

12 reglas de Codd

Codd se percató de que existían bases de datos en el mercado las cuales decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar estas tablas literalmente normalizadas; entonces éste publicó 12 reglas que un verdadero sistema relacional debería tener aunque en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse “más relacional” cuanto más siga estas reglas.

- **Regla 0:** el sistema debe ser relacional, base de datos y administrador de sistema. Ese sistema debe utilizar sus facilidades relacionales (exclusivamente) para manejar la base de datos.
- **Regla 1:** la regla de la información, toda la información en la base de datos es representada unidireccionalmente, por valores en posiciones de las columnas dentro de filas de tablas. Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera: con valores en tablas.
- **Regla 2:** la regla del acceso garantizado, todos los datos deben ser accesibles sin ambigüedad. Esta regla es esencialmente una nueva exposición del requisito fundamental para las llaves primarias. Dice que cada valor escalar individual en la base de datos debe ser lógicamente direccionable especificando el nombre de la tabla, la columna que lo contiene y la llave primaria.
- **Regla 3:** tratamiento sistemático de valores nulos, el sistema de gestión de base de datos debe permitir que haya campos nulos. Debe

tener una representación de la “información que falta y de la información inaplicable” que es sistemática, distinto de todos los valores regulares.

- **Regla 4:** catálogo dinámico en línea basado en el modelo relacional, el sistema debe soportar un catálogo en línea, el catálogo relacional debe ser accesible a los usuarios autorizados. Es decir, los usuarios deben poder tener acceso a la estructura de la base de datos (catálogo).
- **Regla 5:** la regla comprensiva del sublenguaje de los datos, el sistema debe soportar por lo menos un lenguaje relacional que:
 - Tenga una sintaxis lineal.
 - Puede ser utilizado de manera interactiva.
 - Soporte operaciones de definición de datos, operaciones de manipulación de datos (actualización así como la recuperación), seguridad e integridad y operaciones de administración de transacciones.
- **Regla 6:** regla de actualización, todas las vistas que son teóricamente actualizables deben ser actualizables por el sistema.
- **Regla 7:** alto nivel de inserción, actualización, y cancelación, el sistema debe soportar suministrar datos en el mismo tiempo que se inserte, actualiza o esté borrando. Esto significa que los datos se pueden recuperar de una base de datos relacional en los sistemas construidos de datos de filas múltiples y/o de tablas múltiples.
- **Regla 8:** independencia física de los datos, los programas de aplicación y actividades del terminal permanecen inalterados a nivel

lógico cuandoquiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

- **Regla 9:** independencia lógica de los datos, los cambios al nivel lógico (tablas, columnas, filas, etc.) no deben requerir un cambio a una solicitud basada en la estructura. La independencia de datos lógica es más difícil de lograr que la independencia física de datos.
- **Regla 10:** independencia de la integridad, las limitaciones de la integridad se deben especificar por separado de los programas de la aplicación y se almacenan en la base de datos. Debe ser posible cambiar esas limitaciones sin afectar innecesariamente las aplicaciones existentes.
- **Regla 11:** independencia de la distribución, la distribución de las porciones de la base de datos a las varias localizaciones debe ser invisible a los usuarios de la base de datos. Los usos existentes deben continuar funcionando con éxito:
 - Cuando una versión distribuida del SGBD se introdujo por primera vez
 - cuando se distribuyen los datos existentes se redistribuyen en todo el sistema.
- **Regla 12:** la regla de la no subversión, si el sistema proporciona una interfaz de bajo nivel de registro, a parte de una interfaz relacional, que esa interfaz de bajo nivel no se pueda utilizar para subvertir el sistema, por ejemplo: sin pasar por seguridad relacional o limitación de integridad. Esto es debido a que existen sistemas anteriormente no relacionales que añadieron una interfaz relacional, pero con la interfaz nativa existe la posibilidad de trabajar no relacionamente.