REGLAS DE CODD
(MODELO RELACIONAL)
12 Reglas.

En los años 80's comenzaron a surgir numerosos Sistemas de Gestión de Bases de Datos (en adelante SGBD) que se anunciaban como relacionales.

Sin embargo estos sistemas carecían de muchas características que se consideran importantes en un sistema relacional, perdiendo las muchas ventajas del modelo relacional.

Los sistemas relacionales eran simplemente sistemas que utilizaban tablas para almacenar la información, no disponiendo de elementos como claves primarias y foráneas que las definieran como tales.

En 1984 Edgar F. Codd, creador de del Modelo Relacional, publicó las 12 Reglas que un verdadero Sistema Relacional de Bases de Datos debería cumplir.

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.

Regla 1 La regla de la información:

"Cada ítem de datos debe ser lógicamente accesible al ejecutar una búsqueda que combine el nombre de la tabla, su clave primaria, y el nombre de la columna".

Toda la información en la base de datos es representada unidireccionalmente, por valores en posiciones de las columnas dentro de filas de tablas, exactamente de una manera: con valores en tablas.

Clientes			
idCliente	nombreCliente	appCliente	apmCliente
6234	Carlos	Rodríguez	Gutiérrez
6235	Diana	Velázquez	Sánchez
6233	Erika	García	Castillo

Los datos de las columnas deben estar almacenados en tablas dentro de las bases de datos. Las tablas que contienen tal información constituyen el Diccionario de Datos.

Toda la información en la Base de datos es representada de forma explícita y única a nivel lógico, por medio de valores en columnas y filas de tablas. Por tanto los metadatos (diccionario, catálogo) se representan y se manipulan exactamente igual que los datos de usuario, pudiéndose usar el mismo lenguaje.

La primera forma normal (1FN o forma mínima) es una forma normal usada en normalización de bases de datos. Una tabla de base de datos relacional que se adhiere a la 1FN es una que satisface cierto conjunto mínimo de criterios.

Todo atributo en una tabla tiene un dominio, el cual representa el conjunto de valores que el mismo puede tomar.

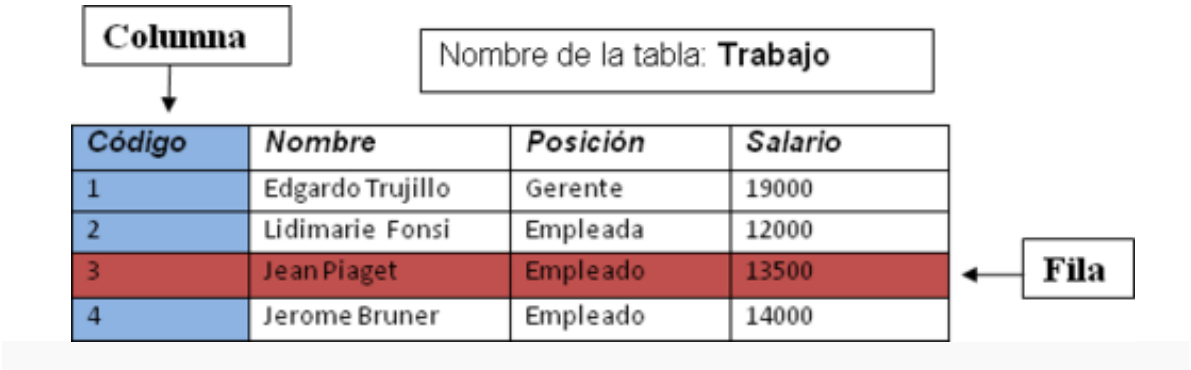
Regla 2 La regla del acceso garantizado:

Todos los datos deben ser accesibles sin ambigüedad. Esta regla es esencialmente una nueva exposición del requisito fundamental para las llaves primarias.

Clave primaria (Primary Key): es el valor o conjunto de valores que identifican una fila dentro de una tabla. Nunca puede ser NULL.

Dice que cada valor escalar individual en la base de datos debe ser lógicamente direccionarle especificando el nombre de la tabla, la columna que lo contiene y la llave primaria. Refuerza la importancia de las claves primarias para localizar datos en la base de datos.

Cada tabla debe tener su nombre único.
No puede haber dos filas iguales. No se permiten los duplicados.
Todos los datos en una columna deben ser del mismo tipo.



Código	Nombre	Posición	Salario
1	Edgardo Trujillo	Gerente	19000
2	Lidimarie Fonsi	Empleada	12000
3	Jean Piaget	Empleado	13500
4	Jerome Bruner	Empleado	14000

Todo dato (valor atómico) debe ser accesible mediante una combinación de tabla, un valor de su clave y el nombre de una columna.

Si se necesita acceder a la información en un orden concreto, se utilizan las construcciones sintácticas que proporcionan los lenguajes.

Cada ítem de datos debe ser lógicamente accesible al ejecutar una búsqueda que combine el nombre de la tabla, su clave primaria, y el nombre de la columna Esto significa que dado un nombre de tabla, dado el valor de la clave primaria, y dado el nombre de la columna requerida, deberá encontrarse uno y solamente un valor. Por esta razón la definición de claves primarias para todas las tablas es prácticamente obligatoria.

Regla 3: tratamiento sistemático de valores nulos

"La información inaplicable o faltante puede ser representada a través de valores nulos". Un RDBMS (Sistema Gestor de Bases de Datos Relacionales) debe ser capaz de soportar el uso de valores nulos en el lugar de columnas cuyos valores sean desconocidos o inaplicables.

Los valores nulos (que son distintos de la cadena vacía, blancos, 0) se soportan en los SGBD totalmente relacionales para representar información desconocida o no aplicable de manera sistemática, independientemente del tipo de datos.

Se reconoce la necesidad de la existencia de valores nulos, para un tratamiento sistemático de los mismos.

Hay problemas para soportar los valores nulos en las operaciones relacionales, especialmente en las operaciones lógicas.

Lógica trivaluada. En una posible solución. Existen tres (no dos) valores de verdad: Verdadero, Falso y Desconocido (null). Se crean tablas de verdad para las operaciones lógicas:

null Y null = null

Verdadero Y null = null

Falso Y null = Falso

Verdadero O null = Verdadero etc..

Un inconveniente es que de cara al usuario el manejo de los lenguajes relacionales se complica pues es más difícil de entender.

Ejemplo: Cuando la base no puede manejar la entrada de datos nulos, lo cual podría producirnos un error, por esto es conveniente tener en cuenta los dominios de cada tipo de atributo o campo de la tabla así como la inexistencia del dato requerido. Hay problemas para soportar los valores nulos en las operaciones relacionales, especialmente en las operaciones lógicas.

Regla 4: Catalogo en línea dinámico basado en el modelo relacional.

La descripción de la base de datos se representa a nivel lógico de la misma manera que los datos normales, de modo que los usuarios autorizados pueden aplicar el mismo lenguaje relacional a su consulta, igual que lo aplican a los datos normales.

Es una consecuencia de la regla 1 que se destaca por su importancia. Los metadatos se almacenan usando el modelo relacional, con todas las consecuencias.

La descripción de la base de datos es almacenada de la misma manera que los datos ordinarios, esto es, en tablas y columnas, y debe ser accesible a los usuarios autorizados. La información de tablas, vistas, permisos de acceso de usuarios autorizados, etc., debe ser almacenada exactamente de la misma manera: En tablas. Estas tablas deben ser accesibles igual que todas las tablas.

Ejemplo: El uso de clave para controlar lo que cada usuario puede ver o manejar dentro de la base de datos es muy importante y debe poder ser accesible en cualquier momento por quien así lo requiera. Debe de ser posible el acceso a datos y metadatos.

Regla 5 La regla del sub-lenguaje Integral:

“Debe haber al menos un lenguaje que sea integral para soportar la definición de datos, manipulación de datos, definición de vistas, restricciones de integridad, y control de autorizaciones y transacciones”. Esto significa que debe haber por lo menos un lenguaje con una sintaxis bien definida que pueda ser usado para administrar completamente la base de datos.

Un sistema relacional puede soportar varios lenguajes y varios modos de usó; Sin embargo, debe haber al menos un lenguaje cuyas sentencias sean expresables, mediante alguna sintaxis bien definida, cómo cadenas de caracteres, y que sea completa en cuanto al soporte de todos los puntos siguientes:

Tenga una sintaxis lineal.

Puede ser utilizado dentro de programas de uso.

Soporte operaciones de definición de datos, de manipulación de datos, seguridad e integridad y operaciones de administración de transacciones.

Autorización.

Fronteras de transacciones (comienzo, cumplimiento y vuelta atrás).

A si mismo debe tener interfaces más amigables para hacer consultas, etc. siempre debe de haber una manera de hacerlo todo de manera textual, que es tanto como decir que pueda ser incorporada en un programa tradicional.

Independientemente de que el SGBD ofrezca interfaces amigables para el usuario/administrador, se debe disponer de un lenguaje relacional con las mismas (o más) funciones que las interfaces. Cuando una base de datos no puede funcionar de forma correcta dentro de cierto tipo de lenguajes.

Regla 6: La regla de la actualización de vistas

Todas las vistas que son teóricamente actualizables, deben ser actualizables por el sistema mismo. La mayoría de las RDBMS permiten actualizar vistas simples, pero deshabilitan los intentos de actualizar vistas complejas.

La actualización debe de ser automática, sin necesidad de que el usuario tenga que estar actualizando manualmente. Una vista puede ser el conjunto de socios de la biblioteca que viven en Ciudad Capital. Si quiero añadir un socio que vive en Izabal a la vista (sería actualizable), debo poder hacerlo sin notarlo, debe encargarse el SGBD de manejarlo.

La base de datos relacional debe de poder ser capaz de realizar con éxito operaciones de inserción, actualización y borrado a través de sus capacidades relacionales. Si quiero eliminar a los socios de la biblioteca que sean de Izabal, no tengo que ir uno a uno. El sistema me debe proporcionar mecanismos para borrarlos todos de golpe.

REGLA 7: INSERCIÓN, ACTUALIZACIÓN Y BORRADO DE ALTO NIVEL:

La capacidad de manejar una relación base o derivada como un solo operando se aplica no sólo a la recuperación de los datos (consultas), sino también a la inserción, actualización y borrado de datos.

• Esto es, el lenguaje de manejo de datos también debe ser de alto nivel (de conjuntos). Algunos sistemas de bases de datos inicialmente sólo podían modificar las filas de una tabla de una en una (un registro de cada vez).

Esto significa que las cláusulas SELECT, UPDATE, DELETE e INSERT deben estar disponibles y operables sobre los registros, independientemente del tipo de relaciones y restricciones que haya entre las tablas.

REGLA 8: INDEPENDENCIA FÍSICA DE DATOS:

El acceso de usuarios a la base de datos a través de terminales o programas de aplicación, debe permanecer consistente; lógicamente cuando quiera que haya cambios en los datos almacenados, o sean cambiados los métodos de acceso a los datos.

El comportamiento de los programas de aplicación y de la actividad de usuarios vía terminales debería ser predecible basados en la definición lógica de la base de datos, y éste comportamiento debería permanecer inalterado, independientemente de los cambios en la definición física de ésta.

Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cualesquiera sean los cambios efectuados, tanto en la representación del almacenamiento, como en los métodos de acceso.

• El modelo relacional es un modelo lógico de datos, y oculta las características de su representación física.

REGLA 9: INDEPENDENCIA LÓGICA DE DATOS:

Los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cualesquiera sean los cambios que se realicen a las tablas base que preserven la información.

La independencia lógica de los datos especifica que los programas de aplicación y las actividades de terminal deben ser independientes de la estructura lógica, por lo tanto los cambios en la estructura lógica no deben alterar o modificar estos programas de aplicación.

• Cuando se modifica el esquema lógico preservando información (no valdría por ejemplo, eliminar un atributo) no es necesario modificar nada en niveles superiores.

Ejemplos de cambios que preservan la información: o Añadir un atributo a una tabla base. o Sustituir dos tablas base por la unión de las mismas. Usando vistas de la unión se pueden recrear las tablas anteriores.

Regla 10: independencia de integridad:

Todas las restricciones de integridad deben ser definibles en los datos, y almacenables en el catálogo, no en el programa de aplicación.

Las reglas de integridad son:

1. Ningún componente de una clave primaria puede tener valores en blanco o nulos. (Esta es la norma básica de integridad).

2. Para cada valor de clave foránea deberá existir un valor de clave primaria concordante. La combinación de estas reglas asegura que haya Integridad referencial.

El objetivo de las bases de datos no es sólo almacenar los datos, sino también sus relaciones y evitar que estas (limitantes) se codifiquen en los programas.

Cada vez se van ampliando más los tipos de limitantes de integridad que se pueden utilizar en los SGBDR, aunque hasta hace poco eran muy escasos. Como parte de los limitantes inherentes al modelo relacional (forman parte de su definición).

Ejemplo: El objetivo de las bases de datos no es sólo almacenar los datos, si no también sus relaciones y evitar que estas (limitantes) se codifiquen en los programas.

Por tanto en una BDR se deben poder definir limitantes de integridad. Una BDR tiene integridad de entidad. Es decir, toda tabla debe tener una clave primaria. Una BDR tiene integridad referencial. Es decir, toda clave externa no nula debe existir en la relación donde es primaria.

Regla 11: independencia de distribución:

El sistema debe poseer un lenguaje de datos que pueda soportar que la base de datos esté distribuida físicamente en distintos lugares sin que esto afecte o altere a los programas de aplicación. El soporte para bases de datos distribuidas significa que una colección arbitraria de relaciones, bases de datos corriendo en una mezcla de distintas máquinas y distintos sistemas operativos y que esté conectada por una variedad de redes, pueda funcionar como si estuviera disponible como en una única base de datos en una sola máquina.

Una base de datos relacional tiene independencia de distribución.

Las mismas órdenes y programas se ejecutan igual en una BD centralizada que en una distribuida.

Las BDR son fácilmente distribuibles:

Las tablas se dividen en fragmentos que se distribuyen.

Cuando se necesitan las tablas completas se recombinan usando operaciones relacionales con los fragmentos.

Sin embargo se complica más la gestión interna de la integridad, etc.

Esta regla es responsable de tres tipos de transparencia de distribución:

Transparencia de localización. El usuario tiene la impresión de que trabaja con una BD local. (aspecto de la regla de independencia física)

Transparencia de fragmentación. El usuario no se da cuenta de que la relación con que trabaja está fragmentada. (aspecto de la regla de independencia lógica de datos).

Transparencia de replicación. El usuario no se da cuenta de que pueden existir copias (réplicas) de una misma relación en diferentes lugares.

Ejemplo: Las mismas órdenes y programas se ejecutan igual en una BD centralizada que en una distribuida. Voy a sacar dinero de un cajero. El dinero de mi cuenta se guarda en un servidor.

Saco un extracto de movimientos. Y esa información se guarda ¿En el mismo servidor de antes? No tiene por qué y yo no tengo porqué saberlo (transparencia). En un servidor, no tiene por qué guardarse todos los datos de todos los clientes de un banco.

Es más lógico que se guarden en varios servidores, uno por zona geográfica, por ejemplo. Los clientes no tienen por qué saberlo. ¿Ellos lo notan?, No y además se tiene que encargar el SGBD de que sea transparente.

Regla 12: regla de la no subversión:

Si un sistema relacional tiene un lenguaje de bajo nivel (un registro de cada vez), ese bajo nivel no puede ser usado para saltarse (subvertir) las reglas de integridad y los limitantes expresados en los lenguajes relacionales de más alto nivel (una relación (conjunto de registros) de cada vez).

Algunos productos solamente construyen una interfaz relacional para sus bases de datos No relacionales, lo que hace posible la subversión (violación) de las restricciones de integridad. Esto no debe ser permitido.

Algunos problemas no se pueden solucionar directamente con el lenguaje de alto nivel. Normalmente se usa SQL inmerso en un lenguaje anfitrión para solucionar estos problemas. Se utiliza el concepto de cursor para tratar individualmente las tuplas de una relación.

En cualquier caso no debe ser posible saltarse los limitantes de integridad impuestos al tratar las tablas a ese nivel.

Ejemplo: No debe ser posible saltarse los limitantes de integridad impuestos al tratar las tuplas a ese nivel. Si se puede, con las facilidades que da el SGBD utilizar un sistema para acceder a los registros (desde aplicaciones externas al SGBD), éste sistema debe respetar todas las reglas anteriores. Debe seguir manteniendo todas las integridades de los datos.