роль первичного ключа при поиске информации в базе данных.

1.

Правило поддержки недействительных значений.

В настоящей реляционной БД должна быть реализована поддержка недействительных значений, которые отличаются от строки символов нулевой длины, строки пробельных символов от нуля или любого другого числа и используются для представления отсутствующих данных независимо от типа этих данных. Третье правило требует, чтобы отсутствующие данные можно было представить с помощью недействительных значений (NULL).

1.

Правило динамического каталога, основанного на реляционной модели.

Описание базы данных на логическом уровне должно быть представлено в том же виде, что и основные данные, чтобы пользователи, обладающие соответствующими правами, могли работать с ним с помощью того же реляционного языка, который они применяют для работы с основными данными. Приведённое правило гласит,

что реляционная база данных должна сама себя описывать. Другими словами, БД должна содержать набор системных таблиц, описывающих структуру самой базы данных.

1.

Правило исчерпывающего подъязыка данных.

Реляционная система может поддерживать различные языки и режимы взаимодействия с пользователем (например, режим вопросов и ответов). Однако должен существовать по крайней мере один язык, операторы которого можно представить в виде строк символов в соответствии с некоторым четко определенным синтаксисом и который в полной мере поддерживает следующие элементы:

- *определение данных;*

- *определение представлений;*

- *обработку данных (интерактивную и программную);*

- *условия целостности;*

- *идентификация прав доступа;*

- *границы транзакций (начало, завершение и отмена).*

Это же правило требует, чтобы СУБД использовала язык реляционной БД. Такой язык должен поддерживать все основные функции СУБД - создание базы данных, чтение и ввод данных, реализацию защиты БД и так далее.

1.

Правило обновления представлений.

Все представления, которые теоретически можно обновить, должны быть доступны для обновления. Шестое правило касается представлений, которые являются виртуальными таблицами, позволяющими показывать различным пользователям различные фрагменты структуры БД.

1.

Правило добавления, обновления и удаления.

Возможность работать с отношением как с одним операндом должна существовать не только при чтении данных, но и при добавлении, обновлении и удалении данных. Это правило акцентирует внимание на том, что БД по своей природе ориентированы на множества. Оно требует, чтобы операции добавления, удаления и обновления можно было выполнять над множествами строк.

1.

Правило независимости физических данных.

Прикладные программы и утилиты для работы с данными должны на логическом уровне оставаться нетронутыми при любых изменениях способов хранения данных или методов доступа к ним.

1.

Правило независимости логических данных.

Прикладные программы и утилиты для работы с данными должны на логическом уровне оставаться нетронутыми при внесении в базовые таблицы любых изменений, которые теоретически позволяют сохранить нетронутыми содержащиеся в этих таблицах данные. Восьмое и девятое правила означают отделение пользователя и прикладной программы от низкоуровневой реализации БД.

1.

Правило независимости условий целостности.

Должна существовать возможность определить условия целостности, специфические для конкретной реляционной базы данных, на подъязыке реляционной базы данных и хранить их в каталоге, а не в прикладной программе. Приведённое правило гласит, что язык БД должен поддерживать ограничительные

условия, налагаемые на вводимые данные и действия, которые могут быть выполнены над данными.

1.

Правило независимости распространения.

Реляционная СУБД не должна зависеть от потребностей конкретного клиента. То есть, язык базы данных должен обеспечивать возможность работы с распределенными данными, расположенными на других компьютерных системах.

1.

Правило единственности.

Если в реляционной системе есть низкоуровневой язык (обрабатывающий одну запись за один раз), то должна отсутствовать возможность использования его для того, чтобы обойти правила и условия целостности, выраженные на реляционном языке высокого уровня (обрабатывающем несколько записей за один раз). Ну и наконец, двенадцатое правило предотвращает использование других возможностей для работы с базой данных помимо языка базы данных, поскольку это может нарушить ее целостность.

Кодд предложил применение реляционной алгебры для расчленения данных в связанные наборы. Он организовал свою систему БД вокруг концепции, основанной на наборах данных. В его реляционной модели данные разбиваются на наборы, которые составляют табличную структуру. Эта структура таблиц состоит из индивидуальных элементов данных, которые называются полями. Одиночный набор или же группа полей известна, как запись.

Вывод:

Используя эти правила, доктор Е. Ф. Кодд смог сделать современные БД намного удобнее, проще, надёжнее и качественнее. Зная принципы работы БД и используя эти правила с умом – любой человек сможет работать, не напрягаясь и понимать базу данных, с которой он имеет дело. Без 12-ти правил Кодда, современные БД были бы намного трудоёмкими и на взаимодействие с ними уходило бы больше сил

и времени. Кодду нужно отдать должное, он в каком-то роде изменил историю в информационной сфере.

Источники информации:

https://www.intuit.ru/studies/courses/93/93/lecture/28077?page=3

http://www.interface.ru/home.asp?artId=16556

https://ru.wikipedia.org/wiki/12_правил_Кодда