

## Содержание:

image not found or type unknown



## Введение

**12 правил Кодда** (англ. *Codd's 12 rules*) — 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.

Теперь надо сказать немного о создателе этих правил и о том почему он их создал.

**Эдгар Франк «Тед» Кодд** (англ. *Edgar Frank Codd*; 23 августа 1923 — 18 апреля 2003) — 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.

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 1963 он вернулся в США и получил докторскую степень по информатике и вычислительной технике в Университете Мичигана (University of Michigan, Ann Arbor). В 1965 он переехал в Сан-Хосе (Калифорния), чтобы работать в Альмаденском Исследовательском Центре IBM.

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.

Кодд продолжил разрабатывать и расширять реляционную модель. В его честь названа одна из нормальных форм (нормальная форма Бойса — Кодда).

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.

Кодд ввёл в оборот термин OLAP и написал 12 законов аналитической обработки данных. Он также занимался клеточными автоматами.

В 1976 Кодд получил почетное звание IBM Fellow. В 1981 он получил премию Тьюринга.

## Правила 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.

### **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.

### **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.

**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.

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

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

1. имеет линейный синтаксис,
2. может использоваться как интерактивно, так и в прикладных программах,
3. 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.

**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.

## Заключение

Кроме правил Кодд предложил применение реляционной алгебры в СУБД, для расчленения данных в связанные наборы. Он организовал свою систему БД вокруг концепции, основанной на наборах данных. Все эти правила Кодда должны

соблюдаться при построении СУБД.

## **Список литературы**

<https://www.intuit.ru/>

<https://bourabai.ru/>

<https://coderlessons.com/>