**12 правил Е.Кодда для реляционной БД (научный стиль)**

Наиболее распространены в настоящее время реляционные БД. Термин "реляционный" произошел от латинского слова "relatio" — отношение. Такая структура хранения данных построена на взаимоотношении составляющих ее частей. Реляционный подход стал широко известен благодаря работам Е. Кодда, которые впервые были опубликованы в 1970 году. В них Кодд сформулировал следующие 12 правил для реляционной БД:

1. **Данные представляются в виде таблиц.** БД представляет собой набор таблиц. Таблицы хранят данные, сгруппированные в виде рядов и колонок. Ряд представляет собой набор значений, относящихся только к одному объекту, хранящемуся в таблице, и называется записью. Колонка представляет собой одну характеристику для всех объектов, хранящихся в таблице, и называется полем. Ячейка на пересечении ряда и колонки представляет собой значение характеристики, соответствующей колонке для объекта соответствующего ряда.
2. **Данные доступны логически.** Реляционная модель не позволяет обращаться к данным физически, адресуя ячейки по номерам колонки и ряда (нет возможности получить значение в ячейке (колонка 2, ряд 3)). Доступ к данным возможен только через идентификаторы таблицы, колонки и ряда. Идентификаторами таблицы и колонки являются их имена. Они должны быть уникальны в пределах, соответственно, БД и таблицы. Идентификатором ряда является первичный ключ — значения одной или нескольких колонок, однозначно идентифицирующих ряды. Каждое значение первичного ключа в пределах таблицы должно быть уникальным. Если идентификация ряда осуществляется на основании значений нескольких колонок, то ключ называется составным.
3. **NULL трактуется как неизвестное значение.** Если в ячейку таблицы значение не введено, то записывается значение NULL. Его нельзя путать с пустой строкой или со значением 0.
4. **БД должна включать в себя метаданные.** БД хранит два вида таблиц: пользовательские и системные. В пользовательских таблицах хранятся данные, введенные пользователем. В системных таблицах хранятся мета-данные: описание таблиц (название, типы и размеры колонок), индексы, хранимые процедуры и др. Системные таблицы тоже доступны, т. е. пользователь может получить информацию о метаданных БД.
5. **Должен использоваться единый язык для взаимодействия с СУБД.** Для управления реляционной БД должен использоваться единый язык. В настоящее время таким инструментом стал язык SQL.
6. **СУБД должна обеспечивать альтернативный вид отображения данных.** СУБД не должна ограничивать пользователя только отображением таблиц, которые существуют. Пользователь должен иметь возможность строить виртуальные таблицы — представления (View). Представления являются динамическим объединением нескольких таблиц. Изменения данных в представлении должны автоматически переноситься на исходные таблицы (за исключением нередактируемых полей в представлении, например вычисляемых полей).
7. **Должны поддерживаться операции реляционной алгебры.** Записи реляционной БД трактуются как элементы множества, на котором определены операции реляционной алгебры. СУБД должна обеспечивать выполнение этих операций. В настоящее время выполнение этого правила обеспечивает язык SQL.
8. **Должна обеспечиваться независимость от физической организации данных.** Приложения, оперирующие с данными реляционных БД, не должны зависеть от физического хранения данных (от способа хранения, формата хранения и др.).
9. **Должна обеспечиваться независимость от логической организации данных.** Приложения, оперирующие с данными реляционных БД, не должны зависеть от организации связей между таблицами (логической организации). При изменении связей между таблицами не должны меняться ни сами таблицы, ни запросы к ним.
10. **За целостность данных отвечает СУБД.** Под целостностью данных в общем случае понимается готовность БД к работе. Различают следующие типы целостности:
* физическая целостность — сохранность информации на носителях и корректность форматов хранения данных;
* логическая целостность — непротиворечивость и актуальность данных, хранящихся в БД.

Потеря целостности базы данных может произойти из-за сбоев аппаратуры ЭВМ, ошибок в программном обеспечении, неверной технологии ввода и корректировки данных, низкой достоверности самих данных и т. д. За сохранение целостности данных должна отвечать СУБД, а не приложение, оперирующее ими. Различают два способа обеспечения целостности: декларативный и процедурный. При декларативном способе целостность достигается наложением ограничений на таблицы, при процедурном — обеспечивается с помощью хранимых в БД процедур.

1. **Целостность данных не может быть нарушена.** СУБД должна обеспечивать целостность данных при любых манипуляциях, производимых с ними.
2. **Должны поддерживаться распределенные операции.** Реляционная БД может размещаться как на одном компьютере, так и на нескольких — распределенно. Пользователь должен иметь возможность связывать данные, находящиеся в разных таблицах и на разных узлах компьютерной сети. Целостность БД должна обеспечиваться независимо от мест хранения данных.