

**INFORME DE BASE DE DATOS RELACIONALES**

Programa de Especialización en Desarrollo Back-end

Alumno:

sv71256629 – Sofia Gabriel Miranda Cardenas

Marzo 2025

Contenido

[**1.** **12 reglas de Codd** 3](#_Toc194257121)

[1.1. Regla de la Información 3](#_Toc194257122)

[1.2. Regla del acceso garantizado 3](#_Toc194257123)

[1.3. Regla del tratamiento sistemático de valores nulos 3](#_Toc194257124)

[1.4. Catalogo dinámico en línea basado en el modelo relacional 3](#_Toc194257125)

[1.5. Regla comprensiva del sub-lenguaje de los datos completos 3](#_Toc194257126)

[1.6. Regla de actualización de vistas 3](#_Toc194257127)

[1.7. Alto nivel de inserción, actualización y cancelación 3](#_Toc194257128)

[1.8. Independencia física de los datos 3](#_Toc194257129)

[1.9. Independencia lógica de los datos 3](#_Toc194257130)

[1.10. Independencia de la integridad 3](#_Toc194257131)

[1.11. Independencia de la distribución 4](#_Toc194257132)

[1.12. Regla de la no subversión 4](#_Toc194257133)

[**2.** **Normalización de Base de Datos** 4](#_Toc194257134)

[2.1. Primera Forma Normal (1FN) 4](#_Toc194257135)

[2.2. Segunda Forma Normal (2FN) 4](#_Toc194257136)

[2.3. Tercera Forma Normal (3FN) 4](#_Toc194257137)

[2.4. Cuarta Forma Normal (4FN) 4](#_Toc194257138)

[**3.** **El modelo relacional para el modelado y la gestión de bases de datos** 4](#_Toc194257139)

[3.1. Características clave del modelo relacional 4](#_Toc194257140)

[**4.** **Línea del tiempo** 6](#_Toc194257141)

[**5.** **Conclusión** 6](#_Toc194257142)

## **12 reglas de Codd**

### Regla de la Información

Todos los datos se organizan en tablas lógicas (como hojas de cálculo), donde cada fila representa un registro y cada columna representa un tipo de información.

### Regla del acceso garantizado

Para encontrar cualquier dato, solo necesitas saber el nombre de la tabla, la clave principal (Primary Key) y el nombre de la columna donde esta almacenado el dato que buscas.

### Regla del tratamiento sistemático de valores nulos

Los valores nulos pueden tener significado y el sistema debe tener la capacidad para manejarlos, reconociéndolos y tratándolos diferente de otros datos como vacíos o 0. La lógica aplicada debe ser correcta, y los valores nulos son independientes del dato de la columna, siendo esenciales para representar la ausencia de información de manera precisa.

### Catalogo dinámico en línea basado en el modelo relacional

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

### Regla comprensiva del sub-lenguaje de los datos completos

El lenguaje que utilizas para trabajar con las bases de datos debe permitirte hacer cualquier cosa que necesites, como agregar nuevos datos, cambiar los existentes o hacer consultas complejas.

### Regla de actualización de vistas

Si puedes ver los datos de una manera especial (como una vista), deberías poder actualizarlos también, y el sistema se encargará de aplicar esos cambios correctamente.

### Alto nivel de inserción, actualización y cancelación

Se centra en simplificar la forma en que interactuamos con la información almacenada. En lugar de tener que programar cada operación individualmente como lo haríamos en lenguajes como C o Java, el objetivo es utilizar un lenguaje que sea más fácil de entender para los humanos. Esto significa que podemos manipular conjuntos de datos a la vez, lo que facilita la inserción, actualización y eliminación de información sin necesidad de escribir código complicado.

### Independencia física de los datos

La forma en que se almacenan los datos no debería importarte al interactuar con ellos. Pueden estar guardados de diferentes maneras, pero eso no afecta como los usas.

### Independencia lógica de los datos

Si cambias la forma en que ves los datos (por ejemplo, reorganizas las tablas), no debería afectar a las aplicaciones que usan esos datos.

### Independencia de la integridad

La regla que garantiza que los datos sean correctos y completos deben estar separadas de las aplicaciones que lo utilizan.

### Independencia de la distribución

Si los datos están en diferentes lugares físicos, el sistema debe poder manejar eso sin problemas, como si estuvieran todos en un solo lugar.

### Regla de la no subversión

Aunque puedas acceder a la base de datos de formas más avanzadas, no deberías poder evitar las reglas y restricciones que has establecido para garantizar la integridad de los datos.

## **Normalización de Base de Datos**

La normalización se refiere a la reducción o eliminación del almacenamiento de datos redundantes. El objetivo principal de la normalización es evitar problemas asociados con la actualización de datos redundantes.

El modelo de datos relacionales, propuesto por Edgar Frank Ter Codd en 1970, sentó las bases para las 12 Reglas de Codd para el modelo relacional. Estas reglas son fundamentales en el tema que abordaremos hoy: la normalización de bases de datos.

### Primera Forma Normal (1FN)

Esta forma nos ayuda a eliminar valores repetidos y no atómicos en nuestra base de datos. ¿Qué significa eso? Piensa en una tabla donde tienes un campo como "nombre\_completo" que realmente contiene múltiples datos distintos como "nombres", "apellido paterno", "apellido materno".

### Segunda Forma Normal (2FN)

Aquí diferenciamos los datos en diversas entidades. ¿Cómo? Asegurándonos de que cada atributo que no es clave principal dependa completamente de la clave principal. Esto implica separar entidades para que cada una dependa de manera única de su campo clave.

### Tercera Forma Normal (3FN)

Esta forma nos ayuda a separar conceptualmente entidades que no son dependientes. ¿Cómo lo logramos? Eliminando cualquier dependencia funcional transitiva en los atributos que no son clave. Esto significa que cada dato debe tener una independencia clara de otros datos no relacionados.

### Cuarta Forma Normal (4FN)

Aquí atomizamos los datos multivaluados para evitar datos repetidos entre filas. ¿Cómo? Identificando campos multivaluados por una clave única y eliminando registros duplicados. Esto simplifica la estructura de nuestra base de datos y facilita el mantenimiento de la consistencia de los datos.

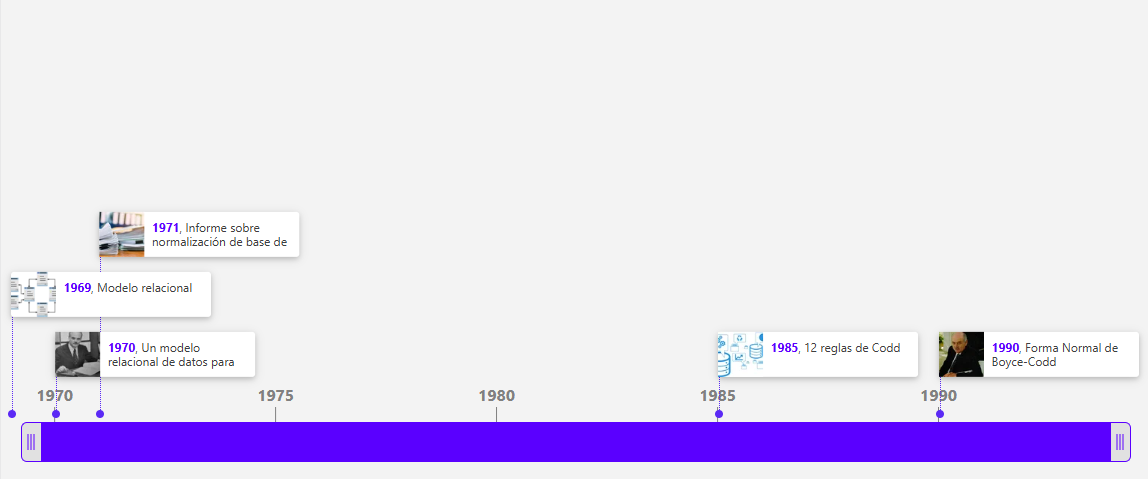
## **El modelo relacional para el modelado y la gestión de bases de datos**

El modelo relacional es un modelo para el diseño y gestión de bases de datos que organiza los datos en tablas o "relaciones", donde cada tabla está compuesta por filas (registros) y columnas (atributos). Este modelo es la base de la mayoría de los sistemas de gestión de bases de datos (DBMS) actuales, como MySQL, PostgreSQL, SQL Server, Oracle, entre otros.

### Características clave del modelo relacional

* + 1. **Tablas (Relaciones):** Las bases de datos relacionales se estructuran en tablas, donde cada tabla tiene un conjunto de columnas (atributos) y filas (tuplas o registros). Cada tabla representa una entidad o un conjunto de entidades con características comunes. Ejemplo: Una tabla llamada "Clientes" podría tener columnas como "ID\_Cliente", "Nombre", "Dirección", "Teléfono".
    2. **Filas (Tuplas):** Cada fila de la tabla representa un registro individual. Cada fila contiene valores para cada una de las columnas de la tabla.
    3. **Columnas (Atributos):** Cada columna de una tabla representa un atributo o característica de los elementos que la tabla almacena. Los atributos tienen un tipo de dato (números, texto, fecha, etc.).
    4. **Clave primaria (Primary Key):** Cada tabla tiene una clave primaria, que es un atributo o conjunto de atributos cuyo valor es único para cada fila de la tabla. Esto permite identificar de manera inequívoca cada registro. Por ejemplo, en la tabla "Clientes", el "ID\_Cliente" podría ser la clave primaria.
    5. **Claves foráneas (Foreign Keys):** Las claves foráneas son atributos que establecen una relación entre dos tablas. Una clave foránea en una tabla hace referencia a la clave primaria de otra tabla. Estas claves permiten la creación de relaciones entre tablas, como una relación de uno a muchos o de muchos a muchos. Por ejemplo, en una tabla "Pedidos", la columna "ID\_Cliente" podría ser una clave foránea que hace referencia al "ID\_Cliente" en la tabla "Clientes".
    6. **Integridad referencial:** Es una propiedad del modelo relacional que asegura que las relaciones entre tablas se mantengan consistentes. Esto significa que si una clave foránea en una tabla hace referencia a una clave primaria en otra, los valores de la clave foránea deben coincidir con los de la clave primaria, o la clave foránea debe ser nula.
    7. **Operaciones relacionales:** El modelo relacional permite realizar varias operaciones sobre las tablas, como:
* **Selección (SELECT):** Extraer datos de una o más tablas.
* **Proyección (PROJECTION):** Elegir un subconjunto de columnas.
* **Unión (UNION):** Combinar los resultados de dos consultas con el mismo número de columnas y tipos de datos.
* **Intersección (INTERSECTION):** Encontrar los registros comunes entre dos conjuntos de resultados.
* **Diferencia (DIFFERENCE):** Encontrar los registros que están en un conjunto, pero no en otro.
* **Producto cartesiano (CARTESIAN PRODUCT):** Combina todas las filas de una tabla con todas las filas de otra.
  + 1. **Normalización:** Es un proceso que busca organizar las tablas de la base de datos de tal forma que se reduzcan las redundancias y dependencias innecesarias, mejorando la eficiencia y evitando problemas de inconsistencia de datos. La normalización se lleva a cabo dividiendo grandes tablas en otras más pequeñas y relacionadas entre sí, asegurando que cada tabla tenga una función clara y evitando la duplicación de información.

## **Línea del tiempo**

****

Link para mejor visualización: <https://www.timetoast.com/timelines/3163207>

## **Conclusión**

El modelo relacional es un pilar fundamental para el diseño y gestión de bases de datos, proporcionando una estructura lógica y clara basada en tablas, filas y columnas. Las 12 reglas de Codd establecen principios que aseguran la consistencia, flexibilidad y accesibilidad de los datos, garantizando una gestión eficiente y robusta. La normalización, a su vez, es crucial para evitar redundancias y dependencias innecesarias, optimizando la base de datos y manteniendo la integridad de los datos.

El uso de claves primarias y claves foráneas, junto con la integridad referencial, permite crear relaciones entre las tablas y garantiza que los datos estén bien organizados. Las operaciones relacionales, como selección, unión e intersección, facilitan la manipulación y consulta de la información. Además, la capacidad para actualizar vistas y la independencia lógica y física de los datos proporcionan una gran flexibilidad.

En resumen, el modelo relacional, con sus reglas y características, es esencial para construir bases de datos eficaces y escalables, asegurando un manejo coherente y sin redundancias de grandes volúmenes de información. La normalización y las operaciones relacionales son herramientas clave para mantener la integridad y eficiencia de las bases de datos.