

Правила Кодда (Codd's rules)

13 правил, которым должна удовлетворять каждая система управления реляционными базами данных.

Предложены английским математиком Эдгаром Коддом в 1985 году.

Правила столь строги, что все популярные «реляционные» СУБД не соответствуют многим критериям.

№	Правило	Описание
1	Основное правило (Foundation Rule)	Реляционная СУБД должна быть способна полностью управлять базой данных, используя связи между данными. Чтобы быть реляционной системой управления базами данных (СУБД), система должна использовать исключительно свои реляционные возможности для управления базой данных.
2	Явное представление данных (The Information Rule)	Информация должна быть представлена в виде данных, хранящихся в ячейках. Данные, хранящиеся в ячейках, должны быть <u>атомарны</u> . Порядок строк в реляционной таблице не должен влиять на смысл данных.
3	Гарантированный доступ к данным (Guaranteed Access Rule)	Доступ к данным должен быть свободен от двусмысленности. К каждому элементу данных должен быть гарантирован доступ с помощью комбинации имени таблицы, первичного ключа строки и имени столбца.

№	Правило	Описание
4	<p>Полная обработка неизвестных значений (Systematic Treatment of Null Values)</p>	<p>Неизвестные значения <u>NULL</u>, 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.</p>
5	<p>Доступ к словарю данных в терминах реляционной модели (Active On-Line Catalog Based on the Relational Model)</p>	<p>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.</p>
6	<p>Полнота подмножества языка (Comprehensive Data Sublanguage Rule)</p>	<p>Система управления реляционными базами данных должна поддерживать хотя бы один реляционный язык, который имеет линейный синтаксис, может использоваться как интерактивно, так и в прикладных программах, поддерживает операции определения данных, определения представлений, манипулирования данными (интерактивные и программные), ограничители целостности, управления доступом и операции управления <u>транзакциями</u>.</p>
7	<p>Возможность модификации представлений (View Updating Rule)</p>	<p>Каждое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции выборки, вставки, модификации и удаления данных.</p>

№	Правило	Описание
8	Наличие высокоуровневых операций управления данными (High-Level Insert, Update and Delete)	Операции вставки, модификации и удаления данных должны поддерживаться не только по отношению к одной строке реляционной таблицы, но по отношению к любому множеству строк.
9	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.
10	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.
11	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.
12	Дистрибутивная независимость (Distribution Independence)	База данных может быть распределенной, может находиться на нескольких компьютерах, и это не должно оказывать влияние на приложения. Перенос базы данных на другой компьютер не должен оказывать влияния на приложения.

№	Правило	Описание
13	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.

Правила Кодда позволили более четко определить требования к реляционным СУБД и помешать производителям ПО «размыть» это понятие, чтобы продавать старые решения под видом реляционных систем.